



International Committee for Future Accelerators

Sponsored by the Particles and Fields Commission of IUPAP

Beam Dynamics Newsletter

No. 47

**Issue Editor:
I. S. Ko**

**Editor in Chief:
W. Chou**

December 2008

Contents

1	FOREWORD.....	11
1.1	FROM THE CHAIR	11
1.2	FROM THE EDITOR	12
2	LETTERS TO THE EDITOR – IN MEMORY OF ROBERT H. SIEMANN	14
2.1	ROBERT H. SIEMANN – ENCOUNTERS AND ESSAYS	14
2.1.1	Introduction	14
2.1.2	Numerous Encounters	14
2.1.3	Memorial Essays	17
2.1.4	References	17
2.2	REMEMBERING BOB SIEMANN AS AN EARLY MENTOR	18
2.3	BOB SIEMANN AND MESON PRODUCTION BY POLARIZED PHOTONS	19
2.3.1	Introduction	19
2.3.2	Polarization Dependence of Meson Photoproduction	20
2.3.3	Construction, Characterization, and Use of the SLAC Polarized Photon Beamline.....	22
2.3.4	References	22
2.4	MEMORIES OF A MENTOR AND FRIEND – ROBERT H. SIEMANN	23
2.4.1	Introduction	23
2.4.2	Cornell Years.....	23
2.4.3	Fermilab Sabbatical.....	25
2.4.4	Lasting Mentor and Friend.....	25
2.5	ROBERT H. SIEMANN: A PERSONAL TRIBUTE	26
2.6	IN MEMORY OF ROBERT SIEMANN.....	29
2.7	SLC DAYS AT SLAC	30
2.8	ESSAY FOR ROBERT H. SIEMANN	32
2.9	ROBERT H. SIEMANN AS LEADER OF AARD.....	33
2.9.1	Introduction	33
2.9.2	Scientist, Colleague, and Collaborator	34
2.9.3	Professor.....	34
2.9.4	Mentor ..	35
2.9.5	Administrator.....	35
2.9.6	Hobbyist	35
2.9.7	Conclusion.....	36

2.10	SIEMANN TRIBUTE	36
	References	37
2.11	BOB SIEMANN'S CONTRIBUTIONS TO ADVANCED ACCELERATORS – A PERSONAL PERSPECTIVE.....	38
	References	40
2.12	BOB SIEMANN AND THE SPALLATION NEUTRON SOURCE: A REMEMBRANCE.....	40
2.13	BOB SIEMANN AND PRST-AB	42
2.14	HANDS ACROSS THE ATLANTIC	44
3	INTERNATIONAL LINEAR COLLIDER (ILC).....	46
3.1	THIRD INTERNATIONAL ACCELERATOR SCHOOL FOR LINEAR COLLIDERS.....	46
4	THEME SECTION: CONTROL SYSTEMS.....	49
4.1	ACCELERATOR CONTROL SYSTEMS – A PERSONAL RECOLLECTION	49
	4.1.1 Introduction.....	49
	4.1.2 The TRIUMF Years.....	49
	4.1.3 A Community	50
	4.1.4 Collaboration - EPICS	51
	4.1.5 The SSC.....	52
	4.1.6 The SNS.....	53
	4.1.7 Conclusions.....	55
4.2	EPICS – EXPERIMENTAL PHYSICS AND INDUSTRIAL CONTROL SYSTEM.....	56
	4.2.1 Brief History	56
	4.2.2 Architecture	56
	4.2.3 Process Database	58
	4.2.3.1 <i>Record Processing</i>	58
	4.2.3.2 <i>Hardware Support</i>	58
	4.2.4 Software Interfaces	59
	4.2.4.1 <i>Driver Layer Interface</i>	59
	4.2.4.2 <i>Record Type Support</i>	59
	4.2.4.3 <i>Channel Access Client Interface</i>	60
	4.2.4.4 <i>Hardware Platforms</i>	60
	4.2.5 Channel Access Clients and Operator Tools	60
	4.2.5.1 <i>Services: Archiving</i>	61
	4.2.5.2 <i>Services: Alarming</i>	61
	4.2.5.3 <i>Services: Logging</i>	61
	4.2.5.4 <i>Operator Tools: X-Window/Motif Based Applications</i>	61
	4.2.5.5 <i>Operator Tools: Java Based Applications</i>	61
	4.2.6 Applications & Their Requirements.....	61
	4.2.7 Robustness and Reliability	62
	4.2.7.1 <i>Robustness</i>	62
	4.2.7.2 <i>Reliability</i>	63

4.2.7.3	<i>Redundancy</i>	63
4.2.8	Current Plans	63
4.2.8.1	<i>EPICS Core: Java IOC</i>	63
4.2.8.2	<i>New Directions in Channel Access</i>	64
4.2.8.3	<i>Operator Tools</i>	64
4.2.8.4	<i>Web Based Applications</i>	64
4.2.8.5	<i>Services</i>	65
4.2.9	Industrial Support	65
4.2.10	The EPICS Collaboration	65
4.2.11	Conclusions	66
4.2.12	References	66
4.3	THE TANGO CONTROL SYSTEM	66
4.3.1	History	66
4.3.2	TANGO Philosophy	67
4.3.3	CORBA	67
4.3.4	TANGO Protocols	68
4.3.5	Performance	68
4.3.6	Code Generation	69
4.3.7	Administration	71
4.3.8	Graphic Tools	72
4.3.9	Diagnostics Applications	74
4.3.9.1	<i>Vacuum</i>	74
4.3.9.2	<i>Closed Orbit</i>	76
4.3.10	Current Status	77
4.3.11	Constant Evolution	78
4.3.12	TANGO Feature Requests	78
4.3.13	New Features	79
4.3.14	Collaboration	79
4.3.15	Stability, Quality, Packaging	79
4.3.16	System Tools	80
4.3.17	Standard Interfaces	80
4.3.18	Conclusion	80
4.3.19	References	80
4.4	THE LHC CONTROLS SYSTEM	81
4.4.1	Introduction	81
4.4.2	Details of the Three-Tier Architecture	82
4.4.3	Other Components	86
4.4.4	Outcome from the Injection Tests	88
4.4.5	Collaborations and Outlook	89
4.4.6	Further Reading	89
4.5	THE ALICE CONTROL SYSTEM	90
4.5.1	Introduction	90
4.5.1.1	<i>LHC and Its Experiments</i>	90
4.5.1.2	<i>Control Systems for HEP Experiments</i>	90
4.5.1.3	<i>The ALICE Experiment</i>	91
4.5.2	The ALICE Detector Control System (DCS)	92

4.5.2.1	<i>Introduction</i>	92
4.5.2.2	<i>Design Strategy and System Architecture</i>	93
4.5.2.3	<i>System Implementation - Components</i>	97
4.5.2.4	<i>System Implementation - Applications</i>	100
4.5.3	Installation, Commissioning and First Operational Experience	103
4.5.4	Summary and Outlook	104
4.5.5	Acknowledgements.....	105
4.5.6	References.....	105
4.6	THE FERMILAB ACCELERATOR CONTROL SYSTEM	106
4.6.1	Introduction.....	106
4.6.2	Fermilab Accelerator Complex.....	106
4.6.3	Control System Overview.....	106
4.6.4	Device Model.....	107
4.6.5	Communication Protocol	108
4.6.6	Timing System.....	109
4.6.7	Supported Hardware	110
4.6.7.1	<i>Data Acquisition</i>	110
4.6.7.2	<i>Ramp Generators</i>	110
4.6.7.3	<i>Field Buses</i>	111
4.6.8	Front-End Systems.....	111
4.6.8.1	<i>Central Services</i>	112
4.6.8.2	<i>Data Logging</i>	112
4.6.8.3	<i>Sequenced Data Acquisition</i>	113
4.6.8.4	<i>Save/Restore</i>	113
4.6.8.5	<i>Alarms</i>	113
4.6.8.6	<i>Front-End Download</i>	113
4.6.8.7	<i>Accountability</i>	113
4.6.8.8	<i>Data Acquisition Engines</i>	113
4.6.8.9	<i>Servlets</i>	114
4.6.8.10	<i>Time-Line Generator</i>	114
4.6.8.11	<i>Experiment Communication</i>	114
4.6.8.12	<i>Databases</i>	115
4.6.8.13	<i>Application Frameworks</i>	115
4.6.8.14	<i>CLIB Framework</i>	115
4.6.8.15	<i>Java Framework</i>	116
4.6.9	Key Applications	117
4.6.9.1	<i>Parameter Page</i>	117
4.6.9.2	<i>Fast Time Plot Utility</i>	117
4.6.9.3	<i>Snapshot Plot Utility</i>	118
4.6.9.4	<i>ACL</i>	118
4.6.9.5	<i>Sequencer</i>	118
4.6.9.6	<i>Lex SA</i>	119
4.6.9.7	<i>Java Synoptic</i>	119
4.6.9.8	<i>Web Applications</i>	120
4.6.10	Console Infrastructure	120
4.6.11	Security	121
4.6.12	Past Evolution.....	122

4.6.13	Future Directions	123
4.6.14	Summary	123
4.6.15	Acknowledgements	124
4.6.16	References	124
4.7	THE DIAMOND LIGHT SOURCE CONTROL SYSTEM	125
4.7.1	Introduction	125
4.7.2	Diamond Description	125
4.7.3	Control System Overview	126
4.7.4	Hardware	126
4.7.5	Physical Structure	127
4.7.6	Network	127
4.7.7	Servers and Consoles	127
4.7.8	General Applications	128
4.7.9	Physics Applications	129
4.7.10	Technical Systems	130
4.7.10.1	<i>Magnets and Power Supplies</i>	130
4.7.10.2	<i>Beam Position Diagnostics</i>	131
4.7.10.3	<i>Orbit Position Feedback</i>	134
4.7.10.4	<i>Timing System</i>	135
4.7.10.5	<i>Machine Protection System</i>	136
4.7.10.6	<i>Linac, RF, Vacuum and Other Systems</i>	137
4.7.11	Conclusions	138
4.7.12	Acknowledgements	138
4.7.13	References	138
4.8	THE ACCELERATOR CONTROL SYSTEM AT DESY	139
4.8.1	Introduction	139
4.8.2	Requirements for a FEL Control System	140
4.8.3	Implementation in DOOCS	141
4.8.3.1	<i>The DOOCS Architecture</i>	141
4.8.4	The DOOCS Application Programming Interface	143
4.8.4.1	<i>Object-Oriented Server Design</i>	144
4.8.4.2	<i>Archiving</i>	145
4.8.4.3	<i>Server Location</i>	146
4.8.4.4	<i>Related Libraries</i>	146
4.8.4.5	<i>Toroid Example</i>	146
4.8.5	Control System Hardware and Infrastructure	147
4.8.5.1	<i>Front-End Hardware</i>	147
4.8.5.2	<i>Timing System</i>	147
4.8.5.3	<i>The Machine Protection System</i>	148
4.8.5.4	<i>Server Hardware</i>	148
4.8.5.5	<i>Software Packaging Systems</i>	148
4.8.5.6	<i>Remote Installation, Configuration and Maintenance</i>	148
4.8.6	Data Acquisition System	149
4.8.6.1	<i>The DAQ Architecture</i>	149
4.8.6.2	<i>The Front-End Hardware</i>	150
4.8.6.3	<i>Data Collectors</i>	150

4.8.6.4	<i>Buffer Manager</i>	150
4.8.6.5	<i>Middle Layer Processes</i>	151
4.8.6.6	<i>Distributor and Event Builder</i>	151
4.8.6.7	<i>Data Storage</i>	152
4.8.6.8	<i>Run Controller</i>	152
4.8.6.9	<i>Control System Integration</i>	152
4.8.6.10	<i>DAQ Graphical User Interfaces</i>	153
4.8.6.11	<i>Current DAQ Status</i>	153
4.8.7	Feedbacks and Automation.....	154
4.8.8	Camera Controls.....	156
4.8.9	MATLAB.....	157
4.8.10	Tools: Alarm, jDTool, Save & Restore.....	159
4.8.10.1	<i>Alarm & Info System</i>	159
4.8.10.2	<i>The jDTool application</i>	160
4.8.11	The DOOCS Data Displays ddd and jddd.....	161
4.8.11.1	<i>The ddd Application</i>	161
4.8.11.2	<i>The jddd Application</i>	162
4.8.11.3	<i>The jddd Editor</i>	162
4.8.11.4	<i>jddd Components and Widgets</i>	163
4.8.11.5	<i>jddd Edit and Runtime Mode</i>	164
4.8.11.6	<i>jddd Editor Features</i>	164
4.8.12	DOOCS Related Web Services.....	166
4.8.13	Conclusion and Outlook.....	166
4.8.14	References.....	166
4.9	TINE: THE CONTROL SYSTEM FOR PETRA3 AND THE DESY PRE-ACCELERATORS	167
4.9.1	Introduction.....	167
4.9.1.1	<i>The DESY Accelerator Chain</i>	167
4.9.1.2	<i>Requisite Features of the Control System for the DESY Accelerator Chain</i>	168
4.9.2	TINE: The Control System Expert's Perspective.....	169
4.9.2.1	<i>Three-fold Integrated Network Environment</i>	169
4.9.2.2	<i>Control System Services</i>	170
4.9.2.3	<i>Special Features</i>	172
4.9.3	TINE: User's Perspective.....	173
4.9.3.1	<i>Accelerator Physicists</i>	173
4.9.3.2	<i>Control Room Operators and Client Application Programmers</i>	174
4.9.3.3	<i>Accelerator Coordinators</i>	177
4.9.3.4	<i>TINE: Remote Access with the Web2c Toolkit</i>	178
4.9.3.5	<i>Synoptic Display</i>	179
4.9.4	References.....	179
4.9.5	DOOCS - TINE: A Common Future.....	180
4.10	FUTURE TRENDS.....	180
4.10.1	Introduction.....	180
4.10.2	Control System Design.....	181
4.10.2.1	<i>The Three Tier Architecture</i>	181
4.10.3	Front-end Controllers.....	181

4.10.3.1	Software.....	181
4.10.3.2	Hardware.....	181
4.10.4	Networking.....	182
4.10.4.1	Control System Protocols.....	182
4.10.4.2	The TINE Control System Protocol.....	182
4.10.5	Operator Tools.....	183
4.10.6	High Level Applications	184
4.10.7	Collaborations and Institutions.....	184
4.10.8	Globalization	185
4.10.9	Software Sharing	185
4.10.10	Conclusions	185
4.10.11	References	185
5	FORTHCOMING BEAM DYNAMICS EVENTS	186
5.1	45 TH ICFA ADVANCED BEAM DYNAMICS WORKSHOP: ENERGY RECOVERY LINACS (ERL09)	186
5.2	ICFA MINI-WORKSHOP: PHENOMENOLOGY WORKSHOP ON ADVANCED QED METHODS FOR FUTURE ACCELERATORS.....	187
5.3	ICFA MINI-WORKSHOP ON LASER-ASSISTED H ⁻ BEAM STRIPPING.....	187
6	ANNOUNCEMENTS OF THE BEAM DYNAMICS PANEL.....	188
6.1	ICFA BEAM DYNAMICS NEWSLETTER.....	188
6.1.1	Aim of the Newsletter	188
6.1.2	How to Prepare a Manuscript	189
6.1.3	Distribution.....	189
6.1.4	Regular Correspondents	190
6.2	ICFA BEAM DYNAMICS PANEL MEMBERS	191

1 Foreword

1.1 From the Chair

Weiren Chou, Fermilab
Mail to: chou@fnal.gov

The International Committee for Future Accelerators (ICFA) met on October 30, 2008 at SLAC during the 2008 ICFA Seminar. Since the present ICFA Chair Albrecht Wagner's term will expire at the end of this year, the committee approved the appointment of Atsuto Suzuki (Director General of KEK) unanimously as Chair of ICFA for a three-year term starting on January 1, 2009. Two senior ICFA members, Robert Aymar and Jonathan Dorfan will also retire from ICFA and be replaced by Rolf Heuer (new Director General of CERN) and Persis Drell (Director of SLAC).

The ICFA Seminar is a major ICFA event that takes place once every three years. This year it was hosted by SLAC and took place from October 28-31 at SLAC. The attendance was by invitation only. About 176 people attended, including government officials involved in strategic decisions for high-energy physics, representatives of the major funding agencies, the directors of major HEP laboratories and leading HEP physicists from around the world. The ICFA Seminar is planned as a venue for the international committee to examine plans for large future science projects from a global perspective and as a forum for discussion on the trends in HEP with emphasis on globalization issues. The agenda and presentations can be found on the web: <http://www-conf.slac.stanford.edu/icfa2008/>. In future seminars, it is planned also to invite science journalists to attend.

The Third International Accelerator School for Linear Colliders was held from October 19 to 29, 2008 at the Oak Brook Hills Marriott Hotel near Chicago, U.S.A. All lectures, homework assignments and final exam problems are posted on the web: <http://www.linearcollider.org/school/2008/>. A school report can be found in Section 3.1. The school is highly selective. From 245 applications, the committee admitted only 57 students from 14 countries. However, a number of them could not come due to visa problems. The school as well as the Fermilab International Affairs Office mounted a significant effort to help the 24 students who needed visas to enter the U.S. But on the last day before the school, visa applications for 9 of the students were still pending and they had to cancel their trip. This was very unfortunate and another example showing that the U.S. visa delays are actually getting worse, not better. It has been a serious problem for many years and must be addressed at the highest government level. We will be watching whether President-elect Barack Obama addresses concerns about the visa issue through an executive order early in his Administration.

The Beam Dynamics Panel approved two ICFA mini-workshops. One is *Phenomenology Workshop on Advanced QED Methods for Future Accelerators*, which will take place March 3-4, 2009 at the Cockcroft Institute, Daresbury, U.K. Another is *Laser-Assisted H^- Beam Stripping* to be held February 18-19, 2009 at ORNL, U.S.A.

This issue also has an announcement of the 45th *Advanced Beam Dynamics Workshop: ERL09*, which will take place from June 8-12, 2009 at Cornell University, U.S.A.

We are pleased to announce two new Regular Correspondents: Jacob Rodnizki from Soreq NRC, Israel and Rohan Dowd from Australian Synchrotron. We welcome them on board. They will cover the accelerator community activities in their region, which is an important part of this newsletter. We are calling for more volunteers from smaller institutions and countries as Regular Correspondents.

On September 16, 2008 we heard the sad news that Bob Siemann (SLAC) passed away. Bob had a huge influence in our community. I remember in 1986 when I just started my career in the accelerator field at Argonne National Laboratory, Bob was the Chair of the Machine Advisory Committee of the Advanced Photon Source (APS) project. He was very tall and always smiled. He carried a used green backpack, which surprised and impressed me – how come a world renowned scientist used such an old backpack to attend such an important meeting. Since then, I always carry a similar green backpack wherever I go. In this newsletter, we have a dedicated section in memory of Bob. Fourteen essays written by Bob's colleagues, friends and former students are published in Section 2. They describe from different perspectives Bob's remarkable career and life. The same series of essays will also appear in the December issue of the journal *Physical Review Special Topics – Accelerators and Beams* (PRST-AB).

The editor of this issue is Prof. In Soo Ko, a panel member and a senior scientist from POSTECH, South Korea. In Soo collected 10 excellent articles in the theme section "Control Systems." As In Soo pointed out, control systems are often neglected or ignored by accelerator physicists because it is a highly specialised field. These articles, all written by leading experts in this field, can serve as an introductory text to accelerator control systems. I want to thank In Soo for editing and producing a newsletter of excellent quality and great value.

1.2 From the Editor

In Soo Ko, POSTECH
Mail to: isko@postech.ac.kr

The special topic of this issue is the control system of various accelerators. The idea for this topic arose immediately after I received my editorial assignment following the ICFA Beam Dynamics panel meeting during PAC07 in Albuquerque, New Mexico. Since I have experience in both the fields of physics and controls, I believe this was a good opportunity to introduce control systems to accelerator physicists. I sometimes feel that easily neglected or forgotten are the enormous efforts hidden behind the control screens or panels in the control room.

Several months after PAC07, I attended another conference, ICALEPCS07 in Knoxville, Tennessee. This stands for "International Conference for Accelerators and Large Experimental Physics Control Systems 2007." During the conference, I began my search for potential topics and authors for this special issue of the beam dynamics panel newsletter. My pocket-sized abstract booklet became filled with my notes and comments. However, this topic idea became better defined when I discussed it with Matthias Claussen of DESY and Andy Götz of ESRF when we met during the

ICALEPCS2009 ISAC meeting at Padova, Italy last June. My original outline for the issue was the following:

- Introduction
- EPICS
- TANGO
- LHC controls must be included because this is the largest accelerator and it began its commissioning phase this past September.
- One system for one of the LHC detectors because this is why LHC was built.
- One system from nuclear physics facility such as JPARC, SNS or GSI.
- One system from a light sources facility, DIAMOND in particular
- ILC for futuristic developments
- Future trends

I asked Dave Gurd to write the issue introduction because he is one of the senior people in this field and Chair of ICALEPCS07. Perhaps I broke into his retirement peace of mind by asking him to write an article, but the excellent article he submitted reminds us of the good old analogue days of controls, and I thank him for his contribution. Authors of the next two articles are about development packages in general: Matthias Claussen is an expert for EPICS and Andy Götz for TANGO.

I thank Hermann Schmickler for accepting my proposal and submitting his article on time despite his heavy schedule and duties for the LHC turn on. From among JPARC, SNS and GSI, I have chosen JPARC simply because I needed one author from Asia, and asked Noboru Yamamoto to write an article. Unfortunately, I did not realize how busy he is. Besides his duties at JPARC, he has the additional work of being Program Chair of ICALEPCS09. So this did not allow us to have his article in this issue. Instead, the SNS system is briefly mentioned in the article by Dave Gurd. For light sources, I chose DIAMOND because my former student Jang H. Han is a staff member there. I thank Mark Heron for his contribution of DIAMOND control system. For an example of detector control systems, I asked Lennart Jirde, whom I met during ICALEPCS07, to write an article on ALICE.

I worried about my arbitrary choice of control systems. In order to avoid complaints from people working on other systems, Weiren Chou, Editor-in-Chief and I circulated an email to panel members to announce that the topic of the next issue is control systems and additional contributions are welcome. Three more articles from two institutes were submitted: one from Fermilab and two from DESY. One of the articles from DESY is about the FLASH control system. I arranged the DIAMOND and FLASH articles in sequence. The other article from DESY is paired with an article on future trends, TINE.

I thank all the authors who contributed to this issue, and hope this issue could perhaps be used as an introductory text to accelerator control systems.

While I was preparing this issue, I heard the sad news from SLAC of the death of Robert Siemann. This is a great loss of accelerator community, and we will miss him.

2 Letters to the Editor – In Memory of Robert H. Siemann

2.1 Robert H. Siemann – Encounters and Essays

Frank Zimmermann
CERN, Geneva, Switzerland, and
American Physical Society (APS), 1 Research Road, Ridge, NY 11961-2701, USA
Mail to: Frank.Zimmermann@cern.ch

2.1.1 Introduction

Sadly Prof. Robert H. Siemann has passed away on 16 September 2008. For more than three decades he had been contributing to the progress and health of accelerator physics and technology – with numerous pre-eminent research contributions, via the training of an uncountable number of graduate students and post-doctorates, as skilful science manager and dedicated conference organizer, by chairing of or serving in multiple advisory committees, and, last not least, via his decisive role in establishing and leading *Physical Review Special Topics – Accelerators and Beams* as a timely, widely circulated, international, pioneering all-electronic, and open-access journal.

This is the first in a series of essays in memory of Robert “Bob” Siemann, which is published jointly in the December-2008 issues of the ICFA Beam Dynamics Newsletter and *PRST-AB*. The essays written by Bob’s former students, colleagues, and friends review his remarkable career and life from many different angles. Their publication coincides with the 10-year anniversary of *PRST-AB*, the Editor of which Bob had been from the very start of the journal in 1998 through spring 2007.

2.1.2 Numerous Encounters

Bob Siemann and his work are omnipresent in accelerator physics. In the year 1990-91 when I was an accelerator-physics graduate student at the University of Hamburg in Germany no week seemed to pass by without a new preprint from Bob and his student Srinivas Krishnagopal on display in the DESY library, presenting recent breakthroughs in the understanding of the beam-beam interaction and its correct modelling [1,2]. Later I learnt that Bob, together with another Cornell student, Robert (Bob) Meller, had also been the first to study the shift of the coherent tune of two colliding beams with respect to the single-particle tune shift [3], which is now being referred to as the “Yokoya factor” after [4], but according to some veteran CERN colleagues could also be named the “Meller-Siemann-Yokoya factor.” Other important contributions of Bob concern the modelling of impedance-driven collective effects in storage rings and linear accelerators, and a clear and concise recipe for modelling synchrotron radiation effects in computer simulations [5]. Even nowadays, and as recently as two weeks ago, I am looking up his original papers whenever I plan to write such type of simulation.

Bob joined SLAC in 1991. In a recent article of *SLAC Today* SLAC Director Emeritus Burton Richter recalls the circumstances of this transition and Bob’s role in

advancing accelerator science [6]: "When we recruited Bob, he asked why accelerators, which were at the heart of the lab's activity, were not also part of its academic charter. He was a regular professor at Cornell and it was clear to me that he would not come here any other way... Bob not only brought us his vision of advanced accelerator science, but also a new charter that has been of great benefit to the lab ever since."

After my own arrival at SLAC as a Research Associate in mid-1993 I was assigned half of a small cubicle just in front of Bob's office. In the evenings I often saw him working with great concentration while listening to opera music. I was deeply impressed by his friendliness, dedication, and practical spirit, which set him apart.

During my early period at SLAC, Bob was strongly involved with the SLAC Linear Collider. He and his team built various types of beam instrumentation for better diagnostics, performed measurements, wrote simulation codes, and developed novel theories. Bob provided a sovereign oversight for every kind of activity. Together with Torsten Limberg he proposed and implemented a successful upgrade of the two SLC damping rings, which allowed overcoming a longitudinal instability threshold and raising the beam intensity, and so the SLC luminosity.

After the successful completion of the SLC it had been planned to publish its story in a journal. Bob Siemann, dedicated as ever, was the first to complete drafts of the sections he was put in charge of, describing the damping rings and the positron system [7]; also I myself finished a draft chapter on the final-focus system [8]. Unfortunately, some other chapters were not completed, and the project was silently abandoned.

Over the years, Bob brought me into contact with many of the students, postdocs, staff and professors working with him, in particular Michiko Minty, with whom I later taught several accelerator physics courses and jointly published a textbook; Ralph W. Assmann, who now is a great colleague and friend at CERN; Mike Seidel, a wonderful physicist whom I already knew from DESY; Tong Chen who further advanced the art of beam-beam halo simulations and is also still in contact with me today; Boris Podobodov, presently at BNL performing an outstanding service as PRST-AB referee; Ping Chou who later returned to Taiwan; David H. Whittum, arguably the most gifted accelerator physicist whom I had the privilege to meet at SLAC; Eric Colby and Mark Hogan, who by the time I left SLAC were two bright newcomers in Bob's team; and Angie Seymour, his ever-caring departmental secretary.

In 1995 Bob Siemann took on the unthankful task of chairing the joint IEEE Particle Accelerator Conference and International High Energy Accelerator Conference, PAC'95, in Dallas. The place had been chosen while the SSC had still been under construction, but in 1995 there was no longer any local laboratory to assist. As usual for him, Bob introduced a number of innovations. PAC'95 was to be the first accelerator conference with electronic proceedings, a feature which these days we are almost taking for granted, but which in 1995 posed formidable challenges. As there had been no experience, for many months after the conference I saw Bob aided by his wife spending the weekends at SLAC to reconcile all the various types of format that had been submitted. They finally succeeded and the PAC'95 proceedings [9] proved remarkably successful. PAC proceedings and those of other accelerator conferences have been published electronically ever since, and can be retrieved free of charge from the JACoW web site [10].

After the creation of his own department, SLAC ARDB, which was dedicated to advanced acceleration concepts, and though I was not a member, Bob often invited me to join ARDB social events, e.g. weekly BBQs and others, where many scientific and

physics problems would be discussed in an informal exciting atmosphere. It was great fun. Today I am dearly missing these stimulating sources of inspiration, and I regret to know that they will never again be.

Fortunately Bob's spirit seems to be alive as ever. Last Friday I had a small meeting with a few colleagues at CERN, including a visitor from the US, to discuss beam-beam effects and the agenda of a planned workshop. At this occasion one of the participants, Jean-Pierre Koutchouk, recalled the summary talk delivered by Bob Siemann at the 1989 ICFA workshop on the beam-beam interaction in Novosibirsk [11], at the end of which he stated, in a pleasant comment, that some beam-beam computer simulation codes managed to reproduce the luminosity of the collider for which they were written, but that none of them was able to correctly predict the performance of any other accelerator. The audience reacted positively, and Bob's statement still rings true today. Without Bob's guidance and experience it will be even harder to ever achieve a reliable universal simulation code.

In 2003 Norbert Holtkamp and Bob Siemann invited me to join the Accelerator Systems Advisory Committee (ASAC) for the SNS, a committee chaired by Bob. This gave me the opportunity to visit Tennessee a couple of times and there to meet Bob again, several years after I had left SLAC. Bob was as generous, cheerful and kind as always and he was obviously happy to hear about my growing family in Geneva. His professional leadership of the ASAC committee helped to complete the SNS on time and on budget, while balancing the interests within a multi-laboratory collaboration. For me it was a superb and unique experience, as well as an excellent preparation for my later service in several other similar committees.

Towards the end of my stay at SLAC in 1998, the new journal *PRST-AB* was launched [12]. I remember telling Bob my opinion that the journal better be free to authors, as otherwise it would be difficult for many of us to publish. Miraculously Bob found an ingenious and forward-looking way to make it free of charge for both authors and readers, based on sponsoring by the major American accelerator laboratories.

Several years later, in 2005 I wrote an article for the CERN Courier about open access publications and *PRST-AB* [13], which may have helped to convince CERN to become the first European sponsor of *PRST-AB* [14].

My most recent direct encounter with Bob also concerned the *PRST-AB* journal. Early in March 2007 I received an email from Bob: "I would like to talk with you in the next few days. Would it be possible to arrange a call for early Monday or Tuesday morning California time." I had no idea what his message was about. In our subsequent phone conversation Bob informed me that due to a health problem he had to stop working for a while and he therefore wanted to step down as Editor of *PRST-AB*. He asked me if I was potentially interested in the *PRST-AB* editorship. Soon thereafter I was invited to a phone interview with the APS Editor-in-Chief Gene Sprouse and several of his colleagues, and a short while later Gene indeed offered me the Editor position. The CERN management turned out to be happy too. On 16 April I received another message from Bob, "I was very pleased to hear from Gene Sprouse that you have accepted the *PRST-AB* editorship. It is a comfort to know that *PRST-AB* will be in good hands into the future, and I hope you get as much satisfaction from your editorship as I did from mine. I would like to be as much help as possible to make the transition smooth." And so he did!

Bob was a wonderful mentor for me as for many others. In much of my career I experienced his support and sympathy. It is an odd and unreal feeling that he should no longer be among us.

2.1.3 Memorial Essays

The following series of essays covers multiple facets of Bob's career and work. It contains essays describing the state of high-energy physics during Bob's early years at Cornell, SLAC and BNL; the instructive experiences of several summer and graduate students in particle physics and accelerator physics who completed their theses under Bob's supervision at Cornell or SLAC; Bob's contributions to the SLC; his launch of the ARDB department; Bob's involvement with advanced accelerator concepts, notably on laser and plasma accelerator schemes, including beam experiments at SLAC that demonstrated world-record acceleration levels; Bob's departmental leadership; his important role as chair of the SNS Accelerator Systems Advisory Committee, which has helped completing this project on time and on budget; the launch of *PRST-AB*, the innovations introduced under Bob as its Editor, and the journal's astonishing success.

2.1.4 References

1. S. Krishnagopal, R. Siemann, "Bunch Length Effects In The Beam-Beam Interaction," Phys.Rev.D41:2312, 1990.
2. S. Krishnagopal, R. Siemann, "Coherent beam-beam interactions in electron - positron colliders," Phys.Rev.Lett.67:2461-2464, 1991.
3. R. E. Meller and R. H. Siemann, "Coherent Normal Modes of Colliding Beams," IEEE Trans. NS-28, 2431 (1981).
4. K. Yokoya and H. Koiso, "Tune Shift of Coherent BeamBeam Oscillation," Particle Accelerators, 27, 181 (1990).
5. R.H. Siemann, "Computer Simulation Of Beams In High-Energy Electron Storage Rings," Proc. 12th Int. Conf. on High Energy Accelerators, pp. 271-275
6. "Bob Siemann Has Passed Away," SLAC Today, September 22, 2008
7. R.H. Siemann et al, "SLC Damping Rings" and "SLC Positron System", unpublished, 1999 and 2000.
8. F. Zimmermann, "SLC Final Focus System", unpublished, 1999.
9. American Physical Society and IEEE, Proceedings of the 16th IEEE Particle Accelerator Conference and International Conference on High-Energy Accelerators, <http://accelconf.web.cern.ch/AccelConf/p95> .
10. Joint Accelerator Conferences Web site <http://www.jacow.org>
11. I. Koop, G.M. Tumaikin (eds.), Proc. 3rd Advanced ICFA Beam Dynamics Workshop on Beam-Beam Effects in Circular Colliders, Novosibirsk 1989.
12. R. H. Siemann, "Essay: Accelerators, Beams and Physical Review Special Topics - Accelerators and Beams," PRST-AB 050003, May 2008.
13. Frank Zimmermann, "Open access in accelerator physics", CERN Courier, 18 July 2005, <http://cerncourier.com/cws/article/cern/29385/6>
14. Corrado Pettenati, Sponsoring open access: more than just wry amusement". CERN Courier, 2 November 2006, <http://cerncourier.com/cws/article/cern/29474/5>

2.2 Remembering Bob Siemann as an Early Mentor

Richard S. Galik, Cornell University, Ithaca, NY 14853, USA
Mail to: rsg1@cornell.edu

My high energy physics career started in January 1972 when I arrived at SLAC on a rainy January morning to work on a search for fractionally charged particles (quarks). I was a junior at MIT and contacts there had called Burt Richter to see if he had a January project for an undergraduate. Indeed, Burt and Bob Siemann were mounting this search using the 20 GeV spectrometer in End Station A and would be taking data during January... I was invited to participate.

Two days later we started data collection and for about three weeks I was on the midnight-to-noon shift. Bob, my mentor, typically was there for about half of each shift, and by the end of the three weeks I knew I had found a role model.

He seemed to know so much (and actually did!) about everything, from statistics to electronics to beamlines. I was (needlessly) worried that I took up too much of his time. For my part I was an enthusiastic sponge and I do not think I learned more in such a short time, before or since.

Now in the intervening years I have met lots of intelligent, knowledgeable people. What set Bob apart were his quiet, kind demeanor, his extreme patience and ability to explain, and his willingness to put so much trust in a twenty-year old neophyte. As an example, there was a magnet problem one morning and Bob and two staff people went to the spectrometer leaving novice me in the control room to remotely ramp magnets, take notes, and handle communications!

The growing respect seemed mutual and I was asked to return to SLAC that summer to help analyze the data. That gave me ample opportunity to learn more about his talents, fairness and integrity... not to mention his love of bicycling. I also got to meet Hannah, as they graciously invited me to dinner... the first of several occasions over the upcoming years.

I recall the first time I found a “peak” in a momentum spectrum. Gleefully bounding into Bob’s office I patiently had it explained to me how one “forces” such peaks by cutting on falling distributions and how one has to avoid the edges of the detector’s acceptance. No Nobel Prize, but a much wiser student.

As I would be applying to graduate schools that fall, Bob (Cornell PhD ’69) talked to me several times about my choices, emphasizing Cornell, which had until then not been on my “radar screen”. “Look at the staff listings of the major labs” he would say; indeed Cornell PhDs were everywhere. I applied; I was accepted; I chose to go. I didn’t then realize how much further that would deepen my interaction with Bob.

That following June I went to BNL for a summer program and was assigned to the Argo spectrometer group, run by Frank Turcot; it included a new staff member named... Bob Siemann. My task was to learn why the calibration of the tracking device seemed so dependent on location within the chamber. Again Bob was the perfect mentor, always supportive with constructive, pedagogic criticism, and always seeming to know how much independence and latitude I could handle. I also had my introduction to writing papers, as Bob and I put the finishing touches on the article for the SLAC quark search. Ever gracious, he arranged that I be first author.

When I arrived at Cornell in August 1973 to start graduate school I discovered that Bob had joined the faculty that same month! We worked on a multi-university Compton scattering experiment and he became my PhD committee chair. I saw why Bob wanted me to be a student at Cornell; I learned so much about all aspects of experimentation and developed a “can-do” attitude... just the traits that attracted me to him at SLAC. Along the way our relationship made the transition from mentor-student to trusted colleagues and I am certain Bob had a lot to do with my eventually joining the Cornell faculty in 1981.

So much of my early career was wrapped up with Bob Siemann that I cannot imagine how much different my life would be today had not spent that bleak January with him at End Station A. I owe him so much for all he did for me, and also for what he allowed me to do for myself. I can only hope I have passed some of his mentorship along to others.

2.3 Bob Siemann and Meson Production by Polarized Photons

Richard Talman

Laboratory of Elementary-Particle Physics, Cornell Univ., Ithaca, NY 14853, USA

Mail to: talman@mail.lepp.cornell.edu

2.3.1 Introduction

Bob Siemann's received his PhD from Cornell in 1969, working with Karl Berkelman. The title was *Wide Angle Bremsstrahlung: Energy Dependence*. After a stint as Research Associate at SLAC, he returned to Cornell as Assistant (later full) Professor in 1973. During the SLAC period Bob was morphing from elementary particle physicist to accelerator physicist. Curiously (since I was at Cornell) that is the period during which I worked most closely with Bob, and the period to which this remembrance will be limited.

This was a period when the field was still optimistic that sufficiently accurate and detailed measurement of pion and kaon scattering and production processes could clarify the strong nuclear interaction force. Vector mesons had recently been discovered and vector meson dominance investigated. More significantly, the quark model had just been introduced, and organization of mesons and baryons into SU3 families had just been understood.

During the period 1970 through 1973, Bob was mainly engaged in measuring meson photoproduction processes as a member of the Richter group at SLAC. Even limited to this brief period, Siemann's research spanned a range far too great even to be summarized here. Also, though group sizes then were minute by modern standards, it had already become futile to attempt to reconstruct who did what from the author lists of the various publications.

I will restrict myself to a solitary thread through Bob's research during this era--- *Polarization Dependence of Meson Photoproduction*. Furthermore I will emphasize one particular paper (or rather, two, one describing the set-up [1], one the measurements [2]) which is work that is unambiguously important and for which Bob was unambiguously the lynch pin; i.e. without him, the research might never have been done.

Of course many scientists were involved in the series of advances to be described here. One assumes the reader is more interested in the general flow of the field of

research and in the people involved, than in the technical details. I will, to the best of my recollections, just drop the names of the leading characters (as it happens, four would later become lab directors) along with just enough description of their contributions to support a continuous, somewhat anecdotal, narrative. An analogy to American football may be apt. Moving the ball up the field was the work of many. Bob Siemann carried the ball over the goal line.

As well as honoring Bob I hope also to make physicists nostalgic for an era in which amazing and varied advances could be made on a time scale of months, not years.

2.3.2 Polarization Dependence of Meson Photoproduction

GeV-scale photons are produced by bremsstrahlung from material targets. In 1956, Uberall [3] showed that, by using an appropriately-oriented single crystal as the target, the radiation would have quite high polarization and would have an energy spectrum far more monochromatic than the ordinary $1/\text{energy}$, bremsstrahlung spectrum. By 1968, workers at Frascati, especially Diambri [4], had demonstrated the feasibility of this method.

By 1970, Schwitters and Richter [5], and others (including Siemann) had measured polarization dependence of π^+ photoproduction using a beam polarized by the Uberall process. Their method of measurement used a subtraction procedure relying on an “edge” in the spectrum. Though adequate for these early measurements, they found one feature of the beam to be less than ideal. The edge near which the high polarization exists lies well below the upper end point of the spectrum. Photons of energy above the edge gave background counts seriously limiting the accuracy of their measurements.

By that time rho meson photoproduction from hydrogen, deuterium and other elements had been measured at Cornell by Mistry, Silverman, Talman and others [6]. In 1970, Diambri spent the year at Cornell, bringing with him the goniometer needed to produce polarized photons from an oriented diamond target. Soon we had measured the polarization dependence of rho photoproduction [7] and found it to be maximal---the rho means “remember” the photon polarization and the plane defined by the pions into which the rho decays remembers the rho polarization.

Diambri also called our attention to a paper by Cabibbo and others [8], which described an alternative method for polarizing a photon beam; namely by passing the beam through a thick, properly-oriented single crystal, copper for example. In this process, since photons polarized in one transverse plane are preferentially absorbed in the (dominant) electron-positron production process, the surviving beam develops the other polarization.

During 1970 I was on sabbatical leave in the SLAC group of Ritson (who was himself on leave elsewhere). Along with Bjorn Wiik and Dave Gustavson, we contemplated the possibility of one-upping the Richter group by using the Cabibbo process to overcome the low energy limitation mentioned earlier for the Uberall process. There is a quite serious loss of intensity in the Cabibbo process (because polarization develops only through differential absorption, like visible light passing through a Polaroid film) and the polarization is not very high. But these disadvantages were not very important for the intended purpose, and they were completely outweighed by the advantage of being able to obtain polarization at the upper end of the energy spectrum.

But we were not particularly comfortable in reciprocal space. We knew that Roy Schwitters had developed an elegant code for calculating the Uberall production process, for installation in the Richter beamline. From QED, we understood the close connection between bremsstrahlung and electron pair production, and assumed the code could be adapted for our purposes. We approached Roy about using his code, (without, it must be confessed, explaining how Gustavson intended to modify it) and he graciously approved.

By that time we were sufficiently confident of success that we had submitted a proposal to use rho photoproduction process, mentioned above, at Cornell to confirm the applicability of the Cabibbo process and to measure the degree of polarization achievable. McDaniel, the Cornell LNS director, had assigned us a few weeks of running time at the Cornell Synchrotron, some months in the future. Such proposals were far less formal in those days than they are today, but it was still a serious commitment. At that time we had no appropriate (foot long) crystalline target, but we assumed that we could acquire in time, and use, a single crystal of silicon.

Soon, to our horror, through a combination of better analytical understanding of the process and the newly modified code, we realized that the experiment was not going to work with silicon. (The Debye temperature is too low). For that matter, the original material suggested by Cabibbo, copper, would have failed for the same reason.

There is nothing like the threat of hanging to concentrate a person's mind. After a few feverish days in the Stanford library I came to realize that carbon was the only appropriate material. A foot long diamond would be ideal. But that was not in the cards. Then I hit upon graphite and, to my amazement, found that carbon exists in crystalline graphite form. Nowadays this form is known as "graphene". Scarcely a week goes by without a seminar on its remarkable properties. At the time, based on its fabrication method, this form of carbon was referred to as "pyrolytic graphite." (To this date, graphene crystals found in naturally occurring deposits have higher quality than can be produced artificially.) But the pyrolytic quality seemed adequate for the Cabibbo process.

A trip to meet A. W. Moore, of Union Carbide, in Cleveland, established that he had some graphene samples. They were shaped like mosaic tiles (few-millimeter-thick, inch-square). With the transverse beam dimension being about a millimeter, a dozen or so of these tiles could be lined up (accurately) to form the equivalent of a foot-long single crystal. They were, however, too expensive for us to purchase. Fortunately, through Moore, Union Carbide graciously allowed us to borrow the material. (Later, when Bob Siemann built a serious polarized beam at SLAC he *purchased* pyrolytic graphite from the same source---possibly including some of the very same tiles.)

With the crisis averted (and no one else even aware it had ever occurred) we completed planning for the test of the Cabibbo theory. The bremsstrahlung beam from a target internal to the Cornell synchrotron, after passing through the azimuthally rotatable graphite polarizer, passed through an amorphous carbon target in which rho mesons were produced. The rho's decayed immediately into two charged pions, which were detected. From their measured 4-momenta, the full kinematics could be reconstructed. The measurements confirmed the (appropriately-repaired) Cabibbo theory [9].

2.3.3 Construction, Characterization, and Use of the SLAC Polarized Photon Beamline

Everything up to this point has been *research*. To exploit this research for a usable accelerator facility requires *development*. As well as being a futuristic thinker, Bob Siemann also had a very practical bent. While the rest of us went on to other things, he did the serious design work needed for full implementation of the Cabibbo-polarized beamline. This work is described in the paper [1] mentioned earlier. It might be correct to identify this paper as Bob's first accelerator physics paper and it is the single most appropriate paper for pursuing more technical detail concerning the topic under discussion.

Many engineering tasks needed to be performed: construction, alignment, cooling, control system, installation of the entire apparatus in a magnetic field to sweep the produced electrons and positrons out of the beam, and so on. The precision crystal mounting was actually done by Union Carbide.

Before the beam line could be used for practical experiments, it was still necessary to determine the optimal crystal orientation, and to accurately measure the resultant degree of polarization. As mentioned already the crystal quality was fairly low, so empirical measurement was required. As well as describing the polarization process, the Cabibbo paper [8] also described how the polarization could be measured, using a second, approximately identical, crystal. To obtain the final intensity spectrum (as a function of momentum) it was also necessary, using an electron-positron pair spectrometer in a subsidiary experiment, to measure the attenuation in both polarizer and analyzer. All this is described in reference [1].

Reference [2] is the last of a sequence of papers describing polarized photon measurements performed with what could fairly be called this "Bob Siemann beamline". From this paper the full history of the group's work can be reconstructed. Much of this work could not have been done without Bob's beamline.

Preliminary results using this beamline were first reported in 1973 at the Bonn Symposium on Electron and Photon Interactions at High Energies, and I had the good fortune to be the *rapporteur* for that session. Corny thought it was, and wishing to publicize the method, I asked the Bonn lecture room apparatus attendant whether he had the sort of wire grid that one uses to demonstrate polarization of microwaves in sophomore E&M lab. The process is the same---one polarization is absorbed, the other is not. He proudly showed me, and allowed me to brandish, the very apparatus that Heinrich Hertz had used for the "same" purpose a century earlier.

Much of Bob's subsequent career has been devoted to a far more ambitious task; overcoming the 100 or so MeV per meter barrier preventing one from obtaining truly high energy electrons. Some would say Bob lived long enough to see the beginnings of success. But Bob would surely disagree. By his standards success could only be claimed after using the beam to perform a significant experiment that could not be performed any other way.

2.3.4 References

1. R. Eisele, D. Sherden, R. Siemann, C. Sinclair, D. Quinn, J. Rutherford, and M. Shupe, *A Polarized Photon Beam Produced by Coherent Pair Production in Oriented Graphite*,

- Nuclear Instruments and Methods 113, 489 (1973).
2. D. Quinn, J. Rutherford, M. Shupe, D. Sherden, R. Siemann, and C. Sinclair, *Study of Charged-Pseudoscalar-Meson Photoproduction from Hydrogen and Deuterium with 16-GeV Linearly Polarized Photons*, Phys. Rev. D20, 1553 (1979). This is the last in a series of related papers.
 3. H. Uberall, Phys. Rev. 103, 1055 (1956).
 4. G. Diambri Palazzi, Rev. Mod. Phys. 40, 611 (1968).
 5. R. Schwitters, J. Leong, D. Luckey, L. Osborne, A. Boyarski, S. Ecklund, R. Siemann, and B. Richter, *Pi+ Photoproduction from Hydrogen with Linearly Polarized Photons at 12 GeV*, Phys. Rev. Lett. 27, 120 (1971).
 6. G. McClellan, N. Mistry, P. Mostek, H. Ogren, A. Silverman, J. Swartz, and R. Talman, *Photoproduction of Rho Mesons from Complex Nuclei*, Phys. Rev. Lett. 22, 377 (1969).
 7. G. Diambri-Palazzi, G. McClellan, N. Mistry, P. Mostek, H. Ogren, J. Swartz, and R. Talman, *Photoproduction of Rho Mesons from Hydrogen and Carbon by Linearly Polarized Photons*, Phys. Rev. Lett. 25, 478 (1970).
 8. N. Cabibbo, G. Da Prato, G. De Franceschi, and U. Mosco, *New Method for Producing and Analyzing Linearly Polarized Gamma-Ray Beams*, Phys. Rev. Lett., 9, 270 (1962).
 9. C. Berger, F. McClellan, N. Mistry, H. Ogren, B. Sandler, J. Swartz, P. Walstrom, R. Anderson, D. Gustavson, J. Johnson, I. Overman, R. Talman, B. Wiik, D. Worcester, and A. Moore, *Polarization of High-Energy Photons Using Highly Oriented Graphite*, Phys. Rev. Lett. 25, 1366 (1970).

2.4 Memories of a Mentor and Friend – Robert H. Siemann

Gerald P. Jackson
Hbar Technologies, LLC
1275 W. Roosevelt Rd., Suite 130, West Chicago, IL 60185, USA
Mail to: gjackson@hbartech.com

2.4.1 Introduction

Bob Siemann was an ardent defender of the scientific method, demanding excellence from his students, peers, and himself. Given his uncommon interest in the education of the next generation of physicists, it is appropriate to remember his life through the eyes of one of his students, one who was proud to also call him a friend. There is no attempt in this retrospective to describe his technical contributions. Instead, the goal is to describe a teacher, colleague, and friend who will be missed.

2.4.2 Cornell Years

Perhaps Bob's biggest joy as a professor at Cornell was housed in two floors of Clark Hall. Referred to officially as the 410 (undergraduate) and 510 (graduate) experimental physics laboratories, this collection of experimental physics apparatus covered the gamut from RF engineering to optical pumping to Mössbauer spectroscopy. To many students, those two floors may have been a medieval torture chamber, with the chief antagonist being Bob himself. Because the 510 lab was the only required course for graduate students, and because it was the first time many students were exposed to the rigors of real-time lab notebook maintenance, precise error analysis, and peer-review quality scientific paper preparation, Bob's demanding standards and frank feedback garnered him a reputation reminiscent of Professor Kingsfield in "The Paper

Chase” played by actor John Houseman. Bob’s physical height only reinforced the intimidation some student felt in his presence.

In 1981, my first year of graduate school, Bob was assigned as my faculty advisor. In the autumn of 1982 he agreed to be my thesis advisor. To my great amusement, graduate student peers were shocked by my choice, asking why I would want to “take the hard road” to a doctorate degree. Bob’s other graduate student at the time, Robert Meller, was also aware of this reputation. One day, noting that students were backing against the corridor walls while Bob walked past, Robert and I started emulating the breathing sounds of Darth Vader while walking behind Bob. The horror on the faces of the other students in the hallway, our laughter, and the look of amusement on Bob’s face as we let him in on the joke, were priceless.

In 1982 Bob was working on a computer simulation of longitudinal beam instabilities caused by wakefields in the CESR RF cavities. The RF engineering setup in the 510 lab, the method of imposing beam instabilities in the simulation code, and many of our joint experimental tasks on the CESR collider were heavily influenced by Bob’s love a problem solving in the frequency domain. One of the experimental problems we worked on together was the determination of the bunch crossing point within the CLEO particle physics detector. There was a beam position monitor on either side of the detector, but at the time the use of oscilloscopes and standard 50 Ω coaxial cable made time-domain determination of the crossing point too uncertain given the small vertical beta-star at the collision point and the extremely short electron and positron bunch lengths. In what turned out to be an extremely sensitive method, we monitored each beam position monitor with a spectrum analyzer, using the modulation of the beam signal spectrum to accurately determine the relative arrival time of a single positron and electron bunch in collision.

The CESR professors, on both the detector and accelerator side, had a spirit or philosophy that expressed itself in such pioneers as Robert Wilson, Boyce McDaniel, Helen Edwards, and Bob himself. No detail was too minor, no task was too menial, no assignment was ever without a learning experience and some level of adventure. My first service job at CESR was to repair a charging power supply for a kicker magnet. Bob pointed it out to me in the tunnel (under the main dipole magnets and next to the main electrical buss that was always uninsulated and active), handed me the electrical drawing of the 3-phase circuits, and told me to be careful after telling him that I never worked on 3-phase circuits before. Though there were several instances of sparks, in the end the experience reinforced the lesson that anything is possible if you put your mind to it, along with a little sweat.

There was also a sense of joy at the privilege of being a physicist and spending your life on something that you truly love. It was inspiring to watch lab director Boyce McDaniel down in the basement winding copper coils for magnets, to work with Bob as he fearlessly opened decades-old control circuits for the synchrotron to make some repair, and to be honored with the expectation to exceed these role models.

One important lesson was to never let something petty stop you from doing your job. One of Bob’s lectures involved doors locked by the administration, especially one silly attempt to lock the CESR store room during the night. While motivated by an attempt to control costs and monitor inventory, the effect was to frustrate students and professors who were making repairs or performing research at night, when the most progress was usually made. The specific solution in this case was to inject Elmer’s Glue into the lock tumblers, following the admonition “if we physicists cannot get in, then no

one is going to get in". After a few more attempts by the administration, physicists were never again locked out. Though it is doubtful that Bob was an actual participant in this specific instance, he certainly did not discourage such behavior.

Because they loved their lives as physicists, it was not unusual to find the Cornell professors in their offices or at CESR at any time of the day or night. Bob was no exception. At one point it became a point of competition to see which of the two of us would be the last to leave and the first to arrive the next morning. Despite the distance to his ranch and the fact he commuted via bicycle, he always worked longer hours.

2.4.3 Fermilab Sabbatical

Bob came to Fermilab on sabbatical in 1987 after attempting to succeed retiring director Boyce McDaniel, in competition with his own thesis advisor Karl Berkelman. Though an unusual location for an avid sailor, he enjoyed prairies and the challenge of the technical issues. Bob's presence was most appreciated during the completion of my thesis while a Fermilab employee (starting an unfortunate Cornell pre-doc tradition that is thankfully now ended). Bob used his frequency domain insights to make significant contributions in the area of beam instrumentation, notably the key horizontal and vertical betatron tune detectors called Schottky monitors. Though misnamed, since there was so much coherent signal (and associated emittance growth) in the Tevatron Collider proton and antiproton beams, these detectors were the key tools used to increase luminosity lifetime and decrease background rates in the particle physics detectors.

Another key contribution was a circuit to monitor bunch lengths in real time using logarithmic amplifiers and a novel Analog Devices chip whose output voltage was proportional to the square root of the input voltage. Since the Fourier transform of a Gaussian bunch has a spectral shape composed of individual peaks at harmonics of the RF frequency, using these chips to take the square root of the logarithm of the amplitude of one peak produced an output voltage that was proportional to the RMS bunch length. Because of the properties of these chips, a properly normalized circuit could have a bandwidth of 10 kHz. This was very important in the early Tevatron Collider days where longitudinal bunch processes such as bunch rotation for antiproton production and beam coalescing for collider bunch formation were suffering from instabilities and beam loading issues.

More importantly, his insights and leadership in these projects inspired engineers, technicians, and students. Years later, when I became leader of the accelerator instrumentation department, these same people led and supported accelerator improvement projects that dramatically increased the luminosity performance of the Tevatron Collider. The same Cornell spirit that made CESR such a great place for graduate students, which had already been instilled at Fermilab by its founder Robert Wilson and reinforced by Helen Edwards, was further nurtured by Bob during that time. The fact that the Tevatron Collider currently runs at 300x its design luminosity is a testimony to this spirit.

2.4.4 Lasting Mentor and Friend

Over the years after Bob's Fermilab sabbatical, he continued to be a mentor and friend. We served jointly on various committees for the APS DPF and with Melvin Month on the U.S. Particle Accelerator School (USPAS) and various international

accelerator physics schools. He led one of the key committees that steered the Superconducting Super Collider (SSC) away from Fermilab towards Texas, a report whose assumptions were so controversial that all copies were confiscated. That experience, which violated Bob's dedication to open science policy, soured his enthusiasm for further government committee service. Years later, in an attempt to help Bill Foster with his proposed Very Large Hadron Collider (VLHC) or Pipetron project, he quietly forwarded a personal copy of the report that he had secretly preserved.

In 1995 Bob was the program chairperson for the particle accelerator conference. This was the first year in which electronic submission of papers was attempted. Picked to serve on his program committee, I witnessed firsthand how frustrating and difficult that task would become. Instead of letting the editing process drag out over the months after the conference, he insisted on working with the authors at the conference to fix each and every paper. Despite the lack of adequate software tools and templates, the experiment was a big success. In concert with the proposal of web posting by Ilan Ben-Zvi, these two scientists were the pioneers of the accelerator community effort that became the JACoW website, where all accelerator conference proceedings are stored. Bob also championed the suggestion to archive earlier conference proceedings, leading to the electronic archival project by Hbar Technologies, LLC that continues to this day with work on the Cyclotron conferences. It was this early experience that undoubtedly fired Bob's vision of an accelerator physics journal that was freely accessible on the web: Physical Review Special Topics – Accelerators and Beams.

In 2003 Bob was the conference chairperson for the U.S. particle accelerator conference held in Portland, Oregon. I had already left Fermilab and had just been through multiple lawsuits and financial travails with the newly formed Hbar Technologies, LLC. Because of his lofty position, the conference hotel had given him the presidential suite that contained two separate bedrooms. On his own, he called and asked if I wanted to share the suite with him. During the course of the conference, he routinely invited graduate students and post-docs up to the room for late night parties. Some of these post-docs were graduate students of my own who he hired to work with him at Stanford.

This was the quintessential Bob Siemann, someone more at home hanging out with young physicists, reinforcing collaborations and mutual friendships, than serving on committees and socializing with the power brokers in the accelerator and particle physics communities. And it is for precisely this reason that he will be so sorely missed by so many.

2.5 Robert H. Siemann: A Personal Tribute

Srinivas Krishnagopal, BARC, India

Mail to: skrishna@tifr.res.in

On 18 September 2008 I received a short email from Miguel Furman, which came as a huge shock: Bob Siemann had passed away.

I knew Bob since 1988, which I started working with him as a graduate student. Earlier, in the fall of 1985, I joined Cornell University, and after three semesters of course-work, started looking for a research topic in accelerator physics. For around a year I worked on different things, but nothing clicked. During that time I only knew of Bob as a name-plate on a locked door, since he was mostly away at Fermilab. But, by

early 1988 he was back, and I approached him. He gave me a copy of a proposal he was working on; I still have the copy, dated 11 January 1988. It was twenty-five pages of stuff I couldn't fully comprehend, but it looked exciting, and I was interested. Besides, by that time I had gotten over the joys of long Ithaca winters, and didn't think I could take many more years of them. So I was less interested in the specifics of the proposal, and more interested in graduating in five years! When I went back to Bob's office I told him this. He looked at me somberly and said, "You know, that leaves you less than three years to write a thesis. You can do it, but it'll mean a lot of hard work." I started working with Bob, and finished my thesis in 5 years and 3 months.

When I look back now, I'm amazed at my gall, but even more so at Bob's positive response. If a student came up to me today and insisted that he'd like to complete his thesis in less than three years, I'd probably launch into a three-hour lecture on the uncertainties of research and the virtues of patience! Bob's attitude was just to treat that as a constraint and push on, without obsessing about the time-frame, and that was absolutely the best way to motivate a student to work hard and graduate. Another major motivating factor was that he worked very hard himself. As far as I could see, he worked from 9 to 6, six days a week, and while I'm sure he took vacations, I can't remember him being away for too long.

One of the things I enjoyed most was his accessibility. I could barge into his office at any time, and, irrespective of what he was doing, he'd put his work aside and give me a patient and unhurried hearing. Friends of mine had thesis advisors who were far less accessible; some had to get appointments from the secretary and others communicated through notes. This was particularly in evidence when we were working on bunch-length effects in the beam-beam interaction. Bob asked me to write a code that included the finite bunch-length of the beam (as opposed to the standard treatments that assumed a 'pancake' beam). The thinking was that bunch-length, through the hour-glass effect, and perhaps through more detailed dynamics, was bad for the beam. But my simulations showed that, for some values of the bunch-length, the situation actually improved – it gave a higher tune-shift! At first Bob was convinced that I must have made a mistake. He went thorough the code line by line, finding numerous mistakes (including some embarrassingly elementary ones). I corrected them, and the numbers changed, but the basic result did not. He was then convinced that the phenomenon was real, and one needed to find an explanation. He asked me to try some analytic calculations with a simplified model, to understand things better. In this context I would barge in on him many times a day, with a new problem or a brilliant new solution. Each time he'd listen to me patiently, point out obvious errors gently, discuss more subtle points avidly, and keep pushing me down the path. At that point I realised that he was a great teacher, who put his graduate students first. I was wrong.

Around the same time, Bob was teaching an undergraduate course in physics. One day I walked into his office to find him busy with a young student. He asked me to come back later. I was rather annoyed, since I'd gotten used to having first dibs on his time. After this happened a couple more times, I asked him querulously who the student was. It turned out she was taking his course, and wasn't finding it too easy, so she came to him for a little extra help. It was a very trying semester for me, but it finally came to an end, the student disappeared, and my exclusivity returned. Imagine my consternation, therefore, when, one fine afternoon the next semester, I found the same student there again! After she left, I walked into Bob's room and asked him rather sharply: 'Are you teaching again this semester? I thought you only have to teach one course a year?' Bob

smiled and explained. Apparently, the student had found his tutoring very helpful the previous semester. This semester she was taking another physics course (not taught by Bob), which she was finding difficult, so she wanted Bob to tutor her again! Bob could be demanding of his students, but he also put his students, graduate and under-graduate, first. His style of mentoring was never didactic, and therefore more effective.

My deadline of 5 years was fast approaching. After the work on bunch-length effects, we'd progressed to looking at collective beam-beam instabilities. After a tense intermediate period, when we seemed to be making no progress, things panned out well. I had worked hard, been lucky, and had enough to write up a thesis. But I'd heard stories of drafts of chapters returning from Bob's room bathed in red ink! He had high standards of scientific prose, and was ruthless in implementing them. Someone (perhaps Gerry Jackson?) warned me that it could take anywhere between six months to never for Bob to approve a thesis. So I worked very hard on the writing part too, and rewrote many drafts before submitting anything to Bob, all of which he'd get back to me in a matter of days. I was lucky. My chapters came back with only modest amounts of red, and were typically okayed on the second iteration. Many years later he told me that the theses that gave him the least trouble were Bob Meller's and mine. It's a compliment I still treasure.

In 1991 Bob moved from Cornell to SLAC. In that same year I graduated and moved to LBL. The relatively close physical proximity made it easier to continue to collaborate on the beam-beam interaction. We generalized our work on collective beam-beam instabilities, started for round beams, to more general beam-profiles. The most natural way to do that was to work in cylindrical coordinates. However, to simulate flat beams, it would be better to work in Cartesian coordinates. Though Bob's interests were already shifting to advanced accelerator concepts, he encouraged me to work on this. Before I could make much progress, I moved back to India at the end of 1992, but continued to work on the problem only because of strong encouragement from Bob. At one point I needed to do a check, for nearly-round beams, with our old code, which I couldn't run because I didn't have the requisite scientific library. I asked Bob, and he had the results faxed to me within a week.

Over the years we continued to keep in touch, and on my occasional trips to the US, I would make it a point to visit him at SLAC. Remarkably, I found almost no change in him. His dedication and enthusiasm for physics remained unabated, as did the joy and satisfaction he derived from remaining a 'hands-on' physicist. As he and his group worked on new acceleration methods, and produced trail-blazing results on plasma wakefield acceleration, it was a privilege to hear of the latest results and future plans first-hand from Bob. As always, he believed strongly in mentoring young students, and took great pride in the large fraction of students in the group, and in their contributions and achievements, both, then, and later in life.

Bob Siemann's contributions to our field are many, deep, and diverse, but perhaps none more important than his mentoring of young scientists. Those of us who were fortunate enough to be his students will miss him; the best tribute we can pay to him is to follow his example with coming generations of scientists. In addition, I hope institutions such as the American Physical Society will commemorate his contributions in this area, perhaps by instituting a student fellowship or prize in his name.

2.6 In Memory of Robert Siemann

Alexander W. Chao, SLAC
Mail to: achao@slac.stanford.edu

Bob Siemann and I first met 32 years ago at a Cornell workshop. The workshop subject was Collective Instabilities. He was a young assistant professor --- energetic, confident, full of ideas, and I was a postdoc. Since then we both had gone our own paths, but curiously from the hindsight, it seemed that I had been trying to follow his footsteps, or at least it might have been the case. Those footsteps included activities in HEPAP, in the APS Division of Physics of Beams, and in the Physical Review journals. It seemed that Bob was a perfect role model, although I will be the first to say that I did not fit those roles nearly as well. He was also the driving force that brought me back to SLAC – again following his footstep – after the demise of the SSC. Although unintentional at the time, I must admit that I draw much pride in finding that my priorities and my values are apparently not dissimilar to those of Bob's. This report will be a personal tribute in memory of Bob.

I mentioned the SSC, the Superconducting Super Collider. Let me begin there. During the SSC days at Berkeley, after the Conceptual Design Report was issued in 1985, Bob was appointed by the DOE to chair a review of the accelerator physics part of the CDR, which I was responsible for. The main issue was the SSC magnet aperture. I defended the design of a 4-cm aperture, while he led the questioning of whether that aperture was too small. Our views were confrontational, to say the least. The aperture issue remained after the review, and eventually it was changed to 5-cm four years later by another SSC management. But throughout the review in 1985, as difficult as it was, never in the heated debates had we lost respect and trust of each other. Bob and I were close friends before, during, and after the encounter. I am confident that this was Bob's view too.

Bob Siemann came to SLAC from Cornell in 1991. With the support from Burton Richter, the then Director of SLAC, he took on a leadership role to formulate an academic program in accelerator physics at SLAC and the development of its accelerator faculty. Throughout his career he championed accelerator physics as an independent academic discipline, a vision that he fought so hard for and never retreated from. He convinced Stanford University and SLAC to create a line of tenured accelerator physics faculty and over the years he also regularly taught classes at Stanford and the US Particle Accelerator School. After the shut down of the SSC Laboratory, I returned to SLAC in 1993 to join the accelerator faculty he was forming. He had always visualized a need to have a professional academic journal for the accelerator field, and played a pivotal role in creating the journal Physical Review Special Topic on Accelerators and Beams, now the community standard for accelerator physics after nine years of his editorship. Today, Bob's legacy of accelerator physics as an independent academic discipline continues at SLAC as well as in the community, from which we all benefit.

Bob fell ill in June 2006. He had not said much to me about that until March 2007 when he came to talk to me. He told me that he had leukaemia. The treatment so far had been successful, but that was only a temporary solution. A long-term solution would be a risky procedure of bone marrow stem-cell transplant. For that he would stay home for

three months, and he might also not make it at the end of the procedure. Two things, he said, were on his mind, and he wanted my help in case the worst happened. First thing he wanted was for me to take care of two of his five students. The second thing was that he wanted me to be one of the three candidates to take over the PRSTAB editorship from him. It was a touching moment when he talked about things closest to his heart. I chatted with him and agreed to his first request. A few weeks later, with much difficulty on my part, I hesitantly declined his second request. Fortunately, a few months later, in June 2007, he told me that he was doing fairly well and could return to work. He said that he really missed the joy of visiting his dusty laboratory and doing experiments with his students. Also happily, I found later that an excellent candidate, Frank Zimmermann, was chosen for PRSTAB editor and that Frank had accepted.

Bob's illness made a bad turn in September. The worse news arrived when I was in the middle of a meeting in Beijing. It was the saddest day for a long time and it took me quite an effort to reassemble myself back to the meeting and to recover from the shock.

Bob was a great experimentalist. He specialized in experimental techniques and instrumentation, but what he wanted to learn is physics. If he had to learn theory -- heaven forbid -- to reach that goal, he would not hesitate one second to do so. In fact, he had written several theoretical papers as results of these efforts. Now this is what I call a true experimentalist! Ultimately, however, I think it was experimental instruments that he loved most. His eyes widened when he talked about his instruments. Prompted by a question, he would proceed to a nearby blackboard, with a satisfying grin, and draw his experimental device in a careful thinking manner, then describe his experiment and educate the questioner with some insightful physics. These moments were most enjoyable, to him and the questioner alike. When I think of Bob today, it is these moments that first come to mind, and it is these moments I will miss the most.

I should like to mention another curious thing about Bob, namely he had a special talent of finding persuasive arguments that went his way. It was difficult to argue with Bob because it was so difficult to win. Generally quiet otherwise, he was too good and too methodical a debater. I had never seen him losing a debate on a policy issue or in a committee setting. However, when it comes to physics, his soft spot, he occasionally let go some weakness. When so doing, he would lose the debate, but his grin revealed that the loss was more than compensated by the physics he gained together with his debater.

It is hard to believe that the office around the corner is now empty. The dear colleague we have come to know, to talk to, and to seek advice from, together with the feet-on-the-desk posture and the familiar grin, are no longer there. I wonder, who will now occupy that office next? And who will continue to carry on Bob Sieman's legacy? Many of us are waiting.

2.7 SLC Days at SLAC

Tor Raubenheimer, SLAC
Mail to: tor@slac.stanford.edu

Bob Siemann was a great experimentalist and an excellent teacher. We will greatly miss him. Bob came to SLAC in early 1991 to work on the Stanford Linear Collider (SLC). The SLC was a challenging accelerator which began operating in the late 1980's but still had numerous obstacles to be overcome years into operation. One of the compounding difficulties was making reproducible measurements, since the stability of

the collider was poor and the diagnostics were insufficient. Bob dove into this challenge and helped design experiments and diagnostics that provided further clarity.

I first got to know Bob while I was still a graduate student, trying to finish my thesis and performing some experimental studies on the SLC, which, at the time, was proving to be very difficult. Most of my expertise had been in beam theory and simulation. Dealing with the real issues of the accelerator was challenging. Bob helped me understand the difference between systematic and statistical errors, and separate operational issues from the fundamental physics. His way of teaching was not to provide an explanation but to ask enough questions so that I could find the answer on my own – this was the best way to learn. I later asked Bob to be a reader on my thesis. As in all things, he took this role extremely seriously. He read through the draft and marked every page to the point where I was regretting my decision. However, his questions again helped me understand my own work better and greatly improved my thesis.

Bob was also the defacto leader of an effort focused on the damping rings and the bunch compressors. He was great to work with. He made people think for themselves and refused to simply provide answers. He also worked hard himself, expressing real interest and curiosity. After the studies of the SLC damping rings identified a sawtooth instability due to the vacuum chamber impedance as a source of many downstream fluctuations, Bob took charge of upgrading the rings. As part of this program, I suggested an extensive upgrade that also replaced the dipoles with combined function magnets which might have reduced the horizontal emittance another factor of three. Although he was extremely busy, Bob helped me develop the proposal and understand the magnetic limitations as well as the potential impacts on the beam dynamics. He helped me consider issues well beyond my initial scope. While the proposal never went anywhere and I think Bob had been aware that there was no funding to pursue the option, he saw that it would be a great learning experience for me and it was.

In the early 1990's I had simulated a new regime for the beam-ion instability and, with Frank Zimmermann, I developed a model for the effect which was predicted to occur within the high current, low emittance bunch trains in future storage rings or linear colliders. I thought this was pretty good work but Bob convinced me that the next step had to be confirming the theory with measurements. Because the growth rate was inversely dependant on beam sizes and proportional to the vacuum pressure, measurements required significantly increasing the vacuum pressure in existing facilities. Most people discounted trying such an experiment, but with Bob's urging and suggestions and John Byrd's excitement, we managed to make the measurements at the ALS at Berkeley in 1995.

By the mid-1990's Bob was completely focused on advanced acceleration concepts. At the time, SLAC was putting together a large effort in designing and documenting a design for the Next Linear Collider (NLC) while constructing the NLC Test Accelerator. Bob was worried that such a straightforward extrapolation of the microwave technology would be difficult to bring to fruition because of the cost. He wanted to focus on more cost-effective approaches. As usual, he was correct. The experimental programs that he started in direct laser acceleration and plasma-wakefield acceleration have made great progress. He accomplished this with lots of hard work and by engaging the people around him, especially students and postdocs. In the process, he created a group of extremely talented people which has enabled these

technologies to be developed to the point where it seems that they will likely offer two cost-effective approaches to high-gradient acceleration.

Sometimes Bob was gruff, and sometimes he was excited but, in all of my interactions with him, he modelled intellectual integrity. Sometimes we disagreed but I always trusted him. He pushed people to think for themselves, but would provide the guidance by asking the right questions. He loved experimental physics and loved designing experiments to get at the physics. We will all greatly miss him.

2.8 Essay for Robert H. Siemann

Ralph W. Aßmann, CERN
Mail to: ralph.assmann@cern.ch

I first met Bob Siemann in January 1994, during the LEP performance workshop in Chamonix in France. I wanted to leave Europe for a job as post-doc in the United and I still remember walking to this tall and impressive professor from the mythical Stanford, feeling quite nervous. Bob immediately put his full focus on me and listened carefully. Without promising anything (the SSC was just shutting down) he told me about the work they do at SLAC and the opportunities for young researchers. His quiet and attentive voice and careful argumentation impressed me. Immediately I was convinced that I should go to SLAC, if possible. Bob Siemann truly liked to work with young people and young people liked to work with him. Over the years he guided almost 20 graduate students to their degree.

In November 1994 I arrived with my wife in California. I had never visited California or SLAC before nor did I meet Bob a second time. I felt at home quickly. Bob Siemann's office was always open for discussion and he would always take time. He loved talking technical issues and we could spend hours. Never would he waste much time with "political" discussions, quickly classifying non-scientific arguments as "wishful thinking" and that was it. Bob did not care how work would be done, as long as it was done and done well. I still remember that I wanted to spend less time in the office in 1996, as my first son was born. Asking Bob whether it would be OK, he only said "sure."

The year 1996 brought important changes for our work. Bob was appointed head of the new advanced accelerator research department ARDB. Bob was the leader, Angie Seymour the soul and I got my first SLAC office with a window. This new department had about 10 members. We compensated the small size with our energy and managed to move quite fast. For example, we got laboratory rooms for ARDB, which were in quite bad shape. As there was no money for remodeling, Bob organized some paint and we painted the lab ourselves. We had a picnic in a park afterwards, Bob's wife Hannah bringing food and drinks. I still remember that Bob got quite worried about the worker's union after our remodeling work. Apparently it was not allowed that physicists would paint laboratories themselves and thus would take work away from the union's workers. Luckily, nobody complained. A lot of work and love went into equipping the various ARDB laboratories, everybody looking for useful equipment and placing the ARDB sticker on it...

The Snowmass 1996 meeting brought an additional important turn for our work. The SLAC hope had been to proceed with the Next Linear Collider project. However, already on the first day I met a scientist telling me, "the future is bleak". It turned out

that this person was right as far as the Next Linear Collider was concerned. Bob Siemann in the meanwhile followed his vision on high-gradient acceleration. He met the plasma-acceleration physicists from Los Angeles and got excited about the possibility to perform an experiment at SLAC. He asked me to come along and immediately I was excited as well. Over the next months we would pursue this idea with Tom Katsouleas and Chan Joshi from USC and UCLA. We produced the written proposal and came up with the 1 million \$, 1 GeV in 1 m experiment. This would become E-157, the first in a series of SLAC experiments on plasma wakefield acceleration. Bob trusted me with many important aspects, taking me to all relevant meetings and giving me the first budget responsibility in my career. Bob asked me to be co-primary investigator and to present the SLAC part to the Experimental Program Advisory Committee of SLAC. After the talk Burton Richter came to me, telling us what a stupid idea this experiment was. Why wouldn't we ask for a big experiment? We got somewhat worried but the experiment E-157 got approved. And E-157 got bigger with time, spawning several bigger follow-up experiments. Bob took many shifts on the experiments and also took over responsibility for the data analysis tools, programming a very powerful Matlab code.

Bob Siemann's thinking is quite well illustrated by his June 2001 e-mail, announcing that the proposed ORION Center for Advanced Accelerator and Beam Physics would not be funded. He was "concerned about the ARDB research program itself and obligations to people who are vital to this research". Research and obligations to people were equally on his mind. This made it such a pleasure and so rewarding to work for Bob. We met about once per year after I left SLAC in 1998, always finding time for lunch or common dinner. Bob would always keep me up-to-date about the progress in advanced acceleration research, the area that he contributed pushing forward so much. The future will certainly prove the practical importance of the groundbreaking work performed under Bob's leadership at SLAC. He was a truly great scientist, teacher, mentor and friend to many researchers in the field. I owe him a lot. I want to end this essay with a quote from Bob's 2001 e-mail on ORION: "What I do know is that our work is too important to let this be more than a setback that will be overcome." What an excellent legacy to always keep in mind for accelerator R&D. Farewell Bob!

2.9 Robert H. Siemann as Leader of AARD

Eric R. Colby and Mark J. Hogan, SLAC

Mail to: ecolby@slac.stanford.edu

2.9.1 Introduction

Robert H. Siemann originally conceived of the Advanced Accelerator Research Department as an academic, experimental group dedicated to probing the technical limitations of accelerators while providing excellent educational opportunities for young scientists. The early years of the Accelerator Research Department – B, as it was then known, were dedicated to a wealth of mostly student-led experiments to examine the promise of advanced accelerator techniques. High-gradient techniques including millimeter-wave rf acceleration, beam-driven plasma acceleration, and direct laser acceleration were pursued, including tests of materials under rf pulsed heating and short-pulse laser radiation, to establish the ultimate limitations on gradient. As the

department and program grew, so did the motivation to found an accelerator research center that brought experimentalists together in a test facility environment to conduct a broad range of experiments. The Final Focus Test Beam and later the Nest Linear Collider Test Accelerator provided unique experimental facilities for AARD staff and collaborators to carry out advanced accelerator experiments. Throughout the evolution of this dynamic program, Bob maintained a department atmosphere and culture more reminiscent of a university research group than a national laboratory department. His exceptional ability to balance multiple roles as scientist, professor, and administrator enabled the creation and preservation of an environment that fostered technical innovation and scholarship.

2.9.2 Scientist, Colleague, and Collaborator

Bob was a scientist of exceptionally high standards and uncompromising integrity. He combined intellectual rigor with a strong, hands-on work ethic. Bob believed deeply in the value of scientific collaborations – especially between universities and national laboratories. He believed successful collaborations leveraged the unique, state of the art facilities and expertise available at national laboratories with the broad range of scientific disciplines and academic rigor found in the university environment. As a SLAC professor, Bob managed to simultaneously thrive in both worlds.

The poker expression ‘all in’ summarizes Bob’s approach to his professional collaborations. Bob took his responsibilities as a collaborator very seriously and participated fully in all aspects of experiments from writing the proposals, planning and executing the experiments, analyzing the data, and writing-up and presenting results. No result was too small to escape his full scrutiny and understanding. Anyone who ever submitted a conference paper without his review will have the scars to prove it!

2.9.3 Professor

Bob approached accelerator physics with the same intellectual rigor as any other field of scientific inquiry. He understood that the farthest-reaching innovations in accelerator science were most likely to be spawned in an academic atmosphere founded on thorough inquiry and painstaking documentation. The founding of this scholarly journal is perhaps the most emblematic example of his dedication to his field as a true science.

He taught numerous courses at Stanford University, writing out his lectures longhand and preferring traditional delivery at the chalkboard to computer presentation. The resulting discipline of mind—of being able to think on the feet from first principles—served Bob well in many arenas. He was revered by his students for his clear physics insight, patient guidance, and flexible style. Although uncompromising in his standards of academic scholarship and intellectual rigor, he was surprisingly accommodating when relating to his students. He would relentlessly push students to work 12-hour days (a “short workday”) and to repeatedly revise papers until the science was accurate and the language precise, but would listen carefully and provide gentle guidance when students sought his counsel.

2.9.4 Mentor

At 6'7", Bob was an imposing presence physically, yet he fostered a nurturing academic environment in AARD that was loved by staff and students alike. As a mentor he was generous to a fault with his time and was always willing to discuss an idea, work through a problem or just listen to a complaint. When working, Bob always led from the front. He would never ask anyone to do something that he himself was not already doing or willing to do – regardless of official titles. Bob believed in getting the job done and would frequently work late into the evening or on weekends and holidays if required. While Bob was capable of doing many things himself, he also understood that part of mentoring is holding back to create room for young people to grow and advance in their careers. Growth requires new opportunities to succeed or to fail, and the support to re-group and try again. Bob did not insist on perfection, only that people give their best effort and not give up.

Bob showed great pride in his work and that of his colleagues, yet he did not seek to take the limelight away from people coming up under him. Bob encouraged people to take risks and bring up new ideas. He fostered a learning environment where people felt comfortable trying new things and taking on additional responsibilities. As a physicist, Bob was equally at home discussing non-linear dynamics or patiently reminding people of the virtues of 50 ohms. He enjoyed the 'how' as much as the 'why' when solving problems. Bob's depth of knowledge served as a constant reminder that there was always so much more to learn. Despite his high standards and uncompromising personality, Bob engendered tremendous respect from both colleagues and competitors alike. The care and pride that Bob showed in his teaching left an indelible impression on all of the students and staff fortunate enough to receive his rigorous and enthusiastic mentorship.

2.9.5 Administrator

Bob established and preserved an academic atmosphere within his department, a challenge to do in a national laboratory environment. Part of this was a natural result of the focus on long-term research, but a significant part was due to his ability to balance multiple roles and shield his students and staff from some of the demands that are normal at a large laboratory.

2.9.6 Hobbyist

Saturdays were often Bob's time to enjoy himself in the lab. His favorite activities included designing and building custom electronic circuitry and assembling first-class instrumentation hardware and software. It was often joked that learning to construct electronic circuitry by the wire-wrap method was a graduation requirement for his students. Over the years he constructed a wealth of analog and digital circuits to control and instrument the experiments. The construction of a w-band vector network analyzer from an array of laboratory instruments was among the most ambitious projects he undertook. Sophisticated rf and lock-in detection techniques were required to achieve more than 60 dB dynamic range over the 75-110 GHz range. Bob also wrote a wealth of LabVIEW software that integrated the equipment into a powerful scientific instrument.

2.9.7 Conclusion

“Success is not the key to happiness. Happiness is the key to success. If you love what you are doing, you will be successful” – Albert Schweitzer.

Bob was tremendously successful by such a measure and he instilled his passion for all aspects of his professional life into the many students, staff and colleagues fortunate enough to have worked with him. Bob’s legacy carries on today in the many lives he touched as well as with initiatives to build a new facility to continue plasma wakefield experiments, and the continuation of efforts to produce laser-powered accelerators.

He will be missed.

2.10 Siemann Tribute

Tom Katsouleas, Duke University
Mail to: tom.katsouleas@duke.edu

I was fortunate to get to know and work closely with Bob Siemann over nearly a decade of exciting collaborative work. It was at the Snowmass meeting in 1996 that the Plasma Wakefield Accelerator collaboration was born. The laser gas jet experiment at Rutherford Appleton Laboratory had recently produced electrons over a 100 MeV in a mm scale gas jet and given a considerable boost to the small plasma accelerator community. Chan Joshi and I were thinking of the next milestone and believed that the most direct path to demonstrating a GeV energy gain in a plasma would be with a high energy electron beam driver, and of course, there was only one of those – at SLAC. I shared this thought and a quick 2-D simulation with Bob Siemann at Snowmass. The simulation showed that the SLAC beam parameters could be expected to produce a wake of amplitude 1 GeV/m in a 10^{14}cm^{-3} density plasma. I did not know Bob well at the time and I was a bit apprehensive about how he would react. He explained to me how his group at SLAC, ARD-B, was charged with developing concepts capable of demonstrating a GeV energy gain in a meter. He was immediately receptive to the idea and brought me in touch with a “talented young postdoc” named Ralph Assmann whom he thought would be the perfect SLAC point person to work with me on developing a proposal for an experimental test.

This was the beginning of a collaboration that would be rich and full of the kind of rewards only those of us who are privileged to pursue a life of science are able to experience. Ralph and I worked on the proposal together along with Bob, Chan and Warren Mori at UCLA and David Whittum at SLAC. Later we were joined by Wim Leemans and his group at LBL. We learned much about each other’s scientific language, culture and the realistic limitations of the beam and the plasma we would soon marry. The experiment was approved and Bob would invite me to sabbatical in 1999 at SLAC to participate in the first runs.

I remember the week before the first run when Bob came in and handed me a blank sheet of paper and said “Do you want to make up a shift schedule?” My face was probably as blank as the paper. So he said, “We are going to operate 24 hours a day for the next 3 weeks.” At that point, I laughed heartily at his joke. Then I found out that Bob was not joking. Having come from a plasma culture, I did not have the appreciation for the precious nature of beam time. Although by this time we had been joined by the two talented young experimentalists Mark Hogan from SLAC and Patric Muggli from

USC as well as others from Wim, Chan and Bob's groups, we were still a spare group to run shifts. It was then I began to appreciate as well Bob's work ethic and how driven he was. He was a hands-on role model for the group, having written most of the post-processing code to do data analysis. He also took full shift duty, despite many outside responsibilities associated with running a journal, a group and a major conference. Sometimes his intensity could barely be contained in the small trailer next to the FFTB that was our data room and control center. But his drive brought much success.

Within a day, we had our first result – we tripped the Li ion detectors and brought down the beam. Though not popular with the B-factory group to whom we were supposed to be parasitic, we were ecstatic. Not one to jump up and down, I think Bob's reaction was to say "I think that's good." Ralph ran a quick plot of beam spot size after the plasma vs. laser energy (proportional to plasma density) for the 100 or so shots before the shutdown, and a smooth curve appeared on the screen showing the strong focusing and then over-focusing of the beam by the plasma that caused the spray that tripped the safety system. The curve fit a focusing strength of 6000 T/m. So we went to the daily 8AM meeting with good news to report – we knew the plasma forces were strong and now we just had to control them.

Over the course of several years, we learned how to tame the forces, but not without a significant learning curve and a few false positives. Here was another area that Bob's rigorous adherence to standards of proof served us well. Bob insisted on scrutinizing all the data himself and on everyone else doing the same. Sometimes this could be maddening as months would drag on before a paper could be submitted. But when it came out, it was always right.

We eventually were the first to accelerate electrons beyond 1 GeV in laboratory plasma, and went far beyond [1]. Thanks to the talents of the students and professionals in the group and the addition of Paul Emma and Patrick Krejcik who were able to bring the bunch length down by a factor 20, we were able to exploit the (nearly) inverse square scaling of wakefield amplitude with bunch length and accelerate 10^8 electrons by 43 GeV over 85 cm.

When I arrived in January 1999 at the SLAC campus, I brought a bottle of Dom Perignon to be opened when we broke a GeV. I don't think we actually drank it until nearly a year after the milestone was achieved. It wasn't for lack of enthusiasm, it was just that Bob was not one to accept a result without completely vetting it. Now that Bob is gone, I suppose that is the part of his legacy I am missing most. I don't want to accept the finality of his passing, certainly not so quickly.

References

1. Ian Blumenfeld, Christopher E. Clayton, Franz-Josef Decker, Mark J. Hogan, Chengkun Huang, Rasmus Ischebeck, Richard Iverson, Chandrashekar Joshi, Thomas Katsouleas, Neil Kirby, Wei Lu, Kenneth A. Marsh, Warren B. Mori, Patric Muggli, Erdem Oz, Robert H. Siemann, Dieter Walz, and Miaomiao Zhou, "Energy doubling of 42 GeV electrons in a metre-scale plasma wakefield accelerator," *Nature* 445, 741 (2007).

2.11 Bob Siemann's Contributions to Advanced Accelerators – a Personal Perspective

Wim Leemans, LBNL
Mail to: WPLeemans@lbl.gov

I first met Bob Siemann at the Advanced Accelerator Concepts Workshop in 1996, held at Lake Tahoe, where Bob gave the plenary talk on the status and future directions for advanced accelerator research, with a focus on concepts for future colliders. Conventional accelerator technology has been evolving and maturing for many decades and now forms the backbone of, for example, the impressive Large Hadron Collider in CERN and the soon-to-be-completed LCLS at SLAC. Advanced accelerator concepts have been pursued with great vigor since the seminal paper in 1979 by T. Tajima and the late John Dawson but, in comparison, are still in their infancy. They hold the promise of building accelerators that are much more compact than those using today's conventional technology.

For someone like Bob – a renowned accelerator scientist with decades of experience in designing, building, and operating machines based on conventional accelerator technology – to discuss this topic provided a much needed legitimacy to our young field. When new concepts are proposed, they are met by some with exuberance and by others with skepticism. The burden of proving the viability of a concept can take many years. It takes tenacity in the face of adversity, it takes optimism and a strong sense of direction, and most of all it takes people who provide intellectual leadership and vision. As things turned out, Bob would make his mark on our community not only by his words but, even more important, by leading groundbreaking experimental work in advanced accelerator science.

In 1997 the first plasma wakefield accelerator experiment, E-157, was proposed at SLAC as a collaboration between teams from SLAC (headed by Bob), UCLA (headed by Chan Joshi), USC (headed Tom Katsouleas), and my own group at LBNL. The goal was to demonstrate 1 GeV energy gain of a fraction of the electrons in a 30 GeV drive beam in a 1 m-long plasma cell. It was Bob's involvement and his belief in the validity of advanced accelerator concept research that made access to the unique SLAC linac possible. Bob provided the drive and intellectual leadership needed for the first demonstration of a plasma-based accelerator at the premier linear accelerator laboratory in the world.

In the E-157 experiments, the high energy (30 GeV) beam from the SLAC linac was focused onto a Li-vapor oven that was pre-ionized with an Excimer laser. As the beam entered the plasma it moved plasma electrons out of its path, leaving the ions unshielded, much as a motorboat moving across a lake generates a wake. Whereas the majority of the electrons lose energy in the process, a small fraction of the bunch could gain energy by surfing the wake generated by the bulk of the bunch. This required fine-tuning the plasma density and electron beam parameters; thus monitoring and controlling the overlap between the electron beam and the plasma column became a major challenge.

For E-157, my group proposed the use of optical transition radiation (OTR) to measure the spatial properties of the electron beam. This had not been tried at the beam energy of 30 GeV and was met with skepticism – including Bob's. Despite his

reservations, he had the curiosity and openness of a true scientist. He made it possible for us to try the diagnostic by negotiating time on the SLAC linac; the OTR test worked, and Bob quickly recognized the importance of implementing these diagnostics to help with the monitoring of the electron beam right at the entrance and exit of the plasma column.

The E-157 experiments proved to be a real learning ground, where laser-plasma physicists and conventional accelerator physicists exchanged knowledge and information and trained one another in the dos and don'ts of their trade. I remember sitting in the makeshift control room in a trailer on the SLAC site near the FFTB, watching the flashing image of the beam on the screens while Bob worked feverishly on programs that allowed instantaneous feedback of what was happening in the experiments. He taught us all the benefits of applying “real high-energy physics” methods (e.g., on-the-fly data sorting and statistics on high-repetition-rate data streams) which, for the single shot laser-plasma community at the time, were new.

In 1999 I decided to concentrate on a different method of acceleration in plasmas and thereafter focused all of our resources back at LBNL. The implementation of the E-157 experiments were replaced by much-improved setups; subsequent experiments by the SLAC/UCLA/USC team, plus the know-how accumulated over the years, resulted in a wealth of plasma wakefield physics, with many top-tier publications and excellent student dissertations. Bob's rigorous approach and his insistence on clean, understandable data were among the main contributors to the success of the plasma-based acceleration experiments at SLAC. In 2007, the SLAC/UCLA/USC collaboration succeeded in demonstrating 40 GV/m accelerating gradients sustained over a meter distance in a plasma and the doubling in energy of a fraction of the electrons in the 42 GeV driver to an energy of 85 GeV. [1]

When the involvement of my group ended in 1999, we concentrated on the other plasma-based track at home in Berkeley: laser wakefield acceleration, conceptually very similar to plasma wakefield acceleration. In laser wakefield acceleration an intense laser pulse propagates through a plasma and excites a density wave by expulsion of electrons away from the laser pulse, leaving the heavy ions unshielded. As a result, a large electric field can be generated that copropagates with the driving laser pulse. Fields three or four orders of magnitude greater than those achieved in breakdown-prone conventional accelerator structures can, in principle, allow acceleration to a GeV or more in just a few centimeters. In 2004 the so-called “dawn of the laser accelerator” occurred with the first demonstration of quasimonoenergetic electron beams by three independent groups from RAL/Imperial College, LOA, and my group in Berkeley.

As far back as the 1996 AAC workshop, Bob had correctly identified the key issues for our field. In the proceedings of the 1996 workshop he writes that, following the observation of gradients of ~ 100 GeV/m, the “field of laser driven plasma accelerators is moving on to achieving this acceleration over long distances, staging of multiple accelerator modules, and beam stability and quality. When these have been successfully addressed the plasma accelerator will attract the interest of the mainstream accelerator community.”

Ten years later, acceleration distances have increased from the mm- to the cm-scale by using plasma waveguide structures that act like an optical fiber to overcome the natural diffraction of the focused laser beam. In 2006, in collaboration with Oxford University, we showed that by extending the guiding distance and lowering the

operating density, GeV beams can indeed be obtained from a cm-scale plasma waveguide structure. [2]

Concepts are now being developed by several groups around the world to cascade or stage these accelerator modules to further increase the beam energy, beyond what a single module can produce after the laser pulse energy has been depleted. Injection techniques are being tested that either rely on additional laser pulses for controlling the injection of electrons or on specially tailored density profiles, where waves can more easily capture background electrons. Efficiently coupling energy to the particle beam while maintaining proper energy spread and emittance is also being investigated. If at the same time laser technology improves at the rate it has been improving since its inception in the 1960s and can provide high average power (multi-kW class), high peak power (Petawatt class) systems, operating at high wall-plug efficiency, we believe the dream of a laser-powered collider could become reality. Over the years, Bob kept a keen eye on the progress in the laser community, and we often had conversations on exciting developments in both particle and laser beam driven plasma schemes and on how to grow the field.

Bob was one of the statesmen of our community. He spoke with clarity and authority about the physics, the challenges, and the progress of the field. Through the years he mentored and trained many students and accelerator-science practitioners who will now continue his legacy through their own outstanding work and contributions. Our community owes Bob eternal gratitude for having been a powerful spokesperson and, even more important, for having been someone who rolled up his sleeves and became a most powerful practitioner, working on real experiments, on real machines, to validate those concepts that all of us believe could revolutionize our field.

References

1. I. Blumenfeld et al., “Energy doubling of 42 GeV electrons in a metre-scale plasma wakefield accelerator”, *NATURE* **445**, p. 741-744 (2007).
2. W.P. Leemans et al., “GeV electron beams from a centimetre-scale accelerator”, *NATURE PHYSICS* **2**, p. 696-699 (2006).

2.12 Bob Siemann and the Spallation Neutron Source: A Remembrance

Stuart Henderson, SNS, ORNL
Mail to: shenderson@ornl.gov

Few people know the impact that Bob Siemann had on the Design, Construction and Commissioning of the Spallation Neutron Source (SNS) at Oak Ridge National Laboratory. Bob chaired the SNS Accelerator Systems Advisory Committee (ASAC) through fourteen meetings, beginning in October 1998 and finishing in June 2006, and therefore saw the SNS construction project from beginning to end. In that role, Bob was asked to lead the committee in providing advice on the technical aspects of the SNS accelerator design, construction, beam commissioning, and transition to operations. Bob also provided advice on management, dealing with the complex multi-lab collaboration partnership as well as the buildup of a strong team at Oak Ridge.

I first became aware of Bob's work and reputation as a post-doc at Cornell University, working on the CESR e⁺e⁻ collider. Although Bob had left Cornell by the time I arrived, his reputation and imprint was evident in many areas, and his name would come up frequently in discussions. I first met Bob when I gave a talk at the American Physical Society (APS) April meeting in the early 1990s in a session that Bob had organized as part of his Division of Physics of Beams responsibilities. At the time I was unaware of, but later came to appreciate, his passion for the success of the APS Division of Physics of Beams, and his passion in working to place accelerator science on an equal platform with other fields of science.

Several years later, in 2001, I joined the Spallation Neutron Source project at Oak Ridge National Laboratory. I recall that during my first ASAC meeting, we were seated at the same dinner table at the obligatory review dinner on the first evening, and we began talking. When he learned that I had recently moved from Cornell, he filled with excitement as he began asking me of people, places and things that he remembered from his days there on the Faculty. We spoke at some length about CESR, about his friends there, about the campus and about Ithaca.

Over the next 5 years I had the pleasure of seeing Bob twice a year, and giving about 10 presentations to him and the committee as we began building and then commissioning the SNS. I quickly learned in those reviews that if you convinced Bob that your ideas and plans were sound, then they probably were in fact sound. On the other hand, he never held back in his urging the project time-and-again to uphold a high standard: to do a better job, do a more thorough analysis, and make smart decisions for the long term.

In retrospect, Bob was the ideal chair of such an advisory committee. Over the course of those 8 years, the SNS design changed from a normal-conducting linear accelerator to the world's first pulsed superconducting high-energy proton linear accelerator, flirted with a rapid-cycling synchrotron as an alternative to an accumulator ring, changed the design beam power from 1 MW to 2 MW to 1.4 MW, changed from an "alpha"-shaped machine layout to an "omega"-shaped layout, and incorporated many other smaller design changes in the process.

Bob was truly a generalist in the best sense of the word. There was not a topic under discussion, whether in linac beam dynamics, ring collective effects, pulsed power systems or laser-based diagnostics for which Bob did not have wise guidance to offer. Re-reading the early advisory committee reports reveals the focus even at the first review on the challenges that would lie ahead, and on those issues that would be of greatest concern and struggle early in operations:

...The SNS Project is in an unusual state outside much of our experience in that it is a construction project with significant money appropriated and yet there remain issues at the conceptual design level that need to be resolved. Three particular areas that should be given attention by the SNS collaboration and management are reliability, beam loss, and the implications of the transition from a multi-laboratory construction project to an operating accelerator at Oak Ridge National Laboratory...

...There are no reliable beam physics models which can credibly predict losses at this small level, and it is unlikely that such models will be developed before the design is finalized. Therefore, the linac and accumulator ring designs must be conservative by today's understanding, and they must be flexible...

One year later, the committee was asked to evaluate a proposal to replace the high-energy normal-conducting linac with superconducting technology. This was a major design change, carrying with it substantial risk, arising in mid-construction. The committee strongly endorsed this plan with advice that proves prescient nearly ten years later:

...Superconducting RF has already been accepted for intense, light-ion cw applications. Its technical application to pulsed light-ion applications, such as the SNS linac, awaited confidence that the RF fields and phases could be controlled precisely with Lorentz force detuning and microphonics present. This control has been developed and demonstrated at the TESLA Test Facility, and the TESLA design seems adequate to fulfill the SNS needs...The SNS design study clearly indicates that superconducting RF is economically and schedule competitive. With superconducting technology also applicable to pulsed high-intensity accelerators, it is clear that superconductivity is the technology of choice for many future linacs.

I last saw Bob at SLAC in mid-July of 2008, just several weeks before his passing. He knew that I was in town for a review, and made a point of stopping by the portion of the review that pertained to Advanced Accelerator R&D, the focus of the most recent phase of his career. During the lunch break we found a spot alone and talked. He asked me how the SNS was doing and was excited to hear of the latest news, and was very pleased that we were ramping up the beam power beyond 500 kW. We chatted about the SNS and then he turned the conversation to one of his students – someone about whom we had corresponded several months earlier. I could tell that his health was failing, but the passion that he showed for his students shone through. His deepest concern on that day was that his students find successful positions when they left SLAC.

The field has lost an exceptional physicist, scholar, spokesman, teacher, mentor and friend to many. He will not be replaced. We can only learn from his example, and continue the work that he began.

2.13 Bob Siemann and PRST-AB

Martin Blume

Editor-in-Chief Emeritus, The American Physical Society

Mail to: blume@aps.org

It is painful for me to write in memory of Bob Siemann. We worked so closely together for such a long time that I find myself, as before, reaching for the telephone to call him, this time to check on the accuracy of my memories. But now my own recollections will have to do.

My first meeting with Bob was in difficult circumstances. In 1983 the Brookhaven National Synchrotron Light Source X-ray ring was behind schedule and not yet operational. The Department of Energy had informed the Laboratory that they were appointing a “blue-ribbon” review panel to determine what was wrong, and even whether to terminate the project. I was asked (told) to take over the management and to prepare for the review. Bob was a member of the review committee, and when some of the experimenters wanted to have the ring brought up to speed while they went to carry out experiments at Stanford, he supported me very strongly, saying that the rings and

the experiments had to be simultaneously commissioned. I also, during this difficult process, learned much about the importance of accelerator physics and engineering. Accelerator scientists from around the world came to help in our examination of the problems of the ring and their solutions. All of them had, at some point in their careers, to sit up at night with sick accelerators in order to nurse them back to health, so they gave generously of their time. In the end, thanks to their help, we were successful in turning around the negatives, making the project whole. I learned too of the scientific and engineering challenges, with substantial intellectual aspects, that were faced in these important facilities, and developed a great respect for those who worked on them.

Bob was in many ways in a unique position. He had obtained a professorship at Cornell, where he had received his Ph. D. as a particle experimentalist. Having done so he became interested in the physics and engineering of accelerators, and was able to pursue this career, with great success, as a tenured professor. Many accelerator scientists and engineers were, on the other hand, not so lucky. They tended to be regarded as technicians, and were relegated to positions as staff scientists rather than as professors. Their publications often appeared in conference proceedings rather than in established journals, and their applied contributions were put forward as technical notes. This made it difficult for the usual academic processes to recognize what they had accomplished. Bob was intent on changing this, and he worked to do so as a senior member of the American Physical Society Division of Physics of Beams (DPB).

In 1996 I was named Editor-in-Chief of the American Physical Society, at an exciting and challenging time. Electronic publication of journals was just coming to maturity and the internet was becoming robust enough to sustain the increased traffic necessitated by rapid dissemination of the journals, especially in the face of increased public use. The first efforts had to go into putting our existing journals on line. I was approached by Bob on behalf of a committee of the DPB concerning the establishment of a new journal that would publish both technical and scientific articles on beam physics and accelerator technology. The purpose, as Bob explained it, was to give accelerator physicists and engineers a place to publish so that they could have a resumé that would give them an opportunity to be considered for higher rank positions at universities and national laboratories. It was in fact pretty clear to me that Bob was doing this to get my rejection and to clear the way for another publisher to establish the journal. Bob realized that since the DPB was a unit of the APS it would be inappropriate to establish the journal without clearing this with the APS – he was very careful about such matters as he was already a Divisional Associate Editor of *Physical Review Letters*. But when he explained what was wanted I reacted enthusiastically, to his surprise. After a few days of discussion with him and with Stanley Brown, Editorial Director of our journals, I suggested a journal with the name *Physical Review Special Topics* (possibly the first in a series). I agreed that the journal should contain technical and applied content, along with the basics of beam physics. I also wanted a title for the journal that had the name *Physical Review* in it, in order to bestow some of the prestige of that journal on the accelerator scientists and engineers who would publish there.

Bob had to return to the DPB leadership with this proposal. Bob suggested the name *Physical Review Special Topics – Accelerators and Beams* with the abbreviation PRST-AB and the unfortunate verbal shortening to STAB. The journal was to be electronic only (no print version) and it was a pioneering effort in this direction. We also proposed that it be supported by the sponsorship of major university and national laboratory accelerator groups, so that the journal would be open access – no subscription required.

The most sensitive point to me was that Bob had to be the editor, and he had at the same time to obtain the support of the DPB. Bob had a wonderful sense of the political, and he was able to do this well.

Bob described much of this in his memoir for the tenth anniversary of ST-AB [<http://prst-ab.aps.org/edannounce/PhysRevSTAB.11.050003>]. He gave an excellent history of the journal, and he mentioned many of the people in the editorial office who made significant contributions to the journal's beginnings. He had sent a draft of this to me in May of this year, and my principal complaint was that he had been much too kind to me, and insufficiently so to himself. I want to redress this here.

Without Bob there would have been no such journal. He made it happen, and he drove its success. In fact, whenever we started a new initiative in our journals we always asked the question "who will be the Bob Siemann for this?"

Bob published an early article in the journal, of which he was quite proud. But the next article that he submitted was rejected! (Of course there was another editor who handled both of these.) And typically of Bob he was even prouder of the rejection than of the acceptance – it showed the stringent standards that he wanted for the journal. He promoted the journal internationally and raised funds for sponsorship throughout the world. When his illness caused him to step down we had to search for a successor, and we have been very fortunate to have Frank Zimmermann take over, a choice of which Bob enthusiastically approved.

My highest praise for Bob is that I answered his phone calls and emails immediately, and I treasure what he taught me, both technically and in how to lead with high standards but with grace. I miss him.

2.14 Hands across the Atlantic

Mike Poole, Liverpool University
Mail to: mike.poole@stfc.ac.uk

In 1997, whilst Bob was busy negotiating with APS on a new journal that was to become PRST-AB, a similar venture was under consideration in the UK, possibly to be sponsored by the Institute of Physics (IOP). The reasons for my involvement in this British project were identical to Bob's: a belief that the discipline in which we were so deeply embedded needed to foster its professional activities through an encouragement of higher level publishing practices than those to which we had mostly become accustomed. This was judged to be essential for the long term health of our areas of science and technology, not least in attracting and sustaining rewarding careers for outstanding young scientists and engineers by demonstrating academic excellence both in its science and technology outputs.

After the launch of the new APS journal early in 1998 I was approached initially on behalf of the APS Division of Particle Beams (APS-DPB) by Andy Sessler during that year's EPAC Conference in Stockholm, to express his concern that a second journal might be badly timed for the success of both. After careful consideration the European accelerator community, through its executive Accelerator Group attached to the European Physical Society (the IGA), decided to investigate an alternative joint approach with the DPB and invited Bob to address their Committee later in 1999. I well remember that meeting for the enthusiasm and clarity with which Bob presented his vision to us. It was my first (but not last) experience of the very considerable diplomatic

skills that supplemented his deep intellectual abilities. It was also a seminal event for publishing policy for our profession, as there is no doubt in my mind that many were still cautious about accepting an American led initiative at that time.

It is I suppose surprising that it took so much longer after that for serious collaboration to emerge, but the European organisation is of course more diffuse than that of our USA colleagues. In 2001 Bob invited me to attend an Editorial Board meeting at PAC that year in Chicago and I also spent time with him discussing practicalities of European involvement. I also saw just how much PRST-AB was his creation and dream. The energy and enthusiasm was undimmed even after three years of such hard work. I have no doubt at all that without Bob in that critical start-up period the enterprise could have faltered.

Eventually, in February 2002, the IGA agreed formally to a joint role with the DPB and soon afterwards I was appointed as Associate Editor, as nominated by Europe. This immediately put me into a close working relationship with Bob and he became very much my mentor in those early months. Only those who have undertaken editorial duties on a major journal can appreciate the complexity of the challenges, especially with the time pressures of an all-electronic system with its commitment to rapid publication. Bob nursed me through referee databases and selection – and sometimes nudged me into more efficient responses. This period coincided with a successful phase of journal expansion, with a dozen papers a month to be processed between us. We had agreed on a split of responsibilities by topic area but he still retained a larger fraction of the total and quite how he coped with the workload only he knows. He was also encyclopaedic in his knowledge of our broad community and this helped him to identify suitable referees over a huge range of activities, varying from theoretical physics through to advanced technologies. He was aware, however, that his background was quite removed from those members of our community specialising in light source development and he was most anxious (and relieved) that my own contacts in that grouping were extensive.

Over and above the routine publication processes we did have a few ‘difficult’ cases to solve. This is where the Editors had to intervene to resolve conflicting views about a paper. I became acquainted with Bob’s diplomatic skills again on these few occasions. Each time he proposed a calm solution that preserved the dignity of authors and referees. I know he took paper rejection very seriously and although the overall refusal rates were not unreasonable for a high level journal (~30%) he was well aware that many of these authors had invested much time and energy on their submissions. Indeed I never heard him make an unkind remark about any author (although he came close with some of our most tardy referees!).

Bob was always open to ideas to evolve the journal and he took advice from his Editorial Board very seriously. The Special Editions attached to conferences occupied us every year at that time but never quite succeeded as much as both of us desired. Bob was especially anxious to widen the refereed publication base by encouraging members of our community who had not traditionally looked beyond conference proceedings; unfortunately the important refereeing standards of a first rate journal to which he was absolutely committed did prove an impossible hurdle to jump for some, much to his regret.

I was privileged to share an important enterprise with Bob in that period from 2002 to 2005 and it was a rewarding, if challenging, experience. Bob was a big man – and not only in physical stature. In fact he was a giant of our community and he will be

sorely missed. Perhaps it is a consolation that he has left behind, in addition to all of his important personal research, such a solid edifice as PRST-AB for which he will always also be remembered.

3 International Linear Collider (ILC)

3.1 Third International Accelerator School for Linear Colliders

Barry Barish and Weiren Chou

Mail to: barish@ligo.caltech.edu, chou@fnal.gov

The Third International Accelerator School for Linear Colliders took place from October 19 to 29, 2008 at the Oakbrook Hills Marriott Hotel near Chicago, Illinois, U.S.A. (<http://www.linearcollider.org/school/2008/>) and was a continuation of the previous two international LC accelerator schools held in 2006 in Japan and 2007 in Italy. This year's school was jointly organised by the ILC GDE, the International Linear Collider Steering Committee (ILCSC) and the ICFA Beam Dynamics Panel. Fermilab hosted the school.

The school was aimed at PhD students, postdocs and young researchers, especially young experimentalists. The response to the school was overwhelming. We received 245 applications from 37 countries and most of the candidates presented strong credentials. However, the school could only accommodate a limited number of students. Through a difficult and rigorous selection process, the Curriculum Committee accepted 57 students from 14 countries. The committee members carefully read the CV and recommendation letter of each applicant, and discussed among themselves before making the decision to admit or reject an applicant. For various reasons especially due to visa problems (see discussion below) twelve admitted students unfortunately could not come. The forty-five students who attended the school were a talented and highly motivated group. They successfully met the challenge of an intensive 8-day lecture program and did well in the final examination.

The curriculum consisted of 12 lectures, homework assignments and a final exam. The lectures covered both basic and advanced topics in linear accelerators, damping rings, linear colliders and the muon collider and were given by a group of accomplished accelerator physicists, many of whom are GDE members. These lecturers not only gave lectures during the day, but also gave tutorials and helped students with their homework in the evenings. They also created the examination problems and graded them. The final exam on the last day lasted four-and-half hours. All 45 students took the final exam. The lecture slides, homework and exam problems can be found on the school web site.

In addition to regular lectures, the students paid a site visit to Fermilab. This gave them an opportunity to learn about real accelerators. Roger Dixon, head of Fermilab's Accelerator Division, presented a special lecture on how the Fermilab accelerator complex works. They also received hands-on training in the Main Control Room and performed several beam measurements and manipulations in the real machines. They visited the Wilson Hall 15th floor public display areas, the Linac Gallery, the Industrial

Buildings, the CDF and D0 Experimental Halls, and the superconducting RF R&D facilities.

The exam problems were challenging and the students did well as shown in the figure of exam scores. The top 9 students were honoured at the banquet and each was awarded a certificate and a book (*Handbook of Accelerator Physics and Engineering*, edited by A. Chao and M. Tigner, and published by World Scientific).

Throughout the school period, the students were encouraged to make new friends since this was a once-in-a-lifetime opportunity for many of them to meet with other young talented people from different origins who shared the same interest (accelerators) and career goals (linear colliders). Some of the friendships nurtured at the school will last a lifetime.

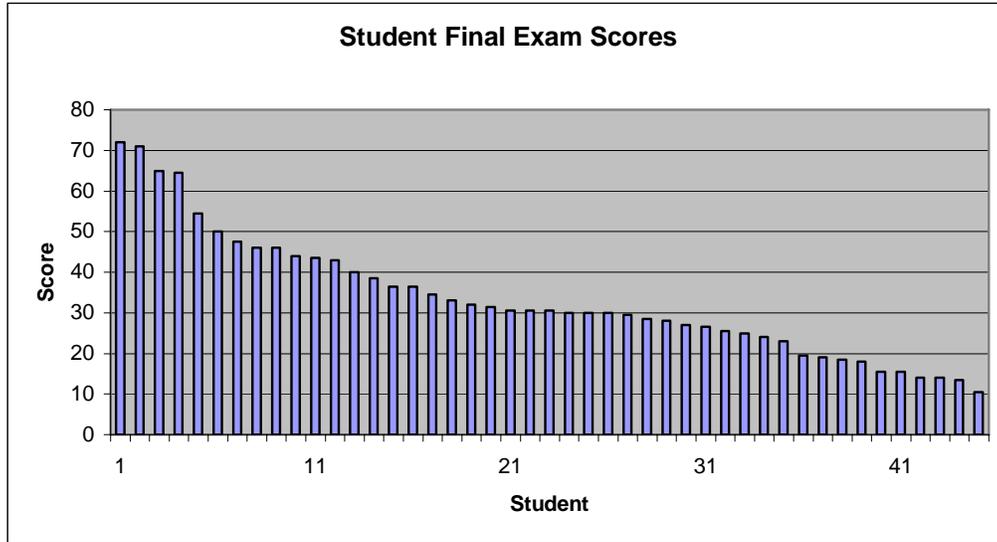
Fermilab's Conference Office was responsible for the organization of this school. Cynthia Sazama, Suzanne Weber and Jean Guyer spent an enormous amount of time and effort over many months to make the school a success. They arranged everything for the school: budgeting, travel, hotel room assignment, lecture room, study room, meals, reception, banquet, coffee breaks, A/V, printing and distribution of lecture slides and homework assignments, site visit, excursion, photos, and students' special needs. As a matter of fact, they moved a big portion of their office on the 8th floor of Fermilab's Wilson Hall to the Marriott Hotel during the school. They did an outstanding job, which we greatly appreciated. Susan Winchester from the USPAS Office negotiated a contract with Marriott for the school and provided valuable assistance. Amanda Petersen, head of the Fermilab International Affairs Office, played a key role in student visa applications.

The school received generous sponsorship from a number of funding agencies and institutions all over the world: U.S. DOE Office of Science, NSF, Fermilab, SLAC, ILC GDE, CERN, DESY, IN2P3, INFN, Oxford University, University of Manchester and University of Bonn. KEK supported all 18 students from Asia. Fermilab and FRA covered a majority of the local expenses.

A special difficulty encountered by the school was the visa problem. Among the 57 accepted students, 24 required a visa to enter the U.S. We were aware of the long waiting time for visas. So we began the process in mid-June, four months prior to the school. The Fermilab International Affairs Office took the lead in this effort. It sent to each of the 24 students the required documents as well as step-by-step instructions. The information was posted on the school web site. We also solicited assistance from the Visa Office of the National Academies of Sciences and American Physical Society. Amanda Petersen sent a list of applicants to the U.S. State Department and kept in touch with them almost on a daily basis. However, despite all these efforts, 9 students did not receive visas by the beginning of the school. (Three other students couldn't attend because of personal reasons.) Their purchased round-trip air tickets (which were required by the U.S. consulates) had to be cancelled. This was very unfortunate. We heard numerous similar stories about people from other countries who couldn't attend international scientific conferences in this country because they failed to obtain a visa in time. A recent article published in *Science* (November 21, 2008, Vol 322) said U.S. visa delays are getting worse, not better. It has been a serious problem for many years and must be addressed at the highest level of government.

We carried out a student survey on the last day of school. The results were given to the lecturers and committee members for improvements for future schools. Based on the interest, demand and success of the first three schools, it was decided to continue the

school in 2009. It will take place in the fall of 2009 near Beijing, China and will be hosted by the Institute of High Energy Physics (IHEP). The exact dates and place will be chosen soon and announced in the next issue of this newsletter.



4 Theme Section: Control Systems

4.1 Accelerator Control Systems – A Personal Recollection

David Gurd, ORNL (retired)
Mail to: davegurd@shaw.ca

4.1.1 Introduction

It is September 10, 2008. I am sitting alone in my home in Vancouver looking at pictures of the LHC Control Room on the (CERN-invented) World-Wide Web. A few hours ago, beam circulated in the LHC for the first time. People are standing, smiling, applauding and pointing at the tens of colorful screens that tell the story of the day's triumph. I wish I were there to share the excitement. I can almost taste the champagne.

I have tasted that champagne twice in my life: on a Sunday afternoon, December 15, 1974 when first beam was extracted from the TRIUMF cyclotron in Vancouver, BC and on a Friday afternoon, April 28, 2006 when the first protons struck the liquid mercury spallation target at SNS in Oak Ridge, Tennessee and neutrons were detected in the Experimental Hall. (Rather inferior champagne in both instances, as I recall, but especially delicious because of the occasions.) These two events were bookends to a 37 year career working in the field of accelerator controls. This newsletter describes in some technical detail a number of more recent accelerator control systems, including that of the LHC. For my part, I wish to introduce the subject with a brief, anecdotal and very personal overview of the evolution of accelerator control systems as seen by me over that 37 year period.

4.1.2 The TRIUMF Years

Those defining moments in the history of particle accelerators are inevitably recorded in control room photos; everyone checking some display to verify that their own system is performing as expected or leaning forward to catch a glimpse of some number or wiggle on a screen that will be the evidence of a critical milestone achieved. Only the controls team is conscious of what is going on behind those display screens – of the tens of kilometers of cable, the hundreds of interconnected computers, the hundreds of thousands of lines of computer code and of the fast-switching networks distributing commands from operators and applications, searching for anomalies and collecting vast amounts of widely-distributed data for display, archiving and analysis.

It was not always so complex. When I joined the controls group at TRIUMF in 1970, the director did not want computers used for control. Indeed when the first terminal and keyboard was approved for programming purposes at TRIUMF (for \$5000), he personally wrote on the order: “Not to be brought into the Control Room!” (I kept that document as a souvenir until it was destroyed in the Los Alamos fire.) Instead, I was sent to Berkeley and UCLA to learn how a cyclotron should be operated – with knobs, relays, “nixie tubes” and running shoes. Real men didn't need computers.

Eventually the director was persuaded by laboratory demonstrations that computers could be used to make some things easier, and when in November/December 1974 he personally operated the machine for 14 – 16 hours a day, seven days a week for almost three months to deliver its first beam, the control system was computer-driven and at least one CRT screen had been smuggled surreptitiously into the control room. At the 1975 Cyclotron Conference in Zurich the director – famous in the cyclotron community for his public scepticism about the use of computers – acknowledged that the computer-based control system had worked and helped to make possible the success of TRIUMF. It was a moment of great satisfaction for me.

That early system used three 32K word (16 bit) Data General Supernova minicomputers, selected only months before the announcement of DEC's eventually dominant PDP 11. These computers were programmed in assembler language with punched cards and paper tape. The real-time operating system (NATS¹ or Nova Asynchronous Tasking Supervisor) was 256 words long (compare VxWorks or RTLinux today) and was delivered on a short strip of punched mylar tape. (Paper was considered too fragile for the many times it would have to be loaded.) The computers communicated with each other over a point-to-point high speed parallel link and with operators using a 10 character/second clickety-clackety KSR 33 teletype machine. The computers were interfaced to equipment using the brand new CAMAC standard with parallel branches (serial branches were still to be invented). It was a very early application of CAMAC to accelerator control – the meson factories being built contemporaneously in Los Alamos, NM and Villigen, CH had invented their own interface buses. The use of computers permitted the distribution of remote consoles (also interfaced using CAMAC) around the laboratory. At TRIUMF these were sometimes accurately referred to as “UHPs” or “User Hostile Panels.” Oh yes – there was also a device naming convention: all devices were assigned a two character system code followed by a three digit octal (!) number. Incredibly, that system is still in use!

I stayed at TRIUMF for a total of twenty years, during which time fancy new “high-level” computer languages such as Basic, Fortran and eventually “C” were introduced to the control system and the computers themselves advanced to the 32 bit Eclipse from Data General and eventually included the ubiquitous Vax. Some intelligence was distributed down to the CAMAC I/O crates using auxiliary crate controllers. You could no longer tell what the computers were doing by looking at flashing front panel lights; you could no longer debug programs by single-stepping through them with front panel switches; and the heavier languages and operating systems significantly slowed the refresh rates of the ever-increasing number of computer display screens.

4.1.3 A Community

In January of 1985, a small workshop was hosted by Brookhaven National Laboratory to solicit advice to deal with problems in the development of the control system for NSLS, then under construction. Some participants from Los Alamos immediately saw the potential benefits of getting a group of accelerator control system designers together to share experiences and perhaps even write something down and a

¹ NATS was the first product of what was then a three person Vancouver start-up company – MacDonald, Dettwiler and Associates – that subsequently grew into a billion dollar plus information systems giant employing over 3300 people in Canada, the US, the UK, Germany and India.

workshop on Accelerator Control Systems was organized there for later that year. Generally considered a success, a larger and more formal workshop/conference on Accelerator Controls was held two years later in Villars near CERN with more participants and many more presentations. Two years after that, in Vancouver in 1989, a fully-fledged conference series, now dubbed ICALEPCS for “International Conference on Accelerator and Large Experimental Physics Control Systems,” had been born. There was now a true community of accelerator controls experts sharing their ideas and experiences and a permanent record of those exchanges. These biennial conferences now circulate around the globe and average 300 – 400 participants. ICALEPCS 2009 will be held in Kobe, Japan.

Much of the evolution of control system technology is recorded in the proceedings of those conferences. In 1987 in Villars there was a very animated evening discussion about whether or not UNIX could be used as a control system operating system with a very vocal contingent declaring it to be inherently unsuitable because of its lack of appropriate interrupt handling capabilities. 1991 in Tsukuba saw the first description of what is now known as the “Standard (three-layer) Model” for accelerator controls. Today’s discussions center around new hardware and software technologies, new communication protocols, the use of commercial SCADA and other systems, and the looming issues of very large international collaborations.

4.1.4 Collaboration - EPICS

In the late eighties and early nineties there was much discussion of the need for collaboration. We (the Accelerator Controls community) were all solving the same problems over and over, which was both inefficient and costly. The concept of an Accelerator Controls Toolkit that could be drawn upon by many was described and developed at a Workshop held in Los Alamos in October of 1988 and co-hosted by LANL and CERN. Notwithstanding many good ideas, not much progress was made.

In 1989 at the Vancouver ICALEPCS a group from Los Alamos submitted a series of papers describing their new system, called GTACS – Ground Test Accelerator Control System. The LANL censors changed the name to LACS at the last minute (very annoying for the program editors) because the GTA was part of the Strategic Defense Initiative – later known as Star Wars. The Los Alamos Team was looking for partners. Until then the prevailing concept of collaboration seemed to be: “You should just use our system – it’s wonderful and we’d be happy to let you have it.” The breakthrough came when the Los Alamos team added the phrase: “...and while you’re at it, please make improvements and add new features. *We’ll incorporate them and use them ourselves.*” With that understanding and spirit the Advanced Photon Source controls team at ANL signed up and the EPICS Toolkit and collaboration was born.

Interest spread. If a new partner complained that EPICS lacked one feature or another, they were invited to provide it themselves and it would be added to the distribution. Only two rules: Use the EPICS distributed real-time database and use the EPICS communication protocol. After that – you’re on your own. The model was chaotic. No one controlled it. If someone didn’t like the display manager, they could (and did) develop another. (At one time there were 4 or 5 to choose from.) If theirs really was better, others would adopt it. In true Darwinian fashion the fittest survived. Although this undisciplined approach was itself inefficient it was the first approach that engendered successful collaboration. New projects could benefit from the fast-growing

base while programmers were still able to exercise their creativity and have their work used and appreciated beyond their home laboratories and projects. And so as EPICS grew in scope and capability, the collaboration grew in size. By the end of the nineties EPICS had been open-sourced by the two originating laboratories and soon after that the constraint on the real-time operating system was removed. EPICS could run on VxWorks (its original home), Linux, Windows and others. Already by the mid-nineties over one hundred projects were using EPICS with perhaps 15 - 20 being significant contributors.

Notwithstanding being based upon software technology almost two decades old, EPICS continues to be selected for new projects large and small. More recently other collaborations, most notable the Tango Collaboration of European Light Sources, have evolved along similar lines, using more modern software technology. Both EPICS and Tango are described in some detail in articles in this newsletter. There is little motivation for changing large existing, functional, maintainable control systems but there is a strong incentive for new projects to consider participating in an existing collaboration and to benefit from the instantly available high performance infrastructure, a pool of “experts” from around the world and a clear interface on which project-specific applications can be developed.

4.1.5 The SSC

In January of 1991 I moved from TRIUMF to head the Controls department at the SSC. You all know how that came out – except for me it came out well. I had become altogether too comfortable working with a mid-sized project team on a mid-sized accelerator; at the SSC I learned to work on a large-scale project. I learned to consider the technical issues related to scale. What works well for half a dozen minicomputers may not scale to thousands of front-end processors. I learned to use and believe in the tools of project management and configuration control. I also learned a lesson in political correctness: When we christened the SONET-based accelerator controls network “BUBBAnet,” (for “Bidirectional Underground Big Big Accelerator”) we were severely chastised by management and told to use the name no longer. (One paper was published with that excellent and very Texan sobriquet.)

When I arrived at the SSC, the budget for controls was \$50M. This was less than we were ultimately to spend on controls for the SNS, which was (approximately) one tenth the cost, size and complexity of the SSC. A better estimate, and far closer to typical “rules of thumb” for controls, would have been \$500M. That correction had not yet even started its change control process, and I guess I was lucky to have been spared the role of its champion!! It could have been me that delivered the death blow.

It was during my almost three years with the SSC that I learned about EPICS. At that time the only users were the two original partners and the applications were relatively small. Like many others, I was at first extremely skeptical that it could be scaled to a facility the size of the SSC; but while competing approaches appeared much sexier at the top, I convinced myself that EPICS had it right at the infrastructure level. Persuaded that infrastructure was the most important issue, we eventually chose EPICS and had started to build small prototypical EPICS systems with the help of LANL when the project was cancelled. The estimate (undoubtedly low) was that there would be over 1000 input-output controllers (IOCs) in that system. It was somewhat of a

breakthrough for EPICS to have been selected for so large a project. Others noticed, and within a year EPICS had been adopted by several other projects.

4.1.6 The SNS

After the SSC debacle, I moved to Los Alamos National Laboratory. There I was briefly associated with another large project – Accelerator Production of Tritium (APT) – that soon met the same fate as the SSC. The Low Energy Demonstration Accelerator (LEDA) for APT was eventually built, but by then, although still at Los Alamos, I was working exclusively on SNS. One of the recommendations from the Department of Energy Conceptual Design Review of SNS was that the controls guy should move to the project site. To expedite the matter, another government agency arranged to have our Los Alamos house burn down (along with 400 others) in May of 2000, and so I made the final career-related move to Oak Ridge, Tennessee.



Figure 1: The SNS Control Room

We did very little technically innovative for the SNS. Once again we selected EPICS and followed a conservative approach. The concept of an “EPICS Tax” – that is a project’s software contribution to the EPICS toolkit and collaboration – was first stated explicitly for SNS (although it had always been a part of the EPICS model.) In this spirit, SNS either funded or developed a new display manager, an archiver and archive viewing tools, a graphical tool for database creation (with others) and an EPICS-compatible high-level environment (XAL) for the development of Java-based applications and accelerator physics models. SNS also developed an interface between Labview™ and EPICS that allows Labview™ to run in an EPICS IOC doing front-end data reduction transparently to EPICS. All of these tools are now extensively used at other EPICS-based facilities.

At the time beam was delivered to the target in April of 2006, the SNS control system included approximately 170 VME/VXI-based IOCs running VxWorks, 50 “soft” IOCs running in six Linux-based servers and 250 PC-based IOCs (“Network Attached Devices”) running Windows and Labview™ for beam instrumentation. In addition there were approximately 30 miscellaneous servers (Linux) and over 30 portable or fixed console servers having from one to six screens. The control room features 11 consoles in a circle, each with six ganged screens and a single mouse and keyboard (Figure 1). There were approximately 1000 inputs to the machine fast protection system. Altogether, there were ~400,000 EPICS process variables supporting an estimated 80,000 “channels” or input-output signals.

SNS was built by a collaboration of US National Laboratories, and the SNS control system was likewise built by a collaboration of teams distributed among the participating partners. Although the technology was conventional and conservative, the challenge was to make all of these separately-built systems work together. On a very small scale, this is the issue that is or will be faced by large internationally-developed physics control systems – notably for ITER and, hopefully in a few years, the ILC. The issues of an international collaboration in controls are far more complex and challenging than those for a single-nation multi-laboratory collaboration such as SNS, but some of the approaches and lessons learned might nonetheless be relevant. Although there were many other factors, three key approaches helped to make the SNS controls collaboration successful:

1. The selection of EPICS. Although not all participating laboratories were familiar with EPICS, all agreed to its use from the outset and training was performed at their respective sites. Commercial vendors supplying subsystems with controls, for example for the liquid mercury spallation target and for the conventional systems (HVAC, Power, etc), were also trained in EPICS and delivered EPICS systems. Systems were developed using agreed tools and incorporated into the software configuration management system at Oak Ridge. Commercial equipment was acceptance-tested at vendor sites using portable EPICS systems. As a result of following this protocol, it is fair to say that all delivered systems integrated without serious problems.
2. Other Standards. Although a few exceptions found their way through the cracks, the enforcement of standards was largely successful. Perhaps the most significant example was the case of Programmable Logic Controllers (PLCs), where one model of PLC was selected, and used almost everywhere. Vendors providing PLCs with their equipment were contractually required to use the selected model, and almost all did. This significantly reduced the complexity and resources needed for integration.
3. Centralized Management. Perhaps the most innovative and important step taken by SNS from the controls point of view was to recognize controls’ essential integration function by making it a Level 2 element in the Work Breakdown Structure (WBS). The top-level WBS elements were: Front End, Warm Linac, Superconducting Linac, Ring, Target, Experimental Facilities, Conventional Facilities and Controls. All controls funding flowed through the central organization in Oak Ridge where there was complete financial control. This was a tool for standards enforcement and also allowed flexibility in the allocation of resources where needed.

At SNS I was able to apply with various degrees of success certain lessons-learned and philosophies that served us well. Here just three:

1. Naming Convention and Technical Database. Get it right, and get it early. A well-conceived naming convention will facilitate the database. The convention should be based upon function, and should be developed with the interests of future operators foremost. Do not underestimate the effort required to design and implement the database. It's expensive, but it's worth it. We weren't completely successful here, but the experience only reinforced my sense of its importance.
2. Integrate Conventional Facilities. This is likely to be unpopular, but every project ends up needing the conventional data (power, water, etc) in the control system, so do it from the start.
3. No Hidden Data!! This one goes on my tombstone. People tell you: "We'll never need that data in the control room – I'll just turn it on with a local switch and check it locally from time to time." Or: "I can always read it on the local PLC screen." Don't believe them. Everything needs to come to the Control Room.

4.1.7 Conclusions

The idea of computer-based control systems for accelerators was not universally accepted when I started in this field in 1970. Early attempts had not been successful, and because these systems were not really expected to work – at least at first – there was always a requirement for a back-up conventional hard-wired control for each device. Today's far more complex systems not only MUST work (the accelerators are far too complex and powerful to run manually) but they are EXPECTED to work on Day One and, more importantly, they DO work when needed. Controls is recognized as an element essential to the success of any new facility, and is given appropriate management and oversight attention.

The development of a controls community with a literature and conferences of its own as well as the advent of toolkits and successful models for collaboration have both contributed to this change. So too has the incredible progress in computer technology over the past forty years. In that first TRIUMF system, all control system programs accessed in real time were stored on two 256 K word (!) fixed head discs. That's a total of one megabyte of memory in two \$5000 boxes that together filled a standard equipment rack. Recently I bought a cigarette-box sized 160GB USB back-up disk for about \$200. Using 1970 TRIUMF technology, my little home storage system would have filled 100,000 racks and cost one billion dollars. We've come a long way, baby!!

4.2 EPICS – Experimental Physics and Industrial Control System

Matthias Clausen, DESY
Mail to: Matthias.Clausen@desy.de
Leo Dalesio, BNL
Mail to: dalesio@bnl.gov

4.2.1 Brief History

The Experimental Physics and Industrial Control System (EPICS)[1] was developed beginning in 1989 at Los Alamos National Laboratory. The toolset approach was used to meet the challenging schedule that was set out in conjunction with a severe lack of requirements. We believed that using the several years before the equipment arrived to provide tools, would allow us to rapidly respond to the requirements as they appeared. In 1991, the Advanced Photon Source decided to adopt what we had started and made extensive changes to further develop clean interfaces at the various levels of the architecture and to make the software portable to other environments. The work created in this collaboration enabled the reuse and portability of EPICS. At the same time, groups from LBL, Duke University, and DESY all started to adopt EPICS for various projects. Now EPICS is in use at a significant proportion of the particle physics laboratories in North America, Europe, and Asia. It has successfully supported the construction, test, and integration phases of many accelerators. It has been an adequate platform for operation and automation. In addition, it has been used for astronomy, water distribution, electric distribution, CNG plants and other industrial applications.

A significant set of tools have been built on top of it's real-time front end and communication protocol and process control database: including many engineering display tools, data archivers, a state transition tool, modeling tools, scripting languages, and alarm loggers. The real-time front end has been ported to many platforms running a variety of operating systems: vxWorks, RTEMS, real-time Linux, Linux, and Windows. A large variety of instrumentation has been integrated through a large variety of protocols: industrial PLCs, industrial multiplexed I/O, GPIB and Ethernet aware intelligent devices, and high speed VME/VXI that cover the broad variety of instrumentation required to instrument a particle accelerator. A full complement of hardware to provide the special functions of machine protection, timing, and synchronous data collection have also been developed at a number of laboratories.

4.2.2 Architecture

EPICS is a distributed architecture that employs a client-server architecture with independent front-end processors. The network typically includes many file servers, operator consoles, and hundreds of front-end processors distributed over hundreds of meters.

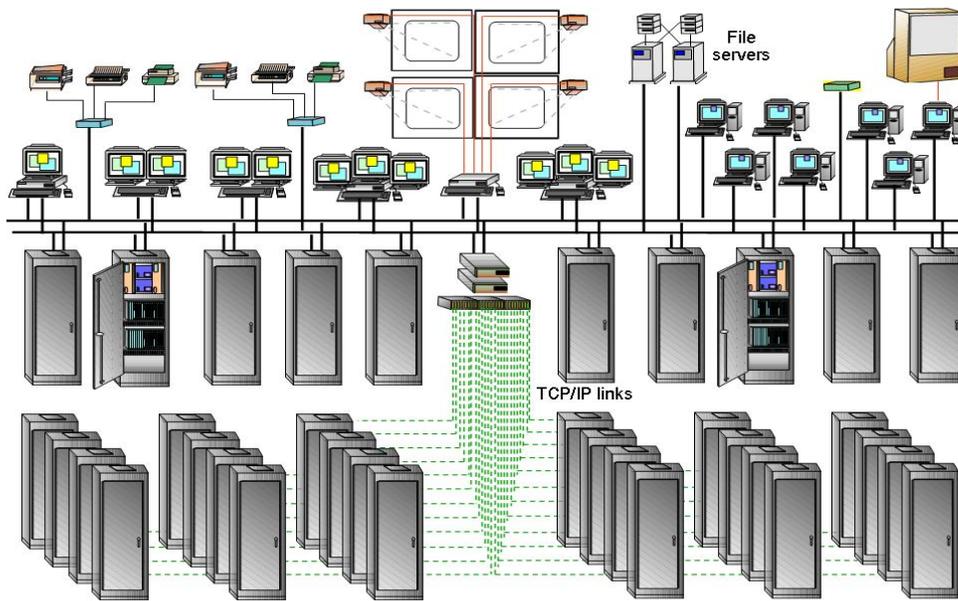


Figure 1: EPICS Architecture

The EPICS core software architecture consists of the Channel Access communication protocol and the Process Database. Interfaces are provided at every level of the EPICS architecture to support user extensions. Many standard extensions exist for operator tools, data archiving, plotting, hardware drivers, application specific tool sets, and straight programming language interfaces.

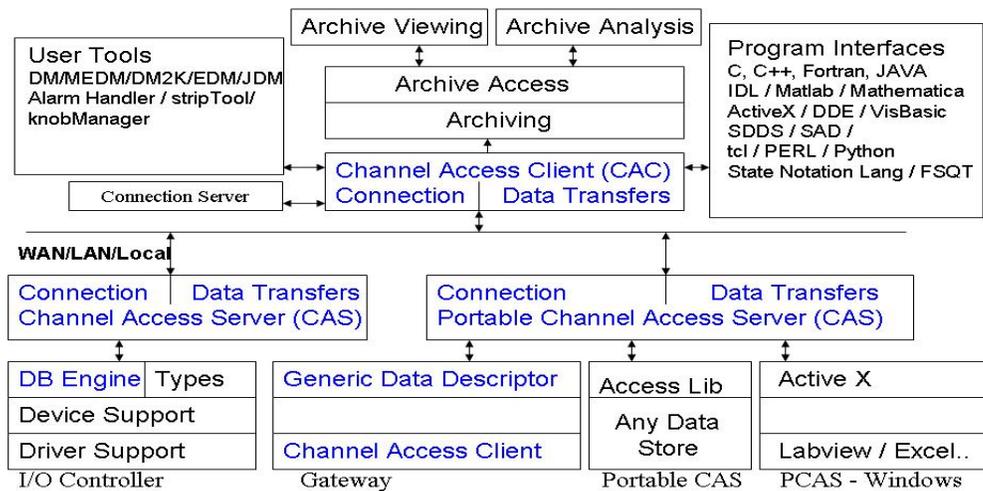


Figure 2: EPICS Core Software Architecture

4.2.3 Process Database

The core processing engine of an EPICS Input Output Controller (IOC) is the so called Process Database. The term Database is misleading and might be better called Engine instead. The Process Database consists of an extendable set of functional blocks called records. Each record type is implemented in 'C' and loaded in a record library to the IOC during boot time. A record configuration file describes the properties of each record in textual form. These files are called EPICS Database files. During IOC initialization the configuration files are parsed in and the record instances are created.

The core set of record types comprises a variety of input, control and output blocks. The basic set is already sufficient to implement a whole control system. Open interfaces to the process engine provide an easy way to add custom records.

All logic and local values are stored in the local process database and exposed to the entire network through the Channel Access Server.

4.2.3.1 Record Processing

One of the properties of each record defines the processing type for each record. A typical processing type is the scan mode. Several default scan times are defined. This set of scan times can be extended to match the individual requirements. Limitations are given by processor- speed and the tick rate of the operating system. The deterministic processing of e.g. PID loops is one of the strongest features of an EPICS IOC. Record processing can also be triggered by I/O interrupt - which is a second strong real time behavior of an EPICS IOC which is implemented in the EPICS core. Last not least records may be passive. In this case processing is activated by other records by activating (forward) processing links or by writing remote to a record from a channel access client.

4.2.3.2 Hardware Support

Hardware interfaces range from simple memory mapped devices such as VME to complex remote systems requiring communication tasks to handle asynchronous protocols. All known field bus protocols are supported in this manner as well as various kinds of Ethernet protocols. Highly specialized hardware – like timing modules – is in many cases already shipped with dedicated EPICS drivers.

4.2.4 Software Interfaces

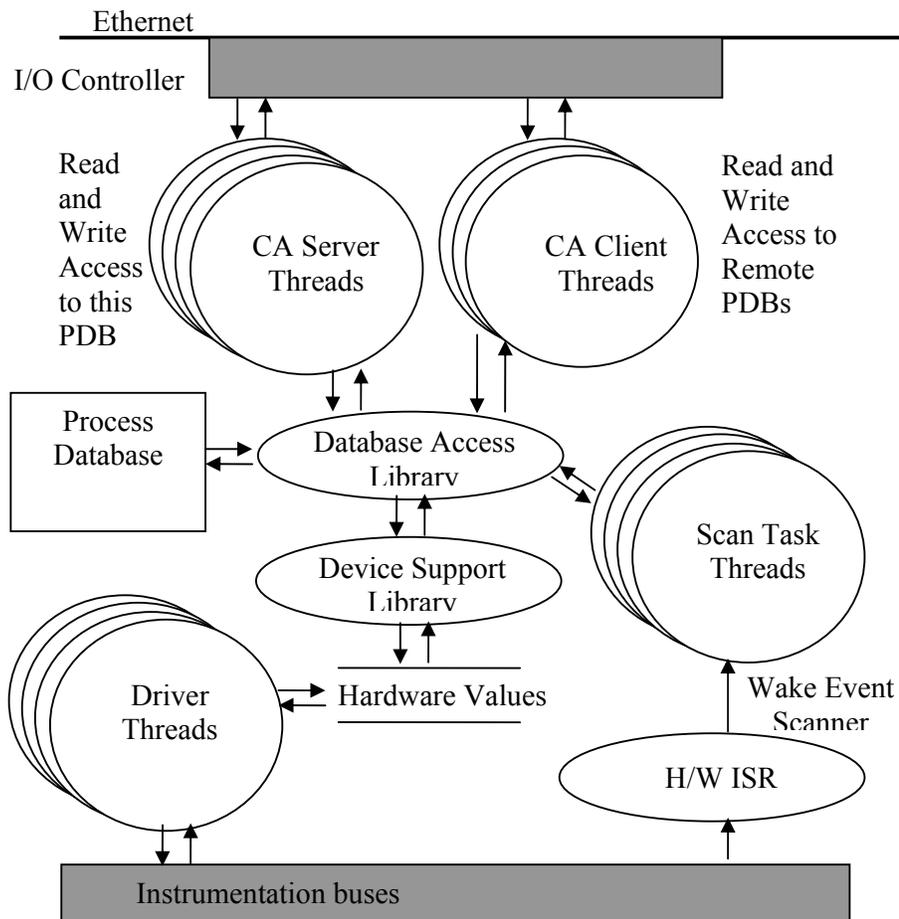


Figure 3: EPICS IOC Interfaces and Tasks

4.2.4.1 Driver Layer Interface

The driver layer interface is a straight forward interface that supports the easy integration of new hardware support. Nearly every laboratory in the EPICS collaboration has added hardware support. In addition to the clean interface, a framework for implementing asynchronous communication for Ethernet, GPIB, and serial devices is provided. The breadth of hardware support is too extensive to list. It includes most PLC families and most instruments used in experimental physics. If the device is not listed on the EPICS web site, a search through the EPICS email log for the hardware type will likely turn up the driver.

4.2.4.2 Record Type Support

The EPICS database library is a set of function blocks that can be configured to produce data acquisition and steady state control. There is support for analog, waveform, image, binary, and pulse train inputs. A large selection of records exists for outputs including analog, waveform, binary, pulse train, and position control. There are

many records that support conversions, closed-loop algorithms, data collection, and general logic. It is less common for function blocks to be added, but the function blocks available have been successfully applied to implement the data acquisition and local loop control by configuring logic, greatly reducing the amount of code that needs to be written, debugged, and maintained.

4.2.4.3 Channel Access Client Interface

The client interface is a narrow interface with well defined data-grams that provide connection to each field of each process block in any channel access server on the local or wide-area network. This allows dynamic modification of the set points, and outputs. It also allows modification of alarm limits, scan rates, conversion parameters, and input and output locations. To limit the potential anarchy, channel access has access control that allows each site to configure who, can modify which fields, from which machines in what state of the accelerator. This dynamic access control allows a strategy that could limit access to all control points to the control room during operation, but allow remote modification only during maintenance. The success of the Channel Access protocol is most obvious in the ability to run multiple versions, seamlessly on networks with hundreds of nodes.

4.2.4.4 Hardware Platforms

As the channel access protocol, process database, and drivers are written on an operating system independent layer, the EPICS I/O Controller has been deployed in a variety of platforms and processors. In addition to the widely used intel and Motorola single board computers, it has been run on PGA boards that are running embedded processor cores. In addition to the real time OS typically found in time critical processes, EPICS is run on windows as a way to integrate systems delivered in this architecture, and Linux for less demanding applications.

The EPICS I/O Controller is operating system independent so that it can be run on any platform including: VxWorks, RTEMS, LINUX, UNIX, Windows, etc.... allowing each laboratory to select hardware platforms that best meet their requirements.

4.2.5 Channel Access Clients and Operator Tools

The channel access protocol is initially implemented in 'C'. A Java implementation is available. Basic commands like 'get', 'put' and 'monitor' are available through a clean software interface in the channel access library. While RPC (remote procedure call) based protocols are limited to synchronous calls like get or put, EPICS provides the asynchronous monitor request which is the most efficient way to communicate between clients and servers. Three different kinds of monitors are available: A) Monitors for display purposes defined by the monitor dead band on the IOC; B) Monitors for archive purposes defined by the archive dead band on the IOC; C) Alarm Monitors defined by the alarm status and alarm dead bands on the IOC. All monitor dead bands are defined on the IOC and checked during each processing of a record. This reduces network traffic to a minimum – only when a monitor limit is reached - .

Two basic channel access client applications can be identified: Services and (operator) Tools.

4.2.5.1 Services: Archiving

Three basic functions define a good archiving system: Fast archiving, fast retrieval and long term storage with fast access. As a consequence several implementations are available which focus on different aspects of the three. While archiving into binary files guarantees fast write performance, archiving into relational databases provides easy access at reduced write speed. Both implementations are available. It is left open to the user to pick the desired implementation.

4.2.5.2 Services: Alarming

The most commonly used alarm tool in the EPICS community is the so called alarm handler (alh). This tool combines alarm configuration with alarm actions and is implemented as an X-Window application. The advantage of this approach is the very efficient approach to use the channel access protocol for alarming and the fact that only one tool is in charge of alarms. The disadvantage is the fact that this functionality is only available with a running X-Window application. (see current plans).

4.2.5.3 Services: Logging

Besides error logging from the IOCs it is possible to log write requests to the IOC database. Log entries are written to disk on a workstation running a log server.

4.2.5.4 Operator Tools: X-Window/Motif Based Applications

As mentioned above, the alarm handler is an X-Window application running on a workstation. Several applications are available to display synoptic information. The most recent one is called EDM – the extendable display manager. Its predecessor are the Motif based display manager (MEDM) and dm2k. While several methods are available to store archive data there is only a limited number of applications available to display them. One of them is the stripTool with archive extensions.

4.2.5.5 Operator Tools: Java Based Applications

While ‘C’ application with X-Window displays were used at the beginning – and are still used in most EPICS places – new applications are written in most cases in Java. This opens a new regime of functionality and portability. One of the prominent applications is the XAL Accelerator Framework [2] which is implemented in Java. Other applications were following like an archive viewer and correlation plot. The latest developments are based on Eclipse like the Control System Studio [3] (see also **1.1.8.3 Operator Tools**).

4.2.6 Applications & Their Requirements

EPICS was developed as a tool set for accelerator control. There are many commercial systems available. Special features of EPICS to meet requirements for accelerators are: open interfaces to integrate and add new hardware and physics applications at every level. This has allowed each site to add whatever hardware was needed regardless of communication bus or platform. From the beginning, event driven process control was included to support synchronous data acquisition, low latency responses (35 usec for interrupt, task switch, and record processing). Time stamps have

always supported integration of an event system. Data synchronization to within 1 nsec is supported in software and has built in support for integrating whatever timing system or alternate communication bus was provided for high speed, low latency applications such as fast orbit feedback. This is pictured below. The communication protocol supports point to point connections seamlessly so that steady state control can be accomplished in the I/O Controllers without some hierarchical processor being involved. The communication protocol is optimized for performance as a nod to the reality that Channel Access does not meet the communication needs, but makes the most optimal use of COTS network hardware. A good example of the success of this interface is the ability to build a High Level Physics Application Environment on top of this protocol. This is pictured below as Figure 4.

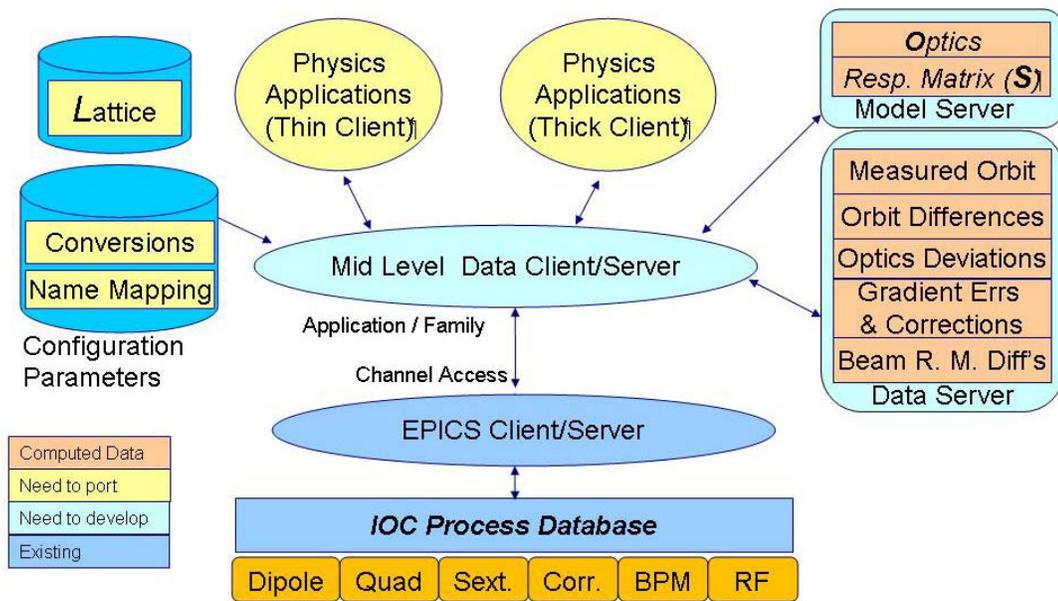


Figure 4: EPICS High Level Applications Architecture

4.2.7 Robustness and Reliability

4.2.7.1 Robustness

Robustness in a process control system is of the utmost importance. Integration of PLCs and the implementation of redundant I/O Controllers provide a fault tolerant front end capability. Software and communication issues must also be robust. When a front end process is no longer communicating for whatever reason, all clients need to be notified in a reasonable and predictable amount of time. The reconnection also needs to be handled in a uniform and reliable way. There is more to consider than just failures. There is also the problem of data being produced in bursts that cannot be consumed by the client. Several things are implemented to handle these situations. There is a heartbeat and beacon that assure that EPICS clients are notified within a user defined period of time when some server is no longer communicating, there are buffers at both

the client and server side for handling bursts of data, there are server tasks for each client so that the failure of one client to process the data stream does not affect any other client and there are careful reviews in the various clients of how to handle data – for instance operator interfaces simply cache new data and only display the most recent value where the archiver will queue the data and try to handle all data. When the clients are not keeping up with the servers, the degradation mode must be carefully handled.

4.2.7.2 Reliability

The reliability of the EPICS core implementation is constantly proven in more than hundred installations comprising several thousand I/O controller running hundreds of thousands of record instances. The concept of functional blocks – or records – is demonstrating its powerful approach. At least core record types are running under various circumstances in multiple instances. One can assume that these functional blocks will also run in a new instance as reliable and bug free as in the numerous instances before. The concept of proven software components based on well defined software interfaces is showing its strength compared to custom programs where each program will have to run through a long commissioning phase before it will reach the robustness of proven functional blocks.

4.2.7.3 Redundancy

To extend the reliability of the front end controllers beyond the proven measures, redundancy support has been added to the EPICS IOC core functionality. The implementation serves two purposes:

- Extended reliability by IOC switchover in case of failures in hardware or software
- Support for 24/7 operations:
IOCs running all year through need to be maintained. Hardware maintenance on fans or power supplies often require CPU shutdowns and reboots as well as software maintenance like security updates on the operating system.

The current implementation supports both aspects of redundant front end controllers. While the implementation to synchronize and update process databases and sequence programs is EPICS specific, the supervisor called Redundancy Monitor Task (RMT) is EPICS and operating system agnostic.

4.2.8 Current Plans

4.2.8.1 EPICS Core: Java IOC

EPICS is a channel based system that requires that each channel of each device be configured and controlled. In the Java IOC project, we prototype a device oriented extension to the distributed channel database. The ability to implement this was demonstrated by prototyping a device aware version of an EPICS I/O controller that functions with the current version of the channel access communication protocol. Extensions have been made to the grammar to define the device oriented database. Only a multi-stage position controller with limit switches was developed in the demonstration, but the grammar should support a full range of functional record types. In the current stage of this project, a full set of record types are being developed to

support all existing record types, a set of process control functions for closed loop control, and support for experimental beam line control. A tool to configure these records is planned. A new communication protocol or extensions to Channel Access will be made to support introspection of components of a device. Performance benchmarks on both communication protocol and the database are planned as part of this project.

The proposed approach to configuring the I/O for a large physics project reduces the time to produce and maintain the massive amount of configuration data by providing the engineers and physicists with tools that configure the system at a device level. The creation, duplication, and management of devices, compresses the external view of the process by an order of magnitude. At the completion of this project, the database engine, record types, configuration tool, and communication protocol provide the fundamentals needed to make a beta release of the Java IOC to potential users.

4.2.8.2 New Directions in Channel Access

Channel access has been in operation and backwardly compatible for many years. We have identified a need to support major new functionality for this device oriented database. The appeal to channel based protocols is that the interface and reusability of clients is clearly demonstrated. Attempts at standardizing on object hierarchies and methods have not been widely reported or reused. To support the ability to introspect the data to determine which channels provide which functions, the Java IOC project includes a prototype extension that supports introspection. Extensions to meta-data must also be defined to support some of these functions. These extensions are needed to support model based steering and distance plots at a minimum.

4.2.8.3 Operator Tools

Most (if not all) of the new applications are written in Java. While individual Java applications are using the AWT/ Swing approach, the SWT/ Jface implementation is used for pluggable applications which are based on the Eclipse framework. Besides the graphical aspects of these two directions it is the pluggable/ collaborative aspect of the Eclipse framework which is favorable for a new development strategy in a collaboration like EPICS. Developing new applications (or plugins as they are called) for the Eclipse-based CSS ecosystem is reducing the development effort dramatically. Besides this technical aspect it is also noteworthy that Eclipse plugins are lazy loaded (not completely initialized during startup) and only require *one* virtual Java machine (VM). This reduces the overhead of one VM for each application in other cases.

4.2.8.4 Web Based Applications

Besides applications running natively on workstations, Web based applications are in use to monitor and control EPICS IOCs. New developments even support the native Channel Access protocol on the Web client. Future will show how far this development will be able to replace applications like synoptic displays running natively on the workstation.

4.2.8.5 *Services*

Writing so called headless application (without native display) in Java is nowadays common practice. The advantage being that a common infrastructure can be shared between the display based applications and the server based applications. Archiving and alarm services are implemented this way.

4.2.9 **Industrial Support**

Industrial support has evolved in the EPICS community. There are now a number of hardware vendors that sell their hardware with EPICS drivers such as: National Instruments, Hytec, ITech, ZTech, and Cosylab. It is possible to purchase entire subsystems that are delivered with EPICS controls such as: turn key linacs, turn key experimental beam lines, and facility control systems. It is also possible to purchase tool development, project engineering, or embedded engineering from a number of companies and contractors in the US, Europe, and Asia. The ability to acquire EPICS aware hardware or augment staff for a project, from industry has grown steadily over the last 5 years.

4.2.10 **The EPICS Collaboration**

Starting from two collaborating institutions - Los Alamos and Argonne – the initial development team has evolved to an ever growing collaboration of hundreds of developers and users in over hundred institutions. It is not easy to describe how the collaboration works. There is no CEO and not even no professional management team. There is a bunch of international specialists which make sure that the core software is clean and that new developments are well tested before they get integrated into the EPICS core. On the other side there is a lot of freedom for each individual site to extend the existing functionality by using the well defined interfaces on the client and the server side.

An example might help how the collaboration works: Redundant EPICS IOCs are necessary to run installations in 24/7 mode for a year or longer. Since this requirement was unique to one EPICS site there was no way to share the development between several collaborators. The requesting site started the development on the current EPICS core version. A lot of support from the collaboration was available to specify and integrate the redundancy support into the core software. After final acceptance tests the implementation had to be retrofitted into the next release. A so called code-athon at Argonne (about 14 EPICS developers from all over the world for one week) worked on the next EPICS release including the final integration of the redundancy code. This describes a clear Win/Win situation where a completely new functionality was provided for free from one collaboration member and a set of specialists from other collaboration members integrated this code into the next release. This way redundancy became a core functionality supported in this and in future releases and the collaboration gets free access to this new feature.

The collaboration is coordinating its work in two meetings each year which are taking place in any place where it is useful. In-between these meetings most of the communication is carried out through the tech-talk mailing list at Argonne [4]. It is worth mentioning that the response time on this list is typically less than an hour – unless a special person is required to answer a question.

4.2.11 Conclusions

For device control, the architecture has proven to be very effectively applied to a variety of accelerators, experiments, telescopes, and industrial applications. The operator tools continue to evolve to reflect the desire to come up with a set of interoperable tools that provide the operator with a similar look and feel. New hardware continues to be integrated without trouble. Many hardware manufacturers are providing EPICS driver as part of the hardware offering. In recent years, attention has been paid to supporting any operating system and robust operation. There is significant progress in the area of integrated operator tools with the emergence of CSS. Application specific tools being developed for machine control, experiment control, and telescope planning give an indication that more is needed to meet the requirements of the specific applications. Work being done on version 4 of EPICS is intended to meet these needs in a similar manner: open, narrow, and extensive interfaces that are robust and have optimal performance.

4.2.12 References

1. EPICS home page, <http://www.aps.anl.gov/epics/>
2. XAL Accelerator Framework, <http://xaldev.sourceforge.net/>
3. Control System Studio, http://css.desy.de/content/index_eng.html
4. EPICS tech-talk mailing list, tech-talk@aps.anl.gov

4.3 THE TANGO CONTROL SYSTEM

Andy Götz, Jean-Luc Pons, Emmanuel Taurel and Pascal Verdier, ESRF
 On behalf of the Tango Collaboration
 Mail to: Andy.Gotz@esrf.fr

4.3.1 History

Tango is the second generation of control systems to originate at the ESRF. It was started as a new development in 1998² based on fifteen years of experience gained developing the Taco control system. Taco was the first control system to pioneer the device server concept.

Tango has taken the good points of Taco and added some missing and many new features. Tango was first presented in 1999 at the ICALEPCS conference in Trieste³. Tango was developed first at the ESRF, then jointly with the SOLEIL synchrotron in Paris. About a year later ELETTRA in Trieste joined the collaboration. The adoption of Tango by 3 institutes was a decisive step in the evolution of Tango. It ensured the future of Tango and opened the way forward for other institutes to follow. ALBA and PETRA 3 (the beamlines) followed suite. With five institutes the critical size was reached needed to make Tango evolve and improve. The next logical step in the evolution of Tango would be for a large non-synchrotron institute to adopt Tango to extend the application of Tango to new domains.

² <http://www.tango-controls.org/History/tango>

³ <http://www.tango-controls.org/Documents/papers/icalepcs1999.pdf/>

4.3.2 TANGO Philosophy

The approach Tango has taken to building control systems is to wrap all parts of the control system using a common wrapper technology called the device server model and let all parts of the control system including the clients communicate via the network using the same protocol. Control systems built with Tango develop device servers for their hardware or software services in one of the languages supported for device server writing in Tango i.e. C++, Python or Java. Users of a Tango control system mainly see high-level applications or use their favourite scripting language to sequence their application. In some cases users also write applications in compiled languages like Java or C++. Refer to Figure 1 for a schematic view of a typical small Tango control system.

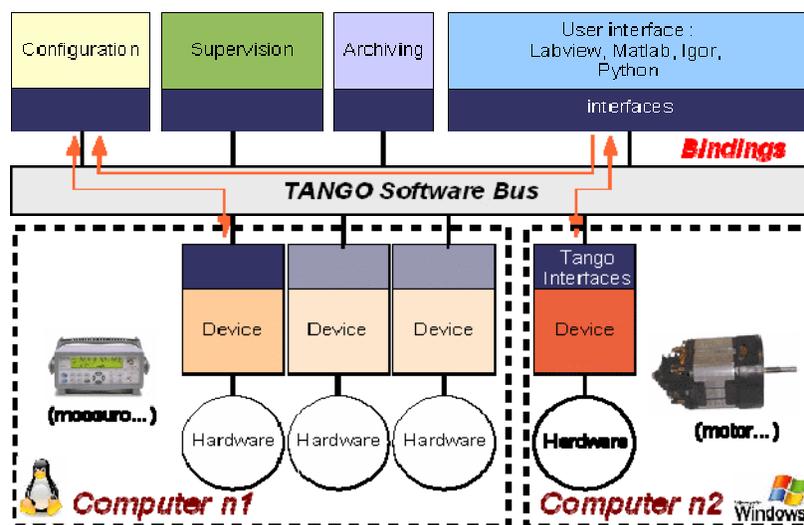


Figure 1: Tango building blocks (cf. Oliver Tache <http://iramis.cea.fr/scm/lions/tango/>)

4.3.3 CORBA

CORBA⁴ is one of the more recent binary protocols which is managed as a standard. The Object Management Group (OMG⁵) manages the CORBA standard (and other common standards like UML, MDI etc.) CORBA defines more than one network protocol. Tango uses CORBA's IIOP protocol. CORBA is object oriented. The objects are in fact interfaces. The interfaces are defined in a language called IDL. For a brief introduction to CORBA refer to the online book “CORBA explained simply.” The CORBA protocol is widely used in business and industrial control. It offers a rich set of functionalities based on a standard. A number of free open source implementations exist.

⁴ <http://www.corba.org/>

⁵ <http://www.omg.org/>

Tango uses the omniORB⁶ implementation. Tango hides all details of the underlying network protocol (CORBA) from programmers and users alike.

4.3.4 TANGO Protocols

Tango defines a CORBA interface called a device. All servers end up implementing devices. Clients import the devices and communicate with the servers using CORBA. We see Tango being extended in the future to support multiple protocols. The Tango Java libraries have recently been extended to support the SOLEIL http web protocol. This allows transparent access across firewalls. In the future Tango will investigate adding support for multi-cast protocols. Various multi-cast protocols e.g. socket-based, ICE and DDS, are currently being evaluated as possible candidates as optional protocols in the future in addition to the CORBA IIOP based protocol currently implemented. Tango will continue to support CORBA IIOP but could have a plugin type architecture for supporting other protocols in the future.

4.3.5 Performance

Tango is an object oriented control system. This means control points are either data belonging to an object, actions on an object or groups of objects. A single command sent to an object or group of objects can result in a series of changes e.g. a simple ON command can result in an entire orbit correction. For this reason it is sometimes tricky to uniquely quantify performance in a way which it can be used to compare systems. However a good figure of merit is the number of data values which can be read per second and the number of bytes which can be transferred via the network per second. These figures should be considered in the case of synchronous, asynchronous and event-driven communication modes.

The following table summarizes timing measurement done in synchronous communication mode. These measurements have been made in three different cases:

1. **Case A** is the time needed by a client process to execute a command on a remote device with one input argument (a 32 bit data) and one output argument (also 32 bits data). The returned data is simply the input one.
2. **Case B** is the time needed by a client to execute a command on a device with no input arguments but returning 2,000,000 32 bit values (i.e. 8,000,000 bytes). The returned array initialisation is not taken into account in this measurement.
3. **Case C** is the time needed to read a Tango attribute (a data corresponding to a device physical value) when the attribute is an array of 1024 32 bits data.

These measurements have been done in two different setups:

- **Setup 1** is when the client process and the server process are running on the same host. This computer is a two dual core AMD Opteron running at 1.7 GHz with 8 Gbytes of memory.
- **Setup 2** is when the client is running on one host and the server process on another host with the same hardware characteristics. The network link between these two computers is a 1 Gbit link.

⁶ <http://omniorb.sourceforge.net/>

Table 1: Synchronous communication

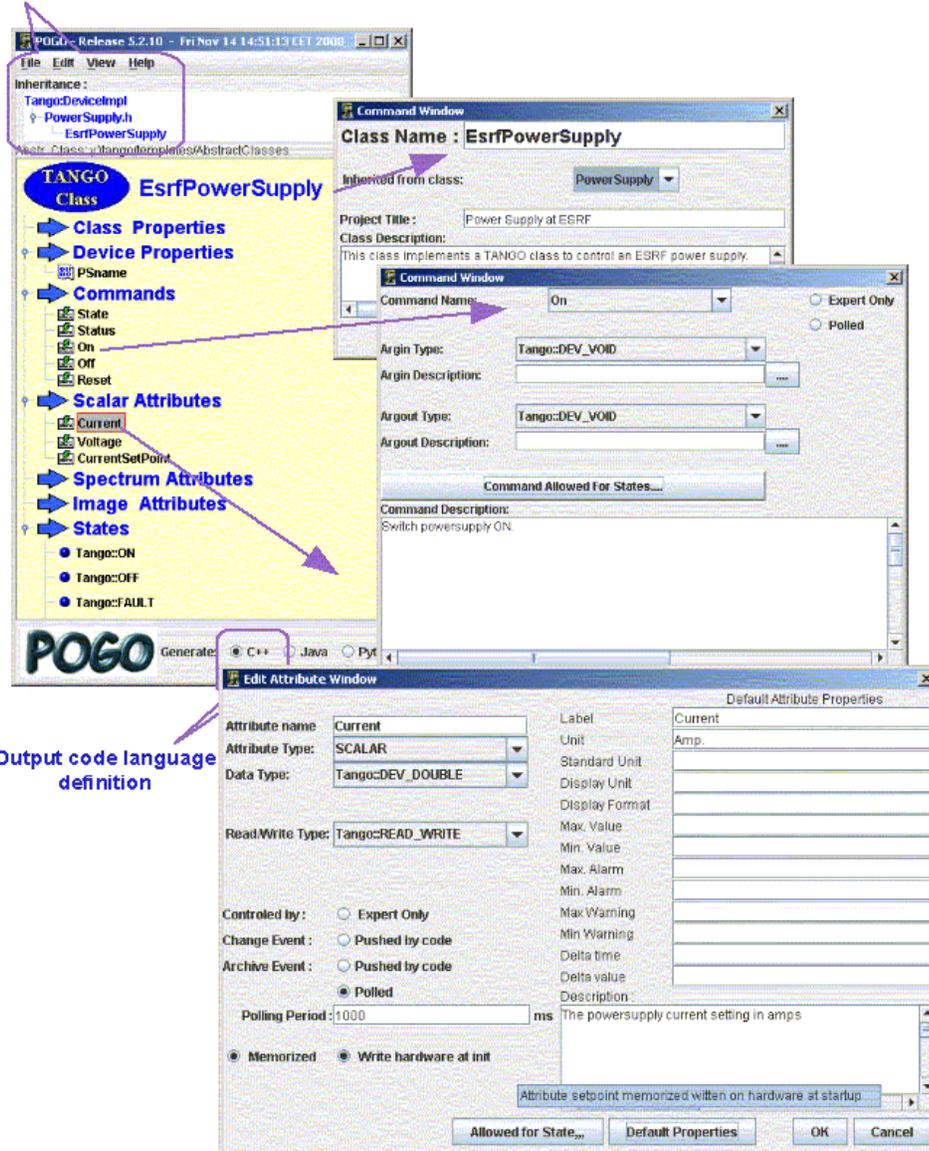
Experiment set-up	Case A	Case B	Case C
Setup 1	50 uS	56 mS	100 uS
Setup 2	115 uS	110 mS	250 uS

In event-driven mode, Tango needs an additional process to notify the event to all the clients interested in the event. With the server and this additional process running on a host and the client running on another computer (the same hardware as previously described), it is possible to send *50 events of one attribute of 1 Mbyte of data in one second.*

4.3.6 Code Generation

A graphic tool, called **POGO** helps developers to generate device classes code in C++, Java and Python. The generated code is a skeleton used by developers to add hardware access routines. **Pogo** is able to re-read source code for future modifications or updates. It is able to generate **Linux** make files and **Win32** project files.

Inheritance from
an abstract class



Output code language
definition

Figure 2: The Pogo tool

The **Pogo** tool allows the developer to define:

- The programming language.
- Class name, description and inheritance.
- Class properties (type and description)
- Device Properties (type and description)
- Commands (input and output argument types and description)
- Attributes (type and description)
- States (description)
- State Machine.

The code is generated including descriptions as comments. An HTML documentation is also generated using the descriptions.

4.3.7 Administration

The couple [Astor/Starter](#) is used to administrate a Tango control system (starting or stopping device server, checking them...):

- **Astor** is a graphical application to have an overview of the whole control system.
- **Starter** is a device server running on each host. It gives the status of the host and the status for all **Tango** device servers running on it.

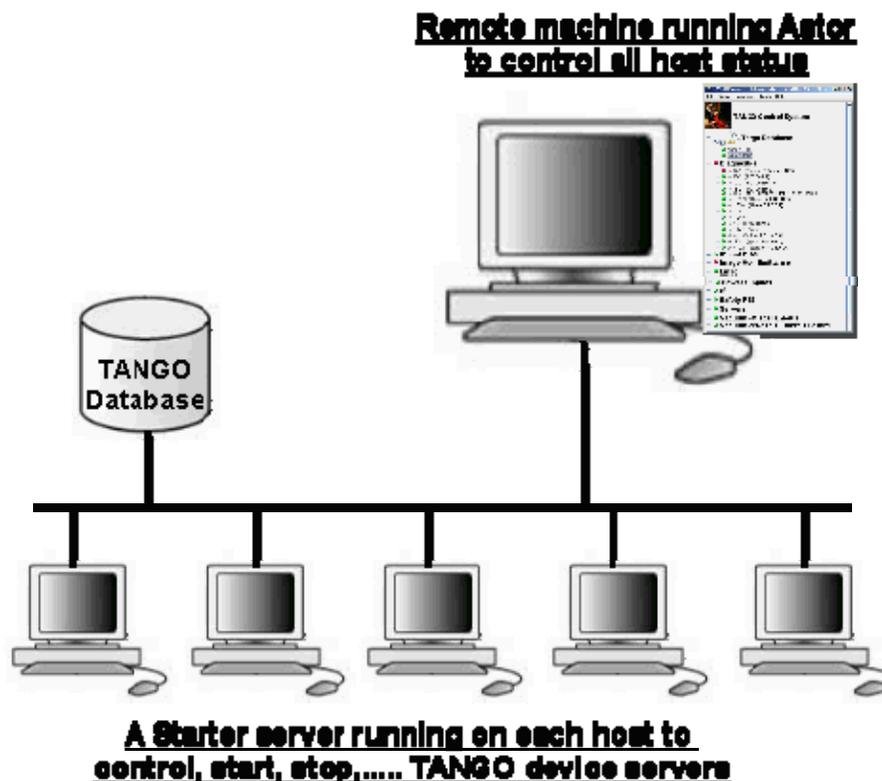


Figure 3: Astor/Starter

Database browser

The screenshot displays the TANGO Database browser interface. On the left, a tree view shows the database structure under 'Tango Database', including categories like 'Diagnostics', 'ID and DIAG', 'Image Processing', 'Laboratories', 'Linac', 'Miscellaneous', 'Power Supplies', 'Safety PSS', 'Servers', and 'Vacuum - Miscellaneous'. A purple arrow points from the 'Servers' category to the 'Servers Controlled on orion' dialog box. The dialog box shows a hierarchical list of servers organized into five levels, with a 'Dismiss' button at the bottom. A legend titled 'Servers' explains the status indicators: a green dot for 'Server is running', a blue dot for 'Server is running but not alive (Starting?)', and a red dot for 'Server is not running'. Another legend titled 'Hosts' explains the status of the host: a green dot for 'All controlled servers are running', a blue dot for 'Starter is starting server(s)', a yellow dot for 'At least, one controlled server is stopped', and a red dot for 'Starter is not running on host'. A 'Database browser' window is also visible in the background, showing device information for 'orion'.

**Device server status and Control
(Start, Stop, Test, Info,)
on host panel**

Figure 4: TANGO Database

4.3.8 Graphic Tools

The Tango Application ToolKit, also called “ATK,” provides a simple client framework for building applications. It is based on a MVC model using the Java Swing platform and the Tango Java core (TangORB). The main goals of ATK are the following:

- Speeding up the development of Tango graphical clients.
- Standardizing the look and feel of Tango applications.

ATK provides several Swing based components to view and/or to interact with Tango device attributes and Tango device commands and also a complete synoptic viewing system. ATK takes also in charge the error handling and display. The ATK Swing components are Java Beans, so they can easily be added to a Java IDE (like NetBeans) to speed up the development of graphical control applications.

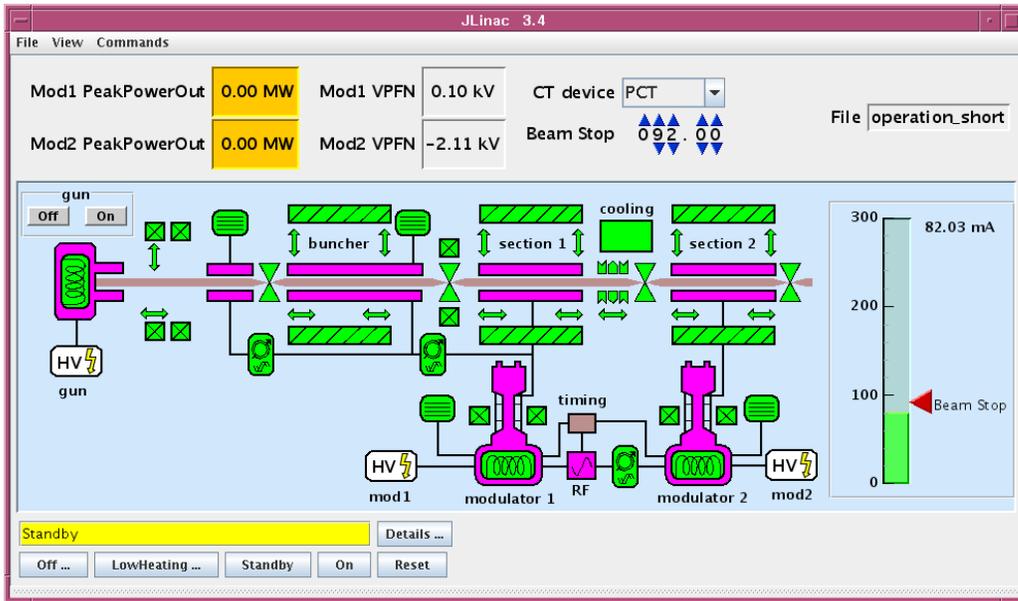


Figure 5: The linac control application used at the ESRF built with the ATK synoptic editor Jdraw and NetBeans.

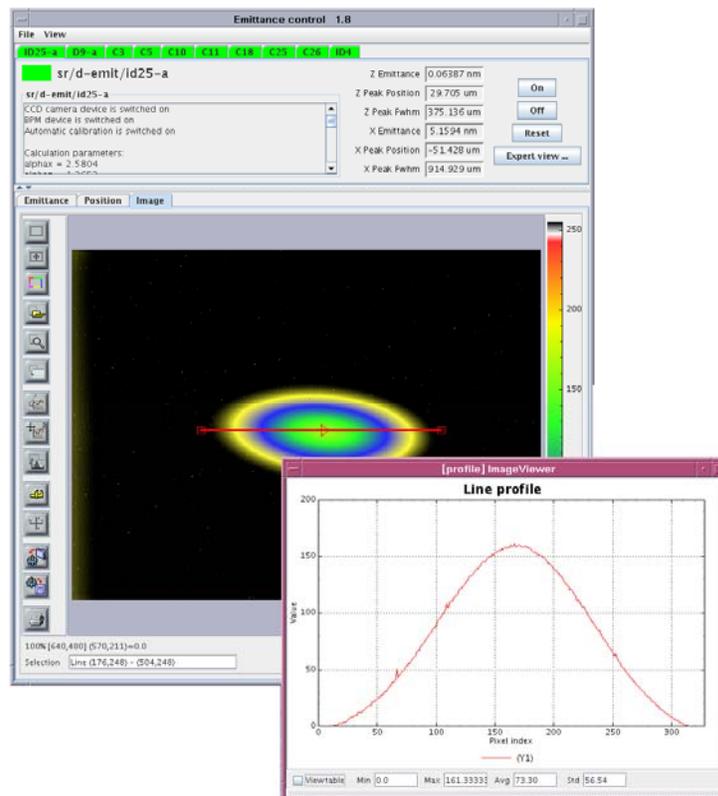


Figure 6: The emittance application used at the ESRF. This example illustrates the usage of the ATK image viewer and of the ATK chart (JLChart).

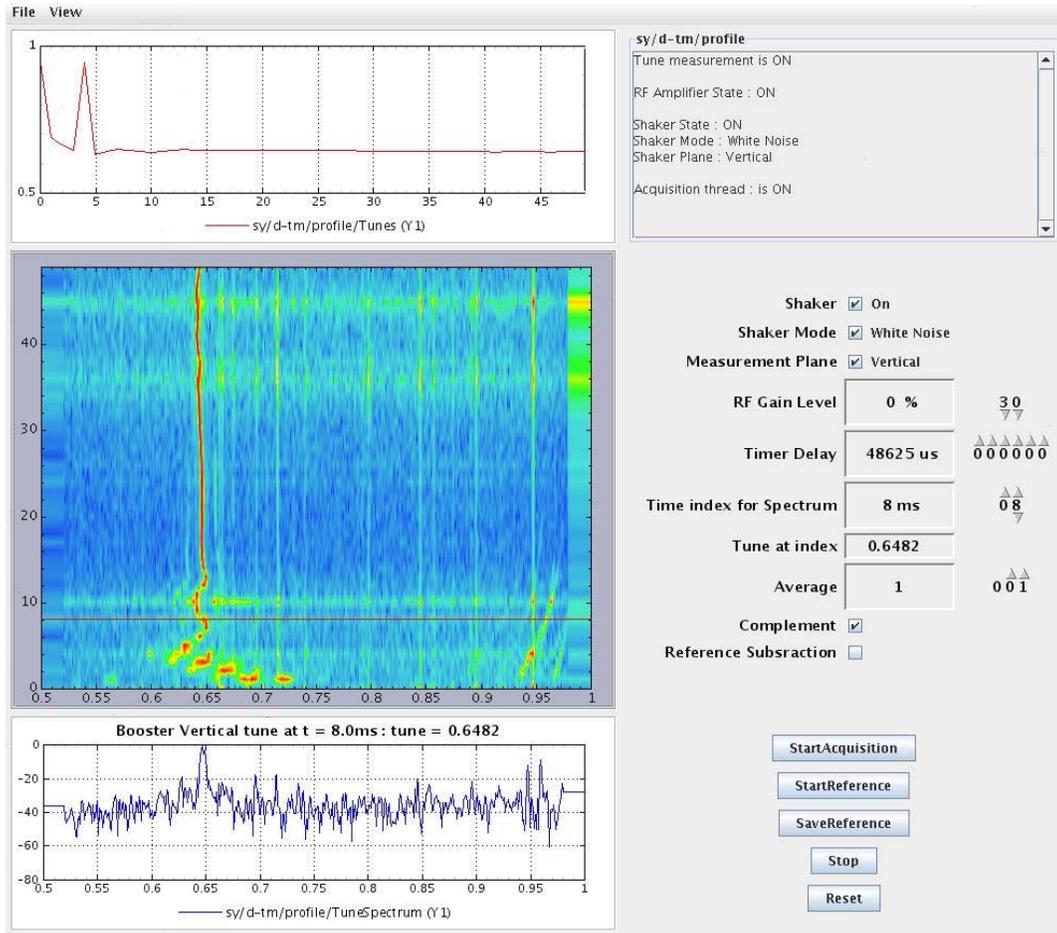


Figure 7: The Booster tune application used at the ESRF.

4.3.9 Diagnostics Applications

4.3.9.1 Vacuum

The ESRF storage ring vacuum system controls:

- 580 gauges
- 900 thermocouples

A Java swing application collects device states and display them on tree cell by cell:

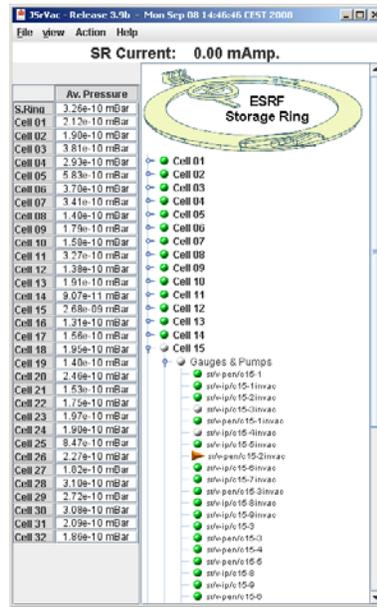


Figure 8: SR Vacuum

It collects also measurements (pressures and temperatures) from tens of device servers and displays them with different methods:

- A storage ring profile with average cell by cell.

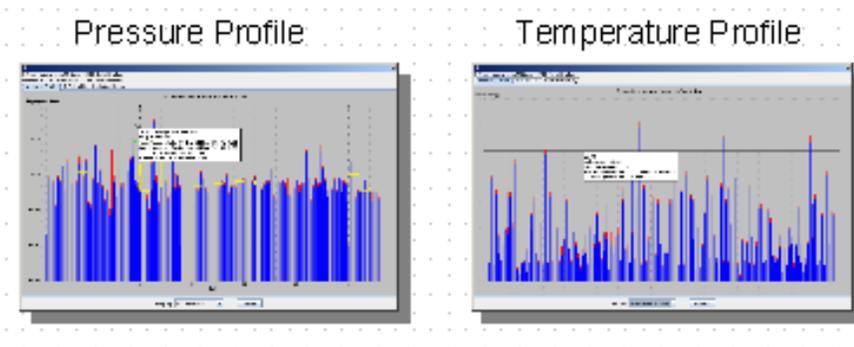


Figure 9: Storage ring profile

- 2D plot of “derivatives” (P_t/P_t-dt) over one hour shows changes much quickly than linear plots. It allows seeing the evolution of the vacuum dynamically on the day to day operations.

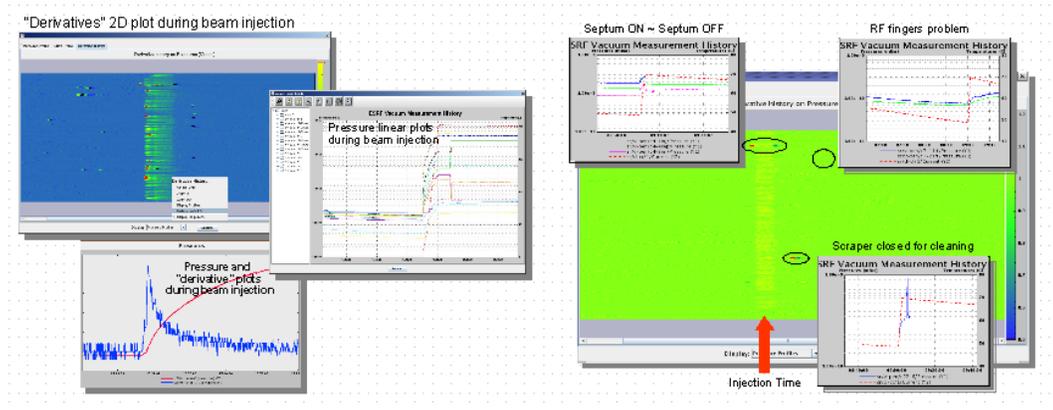


Figure 10: Derivative 2D Plot

- A leak detection tool. A continuous comparison to the saved reference. Each measure is compared to the saved reference, taking in account the filling pattern mode and the storage ring intensity to foresee potential problem and help to plan maintenance.

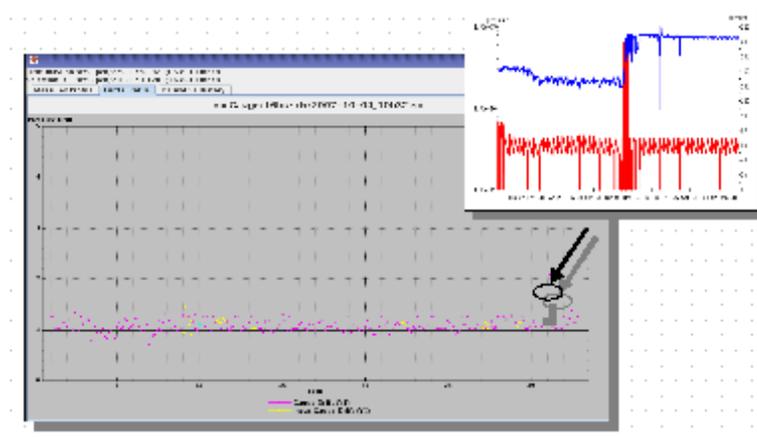


Figure 11: A leak detection tool

4.3.9.2 Closed Orbit

The closed orbit stabilisation of the ring is performed using a slow and a fast system. The slow system, which includes more than 200 beam position monitors, operates in both the horizontal and vertical plane and corrects the closed orbit every 30 seconds. The slow system is efficient for correcting the orbit drift during current decay. The fast orbit correction system operates in the .1Hz to 150Hz frequency range and corrects beam motion induced by quadrupole and girder vibrations.

The fast correction is achieved by a Tango server running on a Windows operating system. It manages communications with a TI C40 floating point DSP which handles calculations. It also manages archiving of 196 position and angular signals. Fast steerers are controlled by 16 Tango servers.

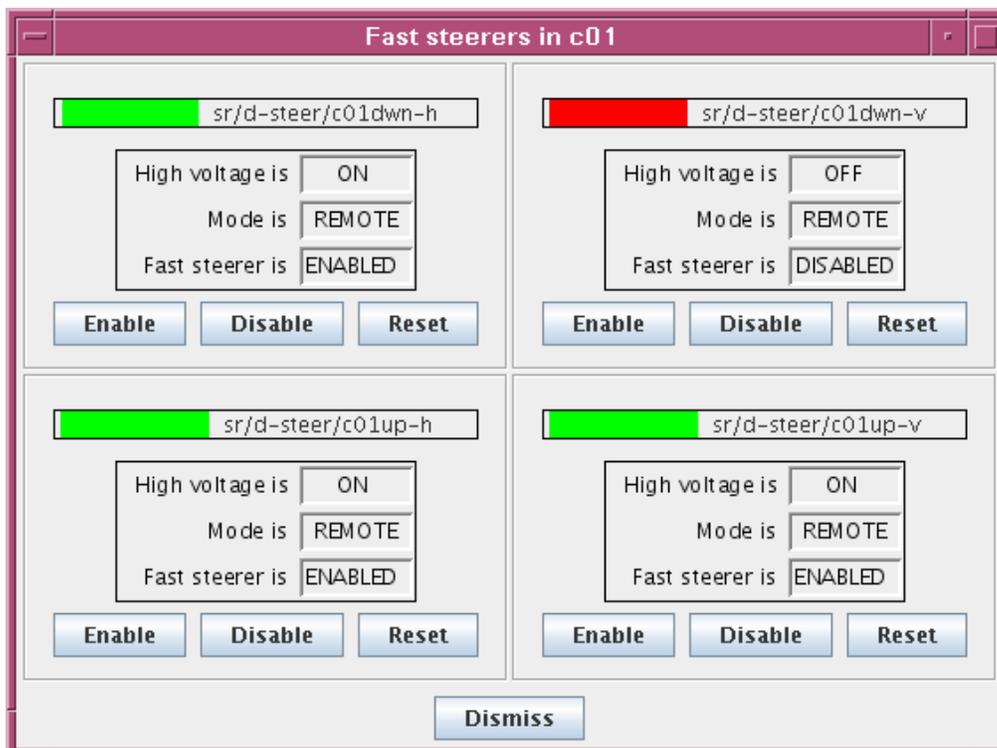


Figure 12: Fast steerer control panel (Cell 1)

4.3.10 Current Status

Tango is just over 5 years old and is been adopted by 5 institutes in the following domains:

1. ESRF – accelerator and beamline control
2. SOLEIL - accelerator and beamline control
3. ELETTRA – accelerator control
4. ALBA - accelerator and beamline control
5. PETRA III (DESY) – beamline control

The institutes share the following in Tango:

1. the CORBA protocol
2. the device server model
3. the database
4. management tools
5. navigation + test tools
6. common device servers
7. a tool to generate device servers
8. an archiving database

But the institutes do not share:

- Device servers for institute specific hardware

- Graphical user toolkits for building user interfaces
- Domain specific applications for accelerator physics, beamline control, online data analysis

Tango is very flexible and does not impose constraints on the choice of hardware or the operating system (Tango runs on Linux and Windows). Due to this flexibility and the fact that there is a huge variety of hardware to choose from, each institute has made different hardware choices. This means that in practice an institute which adopts Tango invests a significant amount of its resources in writing device servers. This could be avoided by sharing hardware choices between institutes but the advantage of choosing hardware based on institute specific criteria is mostly seen as an advantage.

This is similar for the choice of graphical user interfaces. Tango supports language bindings to the following languages:

- C++
- Java
- Python
- Matlab
- Labview
- Igor

Each institute has expertise and preference for a subset of these languages and their graphical toolkits. This means that it is difficult to agree on only one single graphical toolkit. In practice this results in each institute creating and maintaining their own graphical user interface.

Domain specific solutions are what the end-user is most interested in. The aim of the Tango community is to share as much as possible in this area.

4.3.11 Constant Evolution

Tango has been constantly improved since its inception. A major release of Tango has been made on average once every 12 months.

For the next few years *Tango* will concentrate on : services, beamline control, embedded systems on FPGA, 64-bit support, scalability for large systems, faster boot performance, enhanced Python and Java support for servers, more model-driven development, and integrated workbench-like applications. The aim is to keep powering *Tango* so that it remains a modern, powerful control system that satisfies not only the needs of light-source facilities but other communities too.

4.3.12 TANGO Feature Requests

In order to identify and track the evolution of Tango the Tango Feature Request system has been proposed. New features to be added to Tango are registered as ***Tango Feature Requests*** (TFR's). A total of 21 TFR's were identified in 2006 as the result of a meeting held in the French Alps to discuss the future of Tango. The TFR's can be found

in a paper presented on the “*Future of Tango*” at ICALEPCS 2007 (a copy of the paper can be downloaded from the tango-controls.org website⁷).

Two years later approximately half of the TFR's have been implemented. Most of the remaining ones are either currently being implemented or have been dropped. In the meantime news ones have been identified. Priorities of developments are discussed at the regular Tango collaboration meetings.

4.3.13 New Features

Here are some examples of new features which have been requested and added to Tango in the recently:

- (1) Tango data types are raw types in the sense that they represent sequences of simple data types with a minimum of information on what the data mean e.g. voltage, current. A new data type has been added to the latest version of Tango the so-called “encoded” data types with allow pre-defined data representations e.g. JPEG, TIFF, YOUR-FAVOURITE-FORMAT to be transferred.
- (2) Tango uses a polling thread to generate events. Up till now only one polling thread per server. The latest version of Tango implements a major performance improvement for device servers with more than one device by extending the polling thread to one per device.

4.3.14 Collaboration

Collaboration is the key to the continuous evolution of the Tango control system. But collaboration does not always come naturally due to different local requirements in each institute. For this reason it is important to constantly seek ways to strengthen the collaboration and actively work on common projects. Improvements to the Tango core would be implemented more efficiently if the collaboration financed one or two dedicated system developers.

4.3.15 Stability, Quality, Packaging

Stability is an essential part of any software system and even more when it is used to control complicated expensive apparatus like accelerators and experiments 24 hours a day, 7 days a week. All effort is made to keep Tango stable and every alternate release concentrates on bug fixes rather than new features. After that emphasis is again put on new features. The next major release (V7) will concentrate on new features.

Software Quality measures how well software is designed, and how well the software conforms to that design. The design of the Tango libraries is maintained by a small group of system developers. They take input from the community and ensure that the design satisfies the needs. The developer team is open to all member institutes of Tango. Tango currently has a test system for the system libraries which tests that the features are correctly implemented. There is currently no generic test system for device servers.

⁷ <http://www.tango-controls.org/Documents/papers/FOPA01.pdf>

Packaging makes the difference between pain and pleasure in the life of a control system integrator. Tango offers source code packaging of the system libraries for Linux platforms and binary code packaging for Windows platforms.

Documentation is a strong and weak point in Tango. A lot of documentation exists for Tango - the Tango book is almost 500 pages long, but it is difficult to keep it up to date. Tango plans to adopt a new system of documentation which is easier to keep up to date and which is generated automatically as much as possible:

4.3.16 System Tools

Tango system tools consist of two applications – Jive for navigating and testing devices and Astor for managing a Tango control system. These tools are used by all institutes and are constantly improved to take into account feedback from users. One of the main improvements in the future will be the possibility to display device hierarchies and their state in real time. This is indispensable for managing large complicated systems which involve tens of thousands of devices.

Individual device servers sometimes need tuning in function of their hardware and the client load. In the future it will be possible to display live performance metrics for any device (today it is possible only for the database device).

4.3.17 Standard Interfaces

The definition of Standard Interfaces for families of devices is essential to achieve hardware decoupling and encourage client application sharing. Tango defines Standard Interfaces with the help of Abstract classes. The current list will be extended in the future to cover a complete catalogue of common hardware.

4.3.18 Conclusion

The Tango control system is a modern control system used to control accelerators and beamlines at five synchrotrons and one XFEL (currently under construction). New features are constantly being added to Tango. The Tango community keeps on finding new areas of common interest for collaboration. We see Tango as a wrapper technology for accessing and controlling hardware and any kind of software. The Tango community already shares a number of applications for doing diagnostics. The challenge for the community is to increase sharing of applications in domains like diagnostics and physics.

4.3.19 References

1. TANGO home page: www.tango-controls.org
2. CORBA Explained Simply: <http://www.ciaranmhale.com/corba-explained-simply/>
3. Lions tango home page: <http://iramis.cea.fr/scm/lions/tango/>

4.4 The LHC Controls System

E. Hatziangeli and H. Schmickler

Accelerator and Beams Controls Group, CERN

Mail to: Eugenia.Hatziangeli@cern.ch or Hermann.Schmickler@cern.ch

4.4.1 Introduction

The scale and complexity of equipment control for the Large Hadron Collider (LHC) at CERN are unprecedented in the field of particle accelerators. The LHC has the largest number of components and the widest diversity of systems of any accelerator in the world. Tolerances are very tight on settings and function generation for beam acceleration and insertion changes and collimation.

Special emphasis also has to be given to machine protection: As many as 500 objects around the 27 km ring, from passive valves to complex experimental detectors, could in principle move into the beam path in either the LHC ring or the transfer lines. The energy stored in the beams and magnets for example can melt 500 kg of copper. If an incident occurs during operation, it is critical that it is possible to determine what has happened and trace the cause. Moreover, operation should not resume if the machine is not back in a good working state.

The operation of the LHC machine will be extremely complicated for a number of reasons, including critical technical subsystems, a large parameter space, real-time feedback loops and the need for online magnetic and beam measurements. In addition, the LHC is the first superconducting accelerator built at CERN and will use four large-scale cryoplants with 1.8 K refrigeration capability.

The accelerator controls group at CERN has spent the past four years developing a new software and hardware control system architecture based on the many years of experience in controlling the particle injector chain at CERN [1]. The resulting LHC controls infrastructure is based on a classic three-tier architecture: a basic resource tier that gathers all of the controls equipment located close to the accelerators; a middle tier of servers; and a top tier that interfaces with the operators (Figure 1).

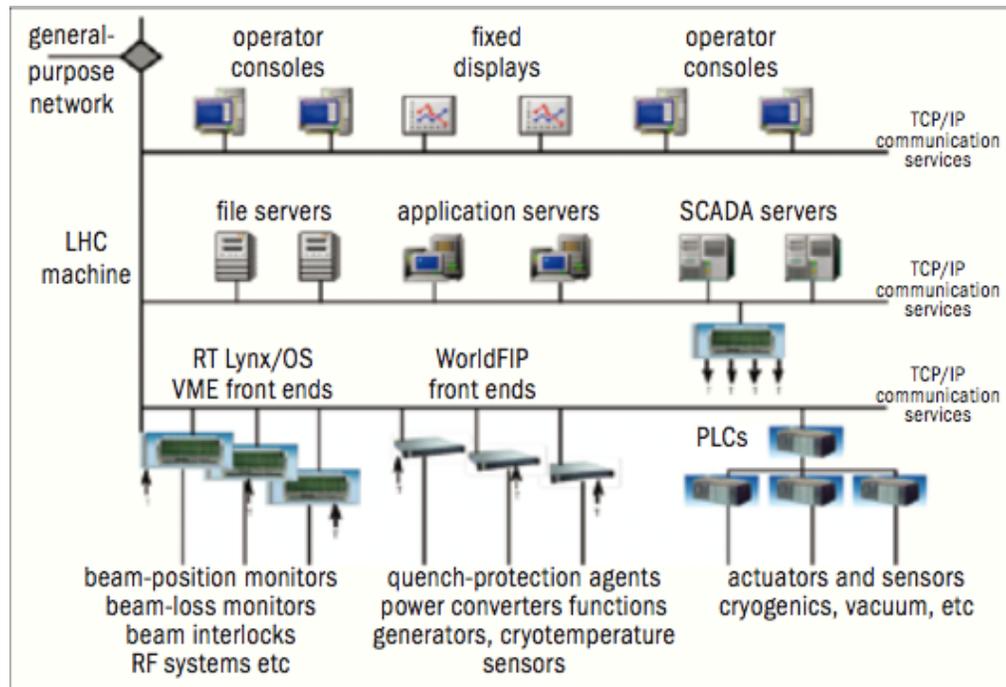


Figure 1: The complex LHC controls infrastructure network comprises three tiers.

4.4.2 Details of the Three-Tier Architecture

The LHC Software Application (LSA) system covers all of the most important aspects of accelerator controls: optics (twiss, machine layout), parameter space, settings generation and management (generation of functions based on optics, functions and scalar values for all parameters), trim (coherent modifications of settings, translation from physics to hardware parameters), operational exploitation, hardware exploitation (equipment control, measurements) and beam-based measurements. The software architecture is based on three main principles (Figure 2). It is modular (each module has high cohesion, providing a clear application program interface to its functionality), layered (with three isolated logical layers – database and hardware access layer, business layer, user applications) and distributed (when deployed in the three-tier configuration). It provides homogenous application software to operate the SPS accelerator, its transfer lines and the LHC, and it has already been used successfully in 2005 and 2006 to operate the Low Energy Ion Ring (LEIR) accelerator, the SPS and LHC transfer lines.

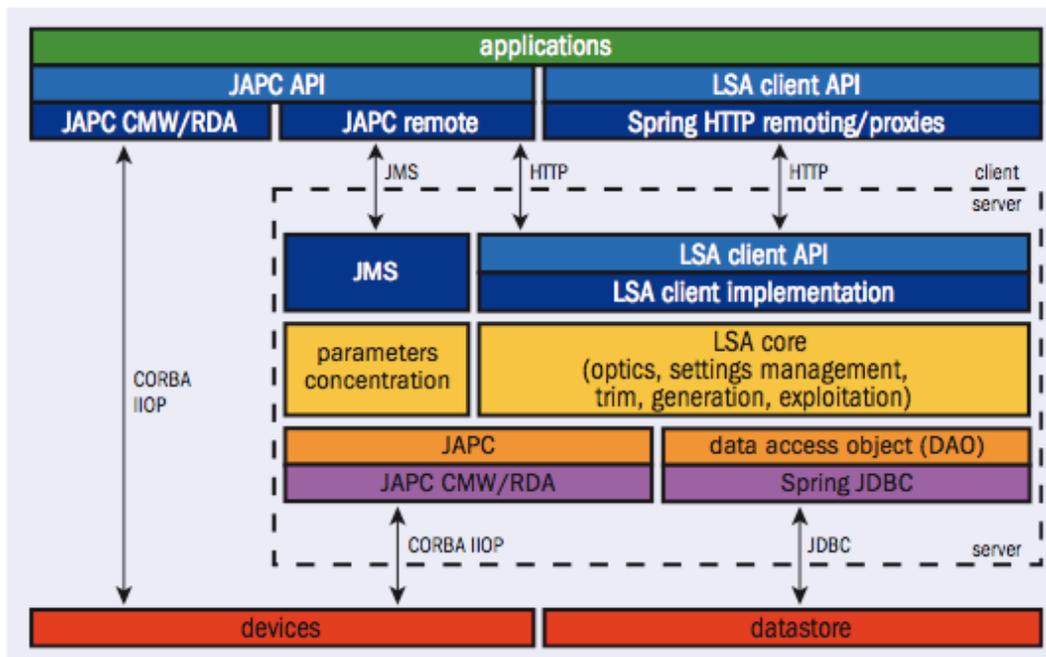


Figure 2: The architecture of the LHC software application (LSA) is designed to be modular, layered and distributed and provides operators with real-time data.

For the applications based on industrial components, the CERN developed UNICOS-PVSS framework (Figure 3) has been used both at the PLC level and at the supervision level. All of these applications have been commissioned and extensively used during the Hardware Commissioning and they have been successfully tuned in terms of bandwidth and data volumes. These industrial applications are the cryogenic process control and the cryogenics instrumentation expert tools, the machine protection systems that include the Power Interlock, Warm Interlock and Quench protection systems and the collimator environment package.

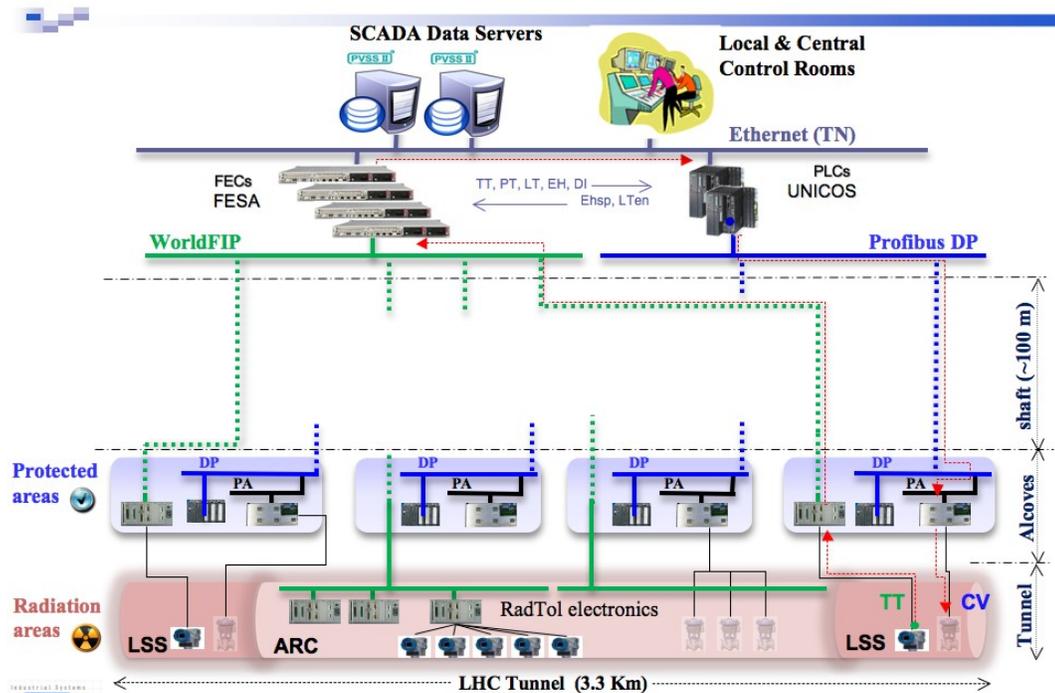


Figure 3: The UNICOS framework

The front-end hardware of the resource tier consists of 250 VMEbus64x sub-racks and 120 industrial PCs distributed in the surface buildings around the 27 km ring of the LHC. The mission of these systems is to perform direct real-time measurements and data acquisition close to the machine, and to deliver this information to the application software running in the upper levels of the control system. These embedded systems use home-made hardware and commercial off-the-shelf technology modules, and they serve as managers for various types of fieldbus such as WorldFIP, a deterministic bus used for the real-time control of the LHC power converters and the quench-protection system. All front ends in the LHC have a built-in timing receiver that guarantees synchronization to within 1 μ s. This is required for time tagging of post-mortem data. The tier also covers programmable logic controllers, which drive various kinds of industrial actuator and sensor for systems, such as the LHC cryogenics systems and the LHC vacuum system.

The middle tier of the LHC controls system is mostly located in the Central Computer Room, close to the CERN Control Centre (CCC) [2]. This tier consists of various servers: application servers, which host the software required to operate the LHC beams and run the supervisory control and data acquisition (SCADA) systems; data servers that contain the LHC layout and the controls configuration, as well as all of the machine settings needed to operate the machine or to diagnose machine behaviors; and file servers containing the operational applications. More than 100 servers provide all of these services. The middle tier also includes the central timing that provides the information for cycling the whole complex of machines involved in the production of the LHC beam, from the Linacs onwards.

A CERN developed communication protocol (CMW = CERN middleware) based on Corba assures the communication between most processes. For more short lived point-to-point messaging a JMS based middleware is used.

At the top level – the presentation tier – consoles in the CCC run GUIs that will allow machine operators to control and optimize the LHC beams and supervise the state of key systems. Dedicated displays provide real-time summaries of key machine parameters. The CCC is divided into four “islands”, each devoted to a specific task: CERN’s PS complex; the SPS; technical services; and the LHC. Each island is made of five operational consoles and a typical LHC console is composed of five computers (Figure 4). These are PCs running interactive applications, fixed displays and video displays, and they include a dedicated PC connected only to the public network. This can be used for general office activities such as e-mail and web browsing, leaving the LHC control system isolated from exterior networks.

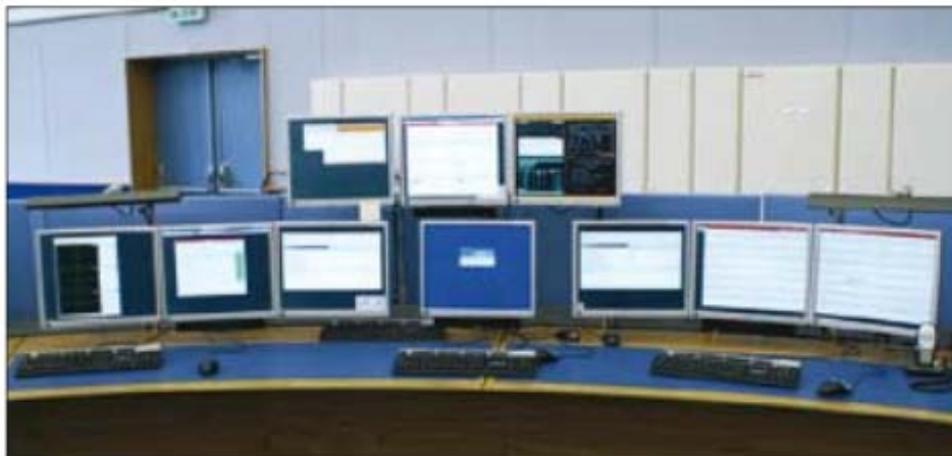


Figure 4: The CERN control centre during the first days of use in 2006 (top) and a close-up of a typical LHC operation console

used operationally for the transfer lines, the SPS, the CERN Neutrinos to Gran Sasso (CNGS) project, the experiments and the LHC, and it has recently been adapted for the PS Complex [4]. LASER provides the collection, analysis, distribution, definition and archiving of information about abnormal situations – fault states – either for dedicated alarm consoles, running mainly in the control rooms, or for specialized applications.

LASER does not actually detect the fault states. This is done by user surveillance programs, which run either on distributed front-end computers or on central servers. The service processes about 180 000 alarm events each day and currently has more than 120 000 definitions. It is relatively simple for equipment specialists to define and send alarms, so one challenge has been to keep the number of events and definitions to a practical limit for human operations, according to recommended best practice.

The controls infrastructure of the LHC and its whole injector chain spans large distances and is based on a diversity of equipment, all of which needs to be constantly monitored. When a problem is detected, the CCC is notified and an appropriate repair has to be proposed. The purpose of the diagnostics and monitoring (DIAMON – Figure 6) project is to provide the operators and equipment groups with tools to monitor the accelerator and beam controls infrastructure with easy-to-use first-line diagnostics, as well as to solve problems or help to decide on responsibilities for the first line of intervention.

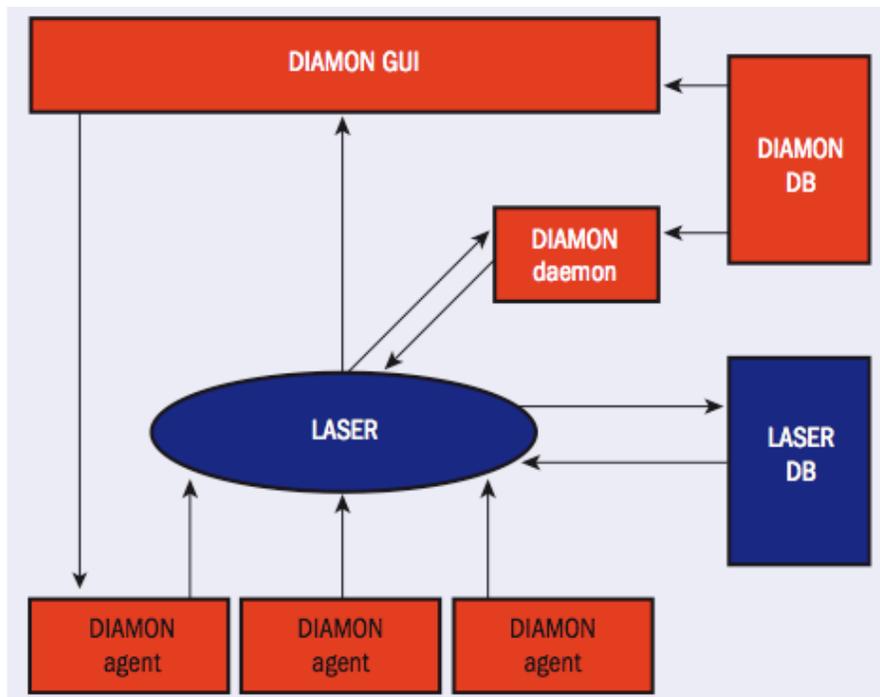


Figure 6: Architecture for the diagnostics and monitoring system, DIAMON.

The scope of DIAMON covers some 3000 “agents”. These are pieces of code, each of which monitors a part of the infrastructure, from the fieldbuses and frontends to the hardware of the control-room consoles. It uses LASER and works in two main parts: the monitoring part constantly checks all items of the controls infrastructure and reports on problems; while the diagnostic part displays the overall status of the controls infrastructure and proposes support for repairs.

The frontend of the controls system has its own dedicated real-time frontend software architecture (FESA). This framework offers a complete environment for equipment specialists to design, develop, deploy and test equipment software. Despite the diversity of devices – such as beam-loss monitors, power converters, kickers, cryogenic systems and pick-ups – FESA has successfully standardized a high-level language and an object-oriented framework for describing and developing portable equipment software, at least across CERN's accelerators. This reduces the time spent developing and maintaining equipment software and brings consistency across the equipment software deployed across all accelerators at CERN.

A Software Interlock System (SIS) is put in place to further protect the LHC machine. The preliminary setup consists of continuous surveillance of 1700 monitored devices. Among others, all power converters, the current of separation dipoles, the MCBX orbit correctors, the screens for ring, injection and extraction and the limits of the circulating beam intensity are surveyed. All devices are grouped in different permit structures providing alarms and beam inhibits when necessary.

To allow for an automatic and failsafe operation of the LHC, the development of a generic sequencer was driven to offer the required functionality for the hardware, the beam commissioning and LHC beam operations.

This article illustrates only some of the technical solutions that have been studied, developed and deployed in the controls infrastructure in the effort to cope with the stringent and demanding challenges of the LHC. This infrastructure has now been tested almost completely on machines and facilities that are already operational, from LEIR to the SPS and CNGS, and LHC hardware commissioning. The estimated collective effort amounts to some 300 person-years and a cost of SFr21 m. Part of the enormous human resource comes from international collaborations, the valuable contributions of which are hugely appreciated. Now the accelerator controls group is confident that they can meet the challenges of the LHC.

4.4.4 Outcome from the Injection Tests

A large number of components of the LHC controls infrastructure have been intensively exercised during several dry runs in 2008 and have been commissioned to near their full functionality. Some of the systems involved include: fixed displays, the sequencer, the LASER alarm system, cryogenics control, the quench protection system, and the underlying infrastructure.

One of the objectives of the prior injection tests was to further commission controls components, which had only been tested locally, but which never had been used in full deployment. During these tests, emphasis was given on the commissioning of two major controls components, the timing system software and the logging system.

The timing system initially demonstrated problems with RF synchronization sending the information about the next injected bucket and ring too late in the SPS cycle. The source of the problem was identified by the system expert and appropriate solutions were put in place to allow for the continuation of the tests.

For the logging system some problems appeared at the moment of high data rates and large data volumes especially for the clients like the BLM, BPM and Power Converters. A new API was developed which offered massive parallelization guarding, in addition, against lost of database connection by persisting the data in local store. This

new API successfully allowed for full LHC logging of all required equipment parameters without any data loss.

The SIS and Beam Interlock System, being part of the machine protection systems, protected the LHC machine by surveying and analyzing the state of various key parameters and devices and dumping or inhibiting the beam when a potentially dangerous situation occurred.

During the injection tests we took the opportunity to commission for the first time the RBAC system. The commissioning was prepared via several dry runs and RBAC was put in effect from the second injection test onwards protecting successfully the access to the BI, PO and BT equipment and critical settings with the appropriate rules and roles defined in the access database. Support was also given to the DIAMON system, which was fully deployed during the tests to monitor the LHC controls infrastructure.

The LHC Controls evolved progressively during the injection tests to a stable and well functioning infrastructure, which supported well the successful LHC start-up. All the prior tests were extremely useful to identify areas of weakness and to prepare the LHC controls infrastructure for beam commissioning. The controls commissioning work will continue well after the first months of LHC beam operation as new applications will be deployed, which have been developed specifically for the next phases of LHC as it moved towards first collisions.

4.4.5 Collaborations and Outlook

The LHC controls infrastructure has mainly been developed at CERN in the AB-CO controls group, which during the development period was strengthened with up to 60 temporary resources. Significant components were also contributed in collaboration with outside institutes, notably with FNAL and GSI.

At CERN a renovation program of controls for the whole injector chain has been launched with the aim of unifying the controls infrastructure throughout CERN to the components developed for the LHC. The estimated time of this renovation is 2009 – 2011.

Also a discussion has been launched with outside institutes (GSI, FNAL, ITER) with the aim of generalizing the CERN developed components, such that part of the LHC controls or the complete system could be reused in other laboratories.

4.4.6 Further Reading

1. P. Charrue – “The Status Of The LHC Controls System Shortly Before Injection Of Beam” - ICALEPCS’07
<http://cern.ch/AccelConf/ica07/PAPERS/MOAB01.PDF>
2. CERN Courier May 2006 - <http://cerncourier.com/cws/article/cern/29603>
3. S. Lüders - CERN Computing Newsletter vol 42 issue 5 page 11 –
<http://cerncourier.com/objects/cnl/4/11/1/nov-decpdf.pdf>
4. K. Sigerud, N. Stapley and M. Misiowiec : “First Operational Experience with LASER” - ICALEPCS’05.
5. Papers related to the LHC controls infrastructure, presented at the ICALEPCS ’07 conference, see <http://cern.ch/AccelConf/ica07>
6. The LHC Injection Tests - LHC Performance Note 1 - 2008-10-21

4.5 The ALICE Control System

André Augustinus, Marco Boccioni, Peter Chochula, Giacinto de Cataldo, Lennart Jirdèn, Svetozar Kapusta, Peter Rosinsky, Pavel Stejskal, Cesar Torcato de Matos, and

Lionel Wallet

CERN, Geneva, Switzerland

Mail to: Lennart.Jirden@cern.ch

4.5.1 Introduction

4.5.1.1 LHC and Its Experiments

The 27km Large Hadron Collider (LHC) at CERN in Geneva Switzerland saw its first circulating beams in September 2008 and will provide first collisions in spring 2009. Currently six experiments are installed at the LHC. Each experiment is distinct, characterised by its unique particle detector techniques and physics goals.

Two experiments, ATLAS and CMS, are built as general-purpose detectors designed to investigate the largest range of physics possible and their first goal to determine the existence of the Higgs boson. The ALICE and LHCb experiments have specialised detectors for analysing the LHC collisions in relation to specific phenomena: Heavy Ion physics for ALICE and B-physics for LHCb. Two more experiments, TOTEM and LHCf, are much smaller in size and are designed to focus on ‘forward particles’ (protons or heavy ions). All six experiments are installed in underground caverns located on the LHC.

4.5.1.2 Control Systems for HEP Experiments

History

With the growing size and complexity of high energy physics (HEP) experiments the need for efficient and high-performing control systems has become increasingly important. Up until the early 1980’s, while the control systems for the accelerators already were quite sophisticated, the controls in experiments was limited to simply controlling individual devices. The controls functionality was not really integrated in the online operational environment. Any monitoring parameter was read by the data acquisition system and entered in the data stream.

It was only with the LEP experiments (ALEPH, DELPHI, L3, OPAL; operational from 1986 to 2000) that more intelligent and integrated control systems were implemented. The LEP control systems had their own dedicated and distributed intelligence. For the DELPHI experiment [1] this was implemented as a simple two layer architecture: ‘front-end’ controllers were responsible for the communication with the hardware, and ‘back-end’ computers for the user interfaces. Some basic automation was implemented on the front-end controllers and they also hid the details of the device access and provided a ‘generic’ interface to the back-end control layer. The front end controllers acquired slowly varying parameters (temperatures, pressures etc.) that were read by the back-end. The back-end layer stored all relevant controls data (conditions data) as a separate data stream in a database, that was consulted offline during event reconstruction. During the lifetime of the experiment more and more routine operations were programmed as automatic actions in the back-end and a thin control layer on top

of the online systems (DAQ, Trigger, LEP machine interface) allowed in the last years for an efficient and fully automated operation of the experiment.

The LHC Control Systems

The control systems for the LHC experiments have been built on the experience gained with the LEP experiments. However, while for the four LEP experiments the controls systems were built completely independent of each other, the LHC experiments decided to collaborate and share the developments. A Joint COntrols Project (JCOP, [2]) was therefore created, as collaboration between CERN and the LHC experiments, to provide a discussion forum and to develop a set of common tools and components.

This has allowed the LHC experiments to use the same basic tools to implement their control systems and to adopt more or less the same architecture. Still, due to the inherent differences between the experiments, each of the control systems has their own particularities. The ATLAS and CMS experiments have only a limited number of large sub-detectors; each of these has many control channels. These experiments have control systems with a large number of similar channels and the systems are only loosely coupled with the other online systems, trigger (TRG) and data acquisition (DAQ).

The LHCb experiment has several smaller sub-detectors, and has adopted a highly integrated approach for the control of all online systems. Control of DAQ and TRG and control of the experiment's infrastructure are all an integral part of an experiment wide homogeneous control system.

The ALICE experiment is made up of many relatively small to very small sub-detectors developed and built by a large number of small institutes around the world. A strong central coordination has therefore been required to come to a coherent and homogeneous control system. Much effort has also been made to come to a relatively tight integration of the various, heterogeneous, online systems. An additional challenge is the 'cultural' background of the collaborating institutes in ALICE. Being a heavy ion experiment, most of the institutes participating in ALICE so far only worked in relatively small collaborations, and had very little experience with the operational requirements of projects on the scale of LHC. The majority of the groups in the other LHC experiments had already collaborated in LEP experiments or other large HEP experiments.

From the above it is clear that the control systems of the LHC experiments have much in common. So, although the following sections of this paper will describe the ALICE detector control system, many of the architectural and technical choices also apply for the other experiments.

4.5.1.3 The ALICE Experiment

ALICE (A Large Ion Collider Experiment) is a general-purpose, heavy-ion detector designed to study the physics of strongly interacting matter and the quark-gluon plasma in nucleus-nucleus collisions at the CERN LHC. It currently involves more than 1000 physicists and senior engineers, from both the nuclear and high-energy physics sectors, from well over 100 institutions in about 30 countries. It will allow a comprehensive study of hadrons, electrons, muons, and photons produced in the collision of heavy nuclei (Pb-Pb), up to the highest multiplicities anticipated at the LHC. The physics programme also includes collisions with lighter ions and at lower energy, in order to vary energy density and interaction volume, as well as dedicated proton-nucleus runs. Data taking during proton-proton runs at the top LHC energy will provide reference

data for the heavy-ion programme and address a number of specific strong-interaction topics for which ALICE is complementary to the other LHC experiments.

The overall dimensions of the detector are 16x16x26m³ with a total weight of approximately 10 000 tons. ALICE consists of a central barrel part, which measures hadrons, electrons, and photons, and a forward muon spectrometer. The central part is embedded in a large solenoid magnet reused from the L3 experiment at LEP. The forward muon arm consists of a complex arrangement of absorbers, a large dipole magnet, and fourteen planes of tracking and triggering chambers (Figure 1).

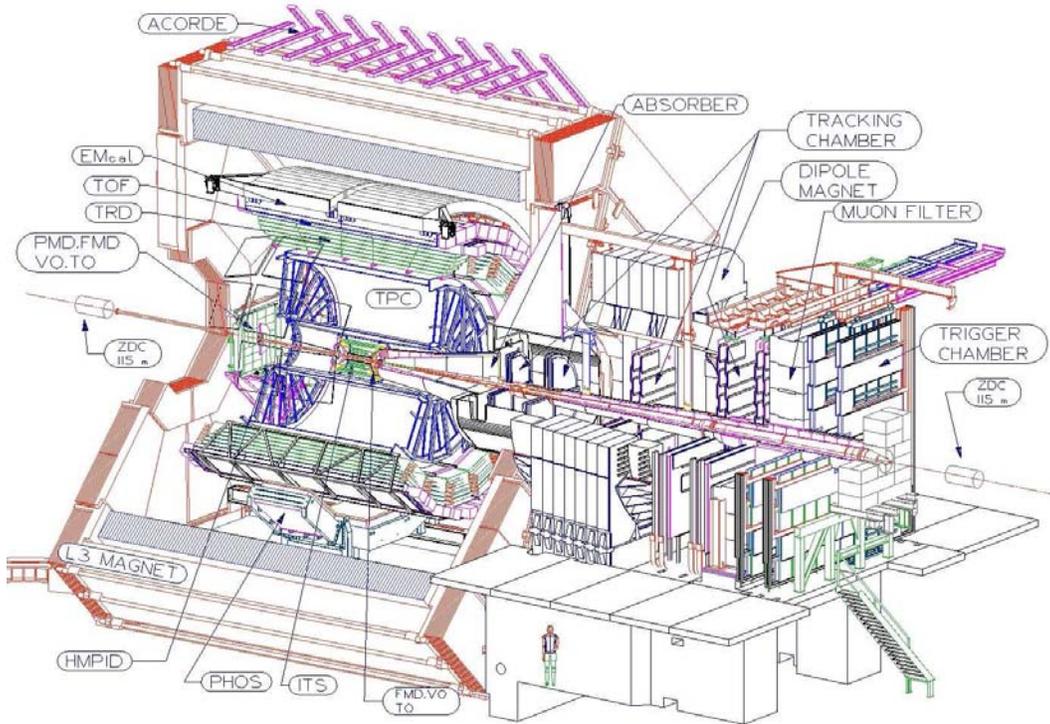


Figure 1: The ALICE detector

The operation of the experiment relies on several independent online systems. Each responsible for a different domain of activities: The Detector Control System (DCS) for the control and safety of the experiment; this will be described in detail in this article. The Data Acquisition (DAQ) system is responsible for the readout of the physics data for event building and for data transport. The Trigger (TRG) system selects the interesting events and triggers the readout of the sub-detectors. The High-Level Trigger (HLT) system performs online reconstruction of data in order to reject or tag events and to allow for data compression.

4.5.2 The ALICE Detector Control System (DCS)

4.5.2.1 Introduction

The primary task of the ALICE Control System is to ensure safe and correct operation of the experiment. It provides remote control and monitoring of all experimental equipment to allow the entire experiment to be operated from the ALICE

Control Room (ACR) at LHC point 2, through a unique set of operator panels. The DCS provides the optimal operational conditions so that the physics data taken by the experiment is of the highest quality.

The control system has been designed to reduce the downtime of the experiment to a minimum and hence contribute to a high running efficiency. It also maximises the number of readout channels operational at any time, and measures and stores all parameters necessary for efficient analysis of the physics data.

4.5.2.2 *Design Strategy and System Architecture*

The ALICE control system has been coordinated and built by a small central team at CERN, the ALICE Control Coordination (ACC) team, in close collaboration with all the sub-detector groups in the institutes around the world. The construction of certain parts of the system has been sub-contracted to various services groups at CERN and in order to assure a tight ALICE online integration close collaboration between the DCS, DAQ and Trigger groups has been important. Thus, in total, more than 100 people have been involved in the development and construction of the control system.

In order to come to a rational and efficient design of the system, past experience and contacts with the other LHC experiments have been valuable inputs. To limit the dispersion of solutions and to reduce the development effort the obvious overall strategy has been to use common tools, common components and common solutions wherever possible.

Requirements

Although covering a wide variety of components, and being developed by various groups in parallel, the DCS as a whole has to stay coherent and homogeneous and allow for easy integration of components. To accommodate the changes of the experiment during its lifetime, the DCS needs to be flexible and scalable. As the control system must stay operational throughout all phases of the experiment (data taking, shutdown etc.), it is designed to cope with different operational modes. It allows for independent and concurrent operation of each sub-detector, or any part of it.

Since the shift crew is not necessarily expert in controls or in the operation of a particular sub-detector, special attention has been given to the presentation of the system to the operator. Remote access to the DCS is needed and a strict access control mechanism has been put in place, based on the origin of the access and the user profile.

Ensuring the integrity of the detector equipment is another important task of the control system. The control system must be reliable and allow for both hardwired and software actions in case of hazardous situations.

Methods

The ALICE experiment is composed of as many as 18 sub-detectors, each with up to 15 different sub-systems, such as high voltage, front-end electronics, cooling and gas systems, that need to be controlled. Given the large number of controls sub-systems that had to be developed it was clear from start that common solutions for common requirements had to be used across the experiment in order to limit the development effort and to obtain a homogeneous and integrated control system. Therefore the ACC team, which was set up to coordinate all controls activities within the experiment, started by systematically collecting the controls requirements of each sub-system and light-weight User Requirements Documents (URDs) were established in order to enable

the identification of communalities. Each URD was complemented by an ‘overview drawing’ that depicts the hardware structure of the control system of each sub-detector.

Since the developers of the controls sub-systems of the detectors are spread out in institutes around the world it was also necessary to establish a strong central coordination. This has been achieved by means of regular ALICE DCS workshops and technical coordination meetings where technical issues and progress is being discussed.

The collaboration between the four LHC experiments has also been very important. A set of components and tools for the implementation of the control systems has been developed in common within the frame of the Joint Control Project (JCOP). JCOP has also constituted an excellent platform for exchange of information and ideas between the LHC experiments.

Architecture

The ALICE control system is responsible for configuring, monitoring and controlling the equipment of the experiment. These range from commercial hardware devices such as power supplies and crates, to sub-detector specific equipment like front-end chips, etc. It also covers computing devices such as PCs and PLCs as well as the software processes running on them. These tasks are accomplished mainly by sending commands and settings to and reading information back from the equipment.

The control system is able to take pre-programmed decisions and automatic actions without operator intervention such as recovering from errors. The operator is able to interact with the control system through a set user interfaces that presents the information and allows issuing of commands.

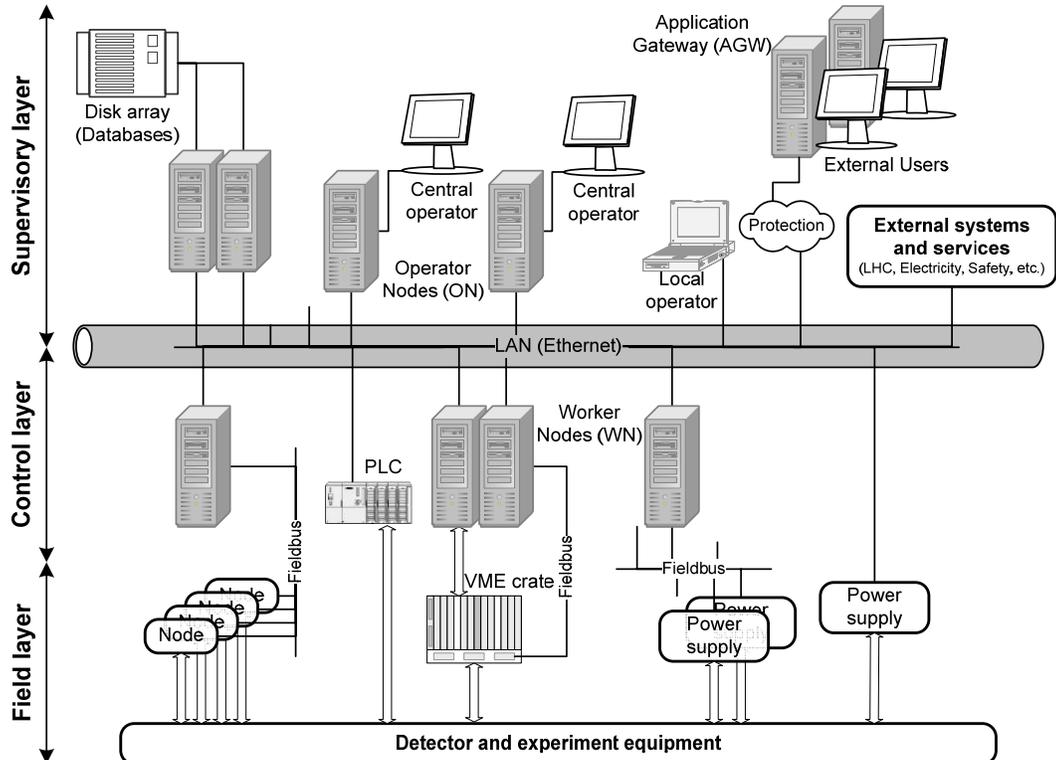


Figure 2: DCS hardware architecture

The hardware architecture can be sub-divided in three layers; a supervisory, a control and a field layer as shown in Figure 2. The supervision layer consists of some 20 Operator Node computers (ON) that provide the user interfaces to the operators. This layer interfaces to the control layer where about 100 Worker Node computers (WN) and a number of PLC and PLC like devices interface to the experiment equipment. These devices collect and process information from the lower, so called field layer, and make it available to the supervisory layer. In a similar way the control layer receive information from the supervisory layer to be processed and distributed to the field layer. The field layer comprises various types of field devices such as power supplies, field bus nodes, sensors, actuators, etc. All computers, PLC's and many of the field layer devices are connected to a dedicated, highly protected [3] and partly redundant DCS Local Area Network (LAN) which runs throughout all the experiment. In total about 1400 devices are connected to the DCS LAN.

The software architecture (Figure 3) is a tree-like structure that represents the structure of sub-detectors, their sub-systems and devices. The structure is composed of nodes which have parents and children. Each node has one single 'parent' except for the 'root node' at the top of the tree which has no 'parent'. Nodes may have zero, one or more children. A node without children is called a 'leaf' and a subset of a tree's nodes is called a 'sub-tree'.

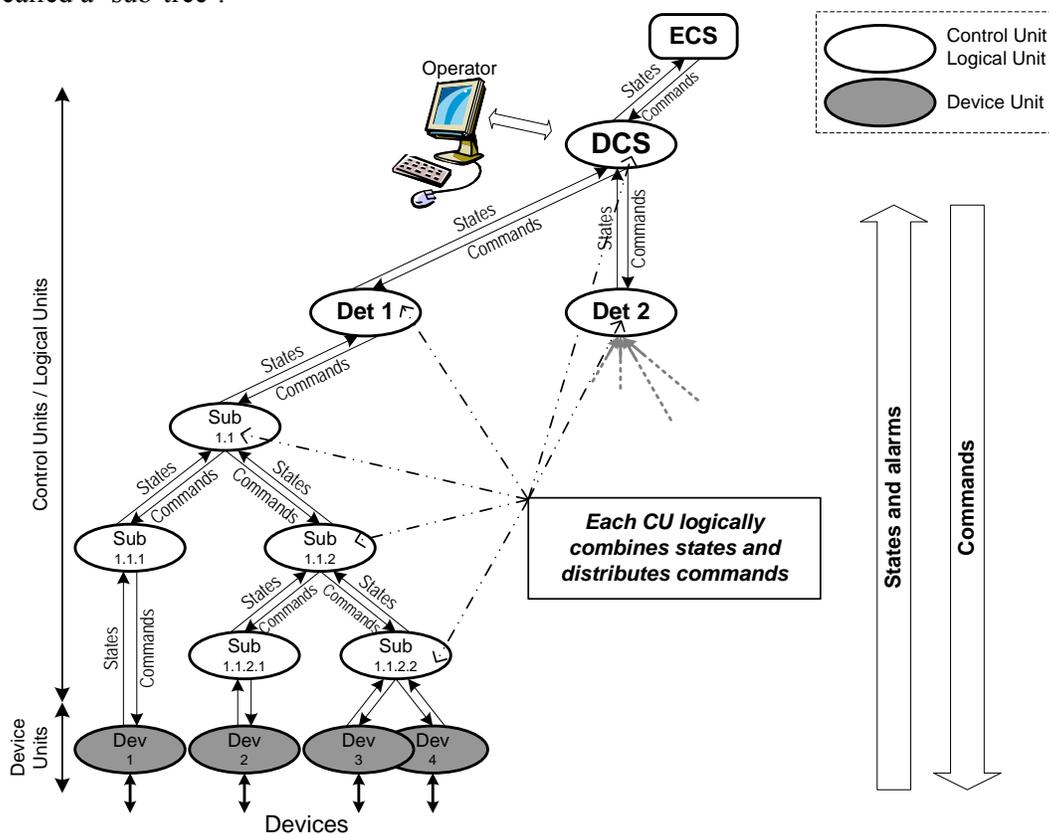


Figure 3: DCS software architecture

Control, Logical and Device Units

There are three types of nodes; a Control Unit (CU), a Logical Unit (LU) and a Device Unit (DU), that serve as basic building blocks for the entire hierarchical control

system. The CU and LU model and control the sub-tree below it and the device unit 'drives' the device. The hierarchy can have an arbitrary number of levels to provide the sub-detectors with as many abstraction layers as required.

Partitioning

The hierarchy also offers a high degree of independence between its components and allows, by means of the concept of 'partitioning', for concurrent use. Partitioning is the capability of independently and concurrently controlling and monitoring parts of the system, typically sub-trees of the hierarchical control tree. This functionality is essential during installation, maintenance and commissioning phases, when only parts of the control system might be available and sub-detectors need to control the available equipment. The partitioning feature is also essential during normal operation for debugging purposes or for sub-detector test or calibration runs. During longer shutdown periods, sub-detectors might need to run their sub-detector control system while other parts of the control system are still switched off or is being controlled by another operator. The same principle is also applicable at the level of the experiment: entire sub-detectors can be 'taken' concurrently by a sub-detector operator or by a central operator. In the control tree only the control units can become root node of a partitioned control tree.

Finite-State Machine

The behaviour and functionality of each unit in the control tree is implemented as a finite-state machine (FSM). The finite-state machine concept is a fundamental component in the control system architecture. It is an intuitive, generic mechanism used to model the functionality of a piece of equipment or a sub-system. The object to be modelled is thought of as having a set of stable 'states' between which it can transit and execute 'actions'. The transitions are triggered either by commands from an operator or by other objects above ('parents') or by state changes of objects below ('children'). Two types of objects can be defined in the FSM concept: abstract objects, represented by a control or logical unit, and physical objects, represented by a device unit in the control tree. This concept allows for distributed and decentralised decision making and actions can be performed autonomously, even when controlled centrally. This will naturally lead to parallelism in automated operations such as error recovery, thus increasing the efficiency of the system.

The FSM concept allows building an 'operations plane' on top of the 'systems plane' that is defined by the computer systems and their connection to the devices. The operator sends commands to the FSM hierarchy on the 'operations plane' and the actions at the level of the devices are executed by the software systems on the 'systems plane' (Figure 4).

This concept has evident advantages; both planes can be developed independently and in parallel and modifications on each plane will have only limited implications on the other. The systems plane will stay relatively stable in time (e.g. it will change when devices are added), whereas the operations plane might be modified more frequently depending on the operational needs of the experiment, yet without implications for the systems plane.

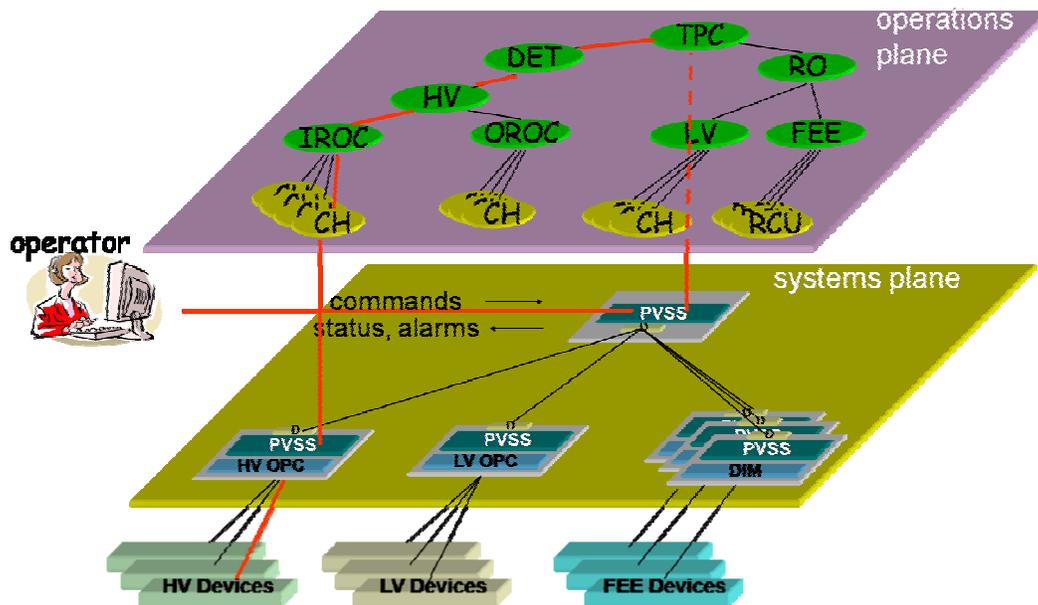


Figure 4: Systems plane (hardware/software) and Operations plane (hierarchical structure modelled by Finite State Machine).

4.5.2.3 System Implementation - Components

PVSSII

The core software of the control system is a commercial SCADA (Supervisory Controls And Data Acquisition) system: PVSSII. This package was selected already in 1999 after an extensive evaluation process performed by JCOP. PVSSII is now used by all four LHC experiments, as well as by several service groups at CERN providing systems such as gas, safety, etc. It offers many of the basic functionalities generally needed by a control system.

All PVSSII applications (projects) are running on top of the Windows operating system on the Worker Nodes (Windows XP) and on the Operator Nodes (Windows Server 2003). A number of specific worker nodes (those interfacing to the Front End Electronics) run the Linux (SLC4) operating system. The whole ALICE DCS runs on a cluster of around 130 rack mounted commercial PCs (Figure 5), and on around 800 single board computers.



Figure 5: Part of the DCS cluster

JCOP and ALICE frameworks

Around PVSSII a framework was built as a joint effort between the four LHC experiments. This framework complements the PVSSII system and provides tools and components for the implementation of the common functionalities that are expected from the control system, such as FSM, database access, access control, basic user interfaces, configuration, etc. (Figure 6). The JCOP-framework also includes interfaces to several hardware devices that are commonly used in the LHC experiments and such interface components hide much of the PVSSII internals from a non-expert end-user. In the same context an ALICE framework was developed to cater for ALICE specific needs.

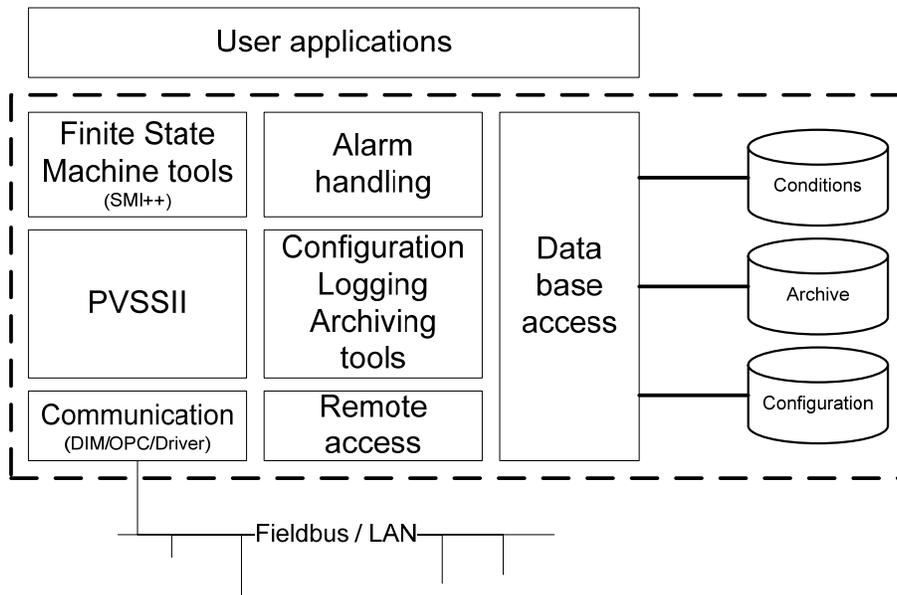


Figure 6: Overview of control system components

These tools are used by the sub-detector expert to build their applications. The tools are accompanied by a set of rules and guidelines (naming conventions, user interface look and feel, etc.) to ensure the homogeneity and to simplify the integration of independently developed components.

Communication and Hardware Access

Communication with the hardware is restricted to a limited set of communication protocols. The two protocols OPC (OLE for Process Control) and DIM (Distributed Information Management) cover all the needs in ALICE. OPC is a widely accepted industry standard used for communication with commercial devices. It implements a client-server mechanism, where PVSSII provides a generic OPC client and the manufacturer of the device usually provides the OPC server. DIM is used to communicate with custom built equipment, it is available for many platforms and libraries are available for several computer languages. DIM is a CERN developed protocol and implements a client-server mechanism over TCP/IP. When a DIM-server is developed for a given piece of equipment, PVSSII can be used to access it through the DIM-client integrated in the framework.

DIM is also the underlying communication layer for the Data Interchange Protocol (DIP) that is used to exchange data between the control system and LHC as well as with the various services such as gas, electricity, cooling etc.

In addition PVSSII can also directly communicate with equipment through drivers. PVSSII drivers exist for Modbus devices and some other (mainly industrial) devices.

Finite-State Machine

The hierarchical tree-like control structure, as described earlier, relies heavily on finite-state machines. As PVSSII itself does not provide any FSM functionality, this has been added in the framework (SMI++, [4], Figure 7). A very close integration between PVSSII and the FSM tools has been achieved.

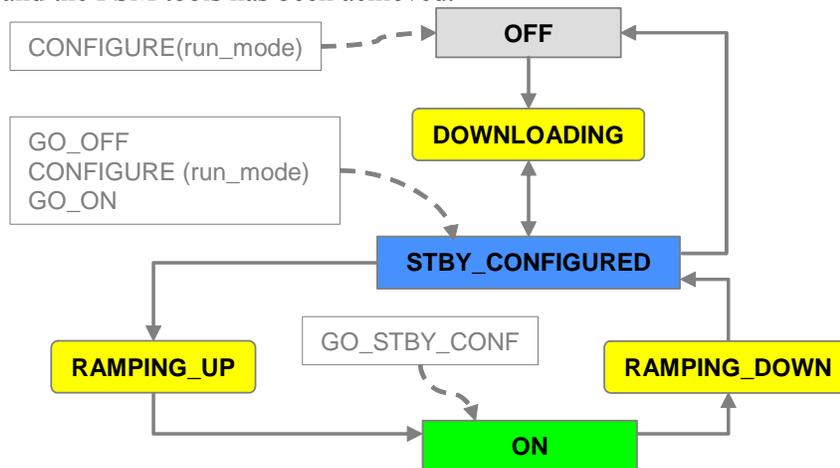


Figure 7: Example of a simple Finite State Machine Diagram

Databases

The ALICE control system has to cope with much larger amounts of data than in previous generation systems [5]. At the start of a physics run, when the experiment needs to be set-up and configured to the actual run conditions, up to 6 GB of data might

be loaded from configuration databases to the hardware devices. This includes the configuration of the control system itself, configuration of hardware devices such as power supplies or VLSI chips in the front-end electronics, configuration of processes, etc. The chips and processors in the front-end electronics require particularly large amounts of configuration data and the performance of the database access and the FEE loading mechanism has therefore been much optimised.

Data archiving is an integral part of PVSSII and allows the user to store the history of any data available in the system that he might need. The steady archival rate for ALICE is estimated to 1000 inserts/s throughout the year. The database service, consisting of an Oracle RAC server (with 30 TB of fully redundant storage) located at the experimental site, is designed to cope with a steady rate of 150 000 inserts/s, which corresponds to a peak load during the ramp-up periods, when most of the channels change.

The PVSSII archive is also an essential tool for system debugging as it contains the history of any data in the system. During a physics run a subset of the archived data is also transferred to the Offline Conditions DataBase (OCDB). The OCDB mirrors data essential for offline analysis and provides an easy access and retrieval mechanism. The interface between the OCDB and the DCS archive (AMANDA) was implemented by the central team. Some of this data is also used in calibration procedures and to prepare new configurations for the detectors.

4.5.2.4 System Implementation - Applications

The ALICE experiment consists of a fairly large number (approx. 130) of sub-systems and controls slices that need to be developed. The aim was to standardise and use common solutions as far as possible. At the device level the sub-detector users were encouraged to use similar types of devices whenever possible, and common specifications were developed for devices to be purchased, such as high and low voltage power supplies, VME crates and VME single board computers. Manufacturers were in this way asked to provide standard control interfaces based on OPC and CERN standard fieldbuses. For the Front-End Electronics (FEE), which is custom made for each sub-detector, a standard software interface for its control and configuration was defined.

Nearly all requirements for High and Low Voltage could be satisfied by three manufacturers (CAEN and Iseg for HV; CAEN and Wiener for LV). This already significantly reduced the development effort.

General purpose monitoring comprises any parameter that is not acquired through any of the other sub-systems. These parameters are typically temperatures measured on the detector. To avoid the dispersion of solutions for this task, the ELMB microcontroller [6] was adopted as a general purpose monitor device. It is a general purpose microcontroller with 64 analogue inputs and 16 digital I/Os connected via CANbus. An OPC server is available to control the ELMB and access the acquired data.

Common solutions however not only consist of standard equipment and standard control interfaces, but wherever possible also implement standard logical behaviour of the equipment. A control unit for a class of devices implements a unique FSM state diagram describing the behaviour and commands for that class of devices. This naturally leads to uniformity in names of states and commands. Also a ‘non-expert’ operator is presented with coherent behaviour for similar sub-systems throughout the experiment. The behaviour includes standard operation features such as automatic recovery of anomalies.

Front-End Electronics

The control of Front-End Electronics (FEE) is a complex and delicate task. It involves control of voltage regulators, power switches, error registers, etc., and monitoring of temperatures, voltages, currents, status registers, etc. of the FEE boards. It also involves configuration and initialization of FEE controllers and of all the various custom chips on the detector boards. It requires a very close interaction between the DCS and DAQ systems, as in many of the sub-detectors both systems share the access path to the FEE or use the FEE concurrently. A further challenge is the fact that the architecture and implementation of the FEE is different for each sub-detector and based on custom chips. Many different techniques are deployed to communicate with these chips such as JTAG, Ethernet, GOL, Profibus, CANbus, etc., each requiring a different access strategy. Also the Detector Data Link (DDL) used by DAQ provides functionality for downloading configuration information. In addition large amounts of data are involved (unusually large compared to other areas of the control system); especially for configuration of the FEE (Figure 8).

In order to achieve maximum commonality between all different FEE architectures a Front-End Device (FED, [7]) was defined. The FED represents a hardware abstraction layer allowing DCS transparent access to the FEE. It responds to standard commands and performs requested tasks such as loading configuration registers, resetting the chips, etc. If the FEE provides data which needs to be monitored, this is gathered by the FED and made available to the supervising software.

The FED is built as a package of software and hardware with a standardized software interface. A common FED client-server model was adopted for the FED which hides the implementation details to higher software layers. The server communicates with the hardware and publishes data as services. A client can subscribe to services and send commands to the server. Several clients can subscribe to the same server in parallel allowing for distributed monitoring. The DIM protocol was chosen as the underlying communication layer.

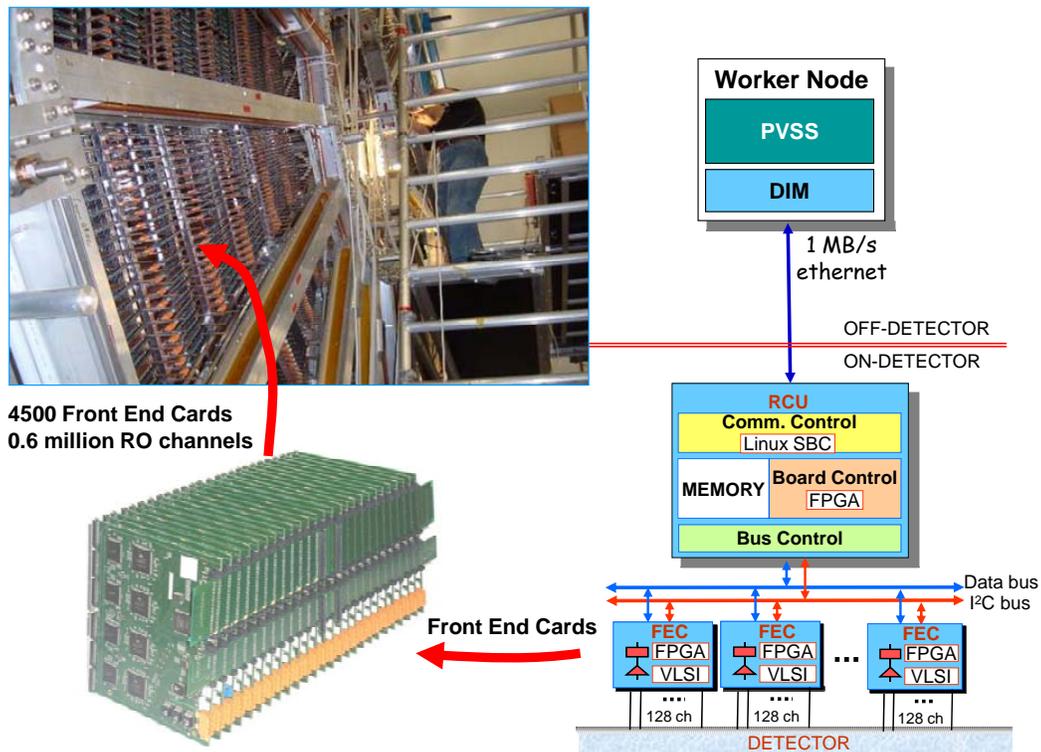


Figure 8: Front End Electronics control for TPC sub-detector

Services

The DCS interfaces to various services in order to keep the control system up to date with the experiment operation environment. Some of the services allow active control from DCS, however the majority of interactions are monitoring only. For communication with the services the Data Interchange Protocol (DIP) was defined, which allows a transparent way to exchange information between the systems involved.

Eight sub-detectors in ALICE have each a dedicated gas system, provided with an associated control system. Any information needed by the sub-detectors from their gas system is made available through DIP. A central application subscribes to the DIP parameters and makes them available in the distributed PVSSII system, for use by the sub-detectors.

Twelve of the ALICE sub-detectors are equipped with individual and in many cases quite sophisticated cooling systems. The corresponding control systems are based on PLC technology and driven from PVSSII using the Modbus TCP/IP protocol.

Both gas and cooling systems generate, as backup to the software interface, hardware interlocks allowing sub-detectors to take actions and protect their equipment in case of serious anomalies.

There are about 200 electronics racks in ALICE and the majority of them are located in the experimental cavern which is not accessible during the operation of the experiment. The control of the power to the individual racks is a PLC based control system that reads status information (on/off, error, etc.) and provides on/off control for each outlet. The DCS interfaces with these PLCs via the Modbus TCP/IP protocol.

Each rack is also equipped with a monitoring system that allows monitoring of the environment inside the rack and the operational state of the rack. This monitoring

system is based on the ELMB. The protection of each rack is ensured by a thermo-switch and a smoke detector; in case of anomaly the power to the rack will be cut.

DCS also exchanges information with the LHC machine, the magnet control system, the safety systems and the primary services, and the DIP protocol is used for this. A central service subscribes to the DIP parameters and republishes the information inside the distributed PVSSII system, for use by the sub-detectors.

Detector Safety System and Interlocks

The Detector Safety System (DSS, [8]) is a robust part of the ALICE DCS, designed for high availability and based on a redundant PLC system. It is designed to monitor the experiments environment (temperature, presence of cooling, water leaks) and to take automatic protective actions (cut power, close water valves) in case of anomalies.

Hardware interlocks are implemented at several levels. Sub-detectors have implemented various protection mechanisms on their detector equipment; high temperatures are detected on the electronics and automatically switch off the power to the failing piece of electronics. Also the DSS is used as an interlock system where independent sensors are available to detect anomalous conditions and DSS is programmed to take protective action.

4.5.3 Installation, Commissioning and First Operational Experience

The ALICE sub-detector parts have been built at the collaborating institutes around the world, transported to CERN, assembled and finally installed in the underground experimental hall at Point 2 of the LHC. The associated control applications have also been implemented by the respective sub-detector groups, according to the guidelines defined by the ALICE controls coordination. Before being installed in the final system at the experimental site these applications were first used at the institutes during the tests of the detector elements.

From early 2006 the infrastructure of the control system (computer cluster, network, database etc.) was installed at the experimental site to be ready to receive the sub-detectors and their associated controls applications. This started at the end of 2006 and the detector control applications, partitioned in 'local control', were used by the sub-detector groups for the commissioning of the sub-detector hardware. In parallel general infrastructure monitoring, interfaces to the services and safety systems were installed and commissioned. Also the ALICE control room with its operator consoles was installed and commissioned.

Once the individual sub-detectors had been installed and commissioned they had to be integrated into the global ALICE control system. This was a very critical issue knowing that the various applications, which had to work together, had been developed in so many places around the world. As from the end of 2007 regular Detector Integration Sessions were held with all sub-detectors to validate their control implementations and to integrate them in the central control system. These sessions were very instructive and although some inconsistencies were revealed all detectors were successfully integrated during spring 2008. Several global tests were also performed, where all sub-detectors were successfully operated by one central operator (Figure 9).

Several tests were also done with the online and offline systems in order to commission the integrated operation of the entire experiment.

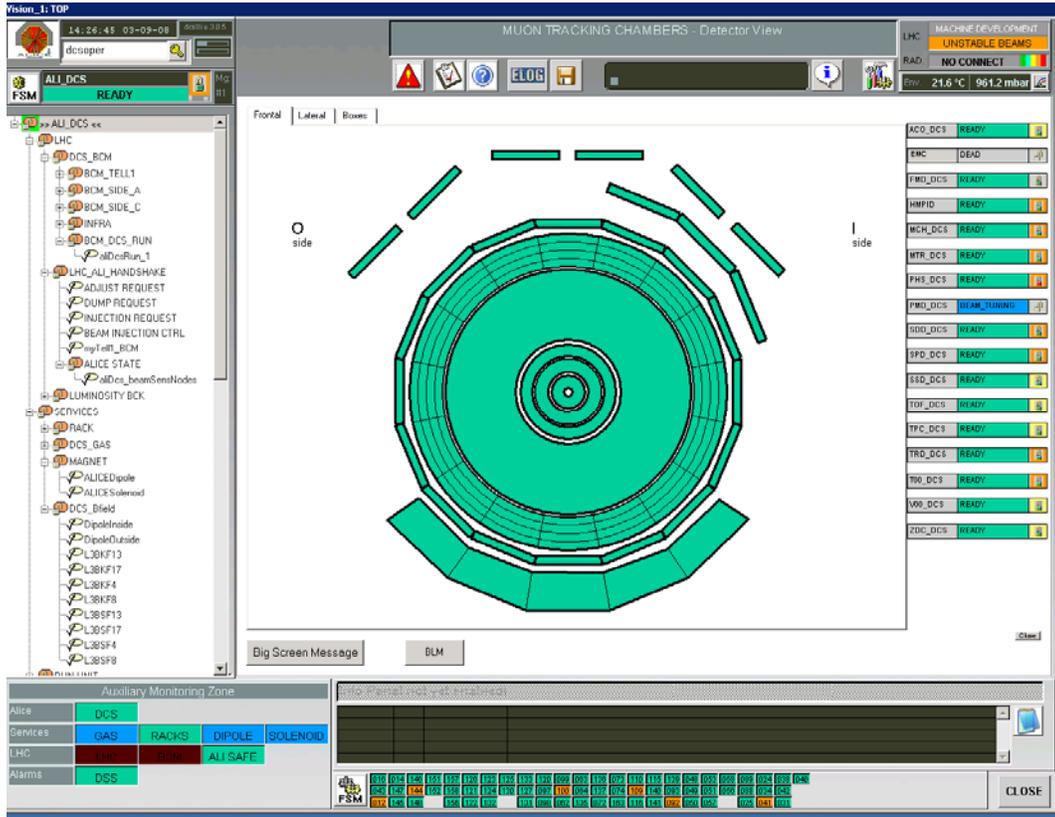


Figure 9: Typical central operator user interface

In addition to the above tests, the control system was used during several ‘cosmic runs’ (using cosmic rays to trigger the detectors) during 2008. During the first two data taking campaigns the sub-detectors were mainly operated in ‘local mode’ by the sub-detector experts, while the central operator was observer and had as main task to survey all safety related aspects of the operation. From summer 2008 onward, sub-detector operations were partly delegated to the central DCS operator.

During the injection tests into the LHC machine in the summer 2008 the interface with the machine was commissioned. During these tests the induced background levels could reach values that are potentially dangerous to some of the sub-detectors. The control system makes it possible for the central DCS operator to ensure that the sub-detectors are at any moment in a state compatible with the activities of the LHC machine.

4.5.4 Summary and Outlook

Despite the large number of sub-detectors that constitute the ALICE experiment, each with their own requirements in terms of controls, we succeeded to build a homogeneous and integrated control system. Already in an early stage a lot of effort was put in the coordination of the activities in the sub-detector groups, and in close collaboration with them we managed to deploy common solutions for similar requirements.

This effort largely paid off when we reached the integration stage earlier this year, and all sub-detectors could be integrated into a homogeneous central detector control system without major problems.

The first operation experience revealed some minor flaws in the current system that we are currently addressing, and will be addressed before the final commissioning runs foreseen in spring 2009. These issues mainly relate to the interface with LHC, in order to cater better for the extended injection periods that were not foreseen in the original scheme. Also the interface to the other online systems in ALICE is under review to streamline their interaction and with the aim to reduce the startup time and make the configuration and calibration more robust and efficient.

4.5.5 Acknowledgements

We would like to thank all control experts in the sub-detector projects and in the other online domains (DAQ, Trigger, ECS and HLT), as well as the control experts in the various CERN services, for their efficient collaboration and their patience with us. Without their effort it would have been impossible to come to the well integrated control system as it exists today. We are grateful for the collaboration in the Joint Controls Project, which showed once again that the total is more than the sum of the individuals. Finally we thank all the technicians for their efforts at the experiment site to install all the required infrastructure; only with equipment to control, cables to access it and computers to run the system, our control system could come to live.

4.5.6 References

1. T.J. Adye, A. Augustinus, M. Dönszelmann, T. Rovelli, R.L. Sekulin, G.R. Smith, "The Design and Operation of the Slow Controls for the DELPHI Experiment at LEP", Nucl. Instr. and Meth. A349 (1994) 160-182.
2. A.Daneels and W.Salter, "The LHC experiments Joint Control Project, JCOP", ICALEPCS99, Trieste 1999, and <http://itco.web.cern.ch/itco/Projects-Services/JCOP/welcome.html>.
3. P.Chochula, Cyber security in ALICE, ICALEPCS07, Knoxville, USA 2007.
4. B.Franek, C.Gaspar, "SMI++ - An Object Oriented Framework for Designing Distributed Control Systems", IEEE Trans. Nucl. Sci., Vol 45, Num 4, p. 1946-1950,
5. P.Chochula, A.Augustinus, L.Jirden, S.Kapusta, Handling large amounts of data in ALICE, ICALEPCS07, Knoxville, USA 2007.
6. B.Hallgren et al., "The Embedded Local Monitor Board (ELMB) in the LHC Front-End I/O Control system", 7th Workshop on Electronics for LHC Experiments, Stockholm 2001,
7. P.Chochula, L.Jirdén, A.Augustinus, "Control and Monitoring of the Front-End Electronics in ALICE", 9th Workshop on Electronics for LHC Experiments, Amsterdam 2003,
8. S. Lüders et al., "The CERN Detector Safety System for the LHC Experiments", ICALEPCS03, Gyeongju, Korea 2003,

4.6 The Fermilab Accelerator Control System

K. Cahill, L. Carmichael, D. Finstrom, B. Hendricks, S. Lackey, R. Neswold, D. Nicklaus, J. Patrick, A. Petrov, C. Schumann, J. Smedinghoff and G. Vogel
Fermilab, Batavia, Illinois, USA
Mail to: patrick@fnal.gov

4.6.1 Introduction

For many years the Fermilab physics program has been dominated by the superconducting Tevatron accelerator producing beams for many fixed target and the proton-antiproton colliding beam experiments CDF and D0. More recently, major experiments have used beam from intermediate accelerators. The MiniBooNE and MINOS experiments use 8 and 120 GeV beam respectively for neutrino oscillation studies. Several other experiments and test beams have also used 120 GeV beam. This paper describes the control system for the accelerator complex that was originally developed for the start of Tevatron operation in 1983. This system is common to all accelerators in the chain, and has been successfully evolved to accommodate new hardware, new software platforms, new accelerators, and increasingly complex modes of operation.

4.6.2 Fermilab Accelerator Complex

The Fermilab accelerator complex (Figure 1) consists of a 400 MeV linac, 8 GeV Booster synchrotron, 120 GeV Main Injector, 980 GeV Tevatron based on superconducting magnets, an anti-proton collection facility, and an 8 GeV anti-proton “Recycler” storage ring in the Main Injector tunnel. Beam is delivered to 8 and 120 GeV fixed target experiments, to an anti-proton production and accumulation facility, a high intensity neutrino source, and a 1.96 TeV proton anti-proton collider. The final Tevatron fixed target experiments ended in 2000. In 2001 a Tevatron collider run (“Run II”) began with substantial upgrades from the previous 1992-96 run and continues at this time. Prior to that, Fermilab had never mixed collider and fixed target running. However, in late 2001 an 8 GeV fixed target experiment (“MiniBooNE”) began operation, followed by 120 GeV fixed target experiments in early 2003, and the NUMI neutrino beam generated from the Main Injector in late 2004. The control system is required to support all these operation modes simultaneously.

4.6.3 Control System Overview

The Fermilab accelerator control system, often referred to as ACNET (Accelerator Network), is a unified system controlling all accelerators in the complex including all technical equipment such as water and cryogenics. ACNET is fundamentally a three tiered system (Figure 2) with front-end, central service, and user console layers. Front-end computers directly communicate with hardware over a wide variety of field buses. User console computers provide the human interface to the system. Central service computers provide general services such as a database, alarms, application management, and front-end support. The central database is a key component of the system by not

only providing general persistent data support, but also by providing all of the information to access and manipulate control system devices. Communication between the various computers is carried out using a connectionless protocol also named ACNET over UDP. The global scope of the control system allows a relatively small operations staff to effectively manage a very large suite of accelerators and associated equipment.

The control system was originally developed for the Tevatron, which began operation in 1983, and applied to all accelerators in the complex at the time. While the fundamental architecture has remained similar, in the years since there has been considerable evolution in field hardware and computing technology employed. This has allowed the system to handle new accelerators and increasingly complex operational demands.

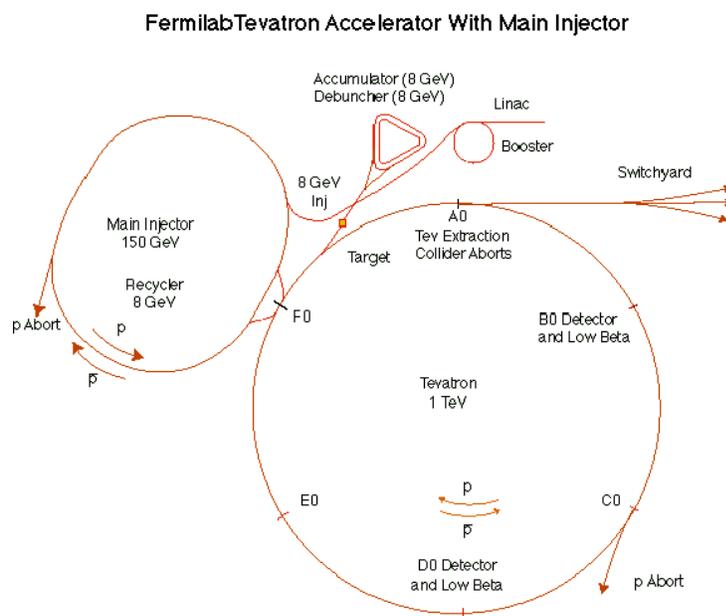


Figure 1: The Fermilab Accelerator Complex

4.6.4 Device Model

Access to the control system follows a device model. The ACNET system employs a flat model with names restricted to 8 characters. While there is no formal naming hierarchy, by convention the first character of the device refers to the machine and the second character is always a ":" For example, T: for Tevatron devices, B: for Booster devices, etc. Informal conventions in the remaining 6 characters provide some categorization by machine subsystems. Recently it has become possible to assign a 64 character alias, allowing a more verbose device name. However this is not yet in wide use. Each device may have one or more of a fixed set of properties including reading, setting, digital status and control, and analog and digital alarm information. Reading and setting properties may be single values or arrays. While mixed type arrays are not transparently supported, this is often done with the required data transformation between platforms done in library code. Device descriptions for the entire system are

stored in a central database. Entries may be created and modified using either a graphical interface or a command language utility known as DABEL. There are approximately 200,000 devices with 350,000 properties in the Fermilab system.

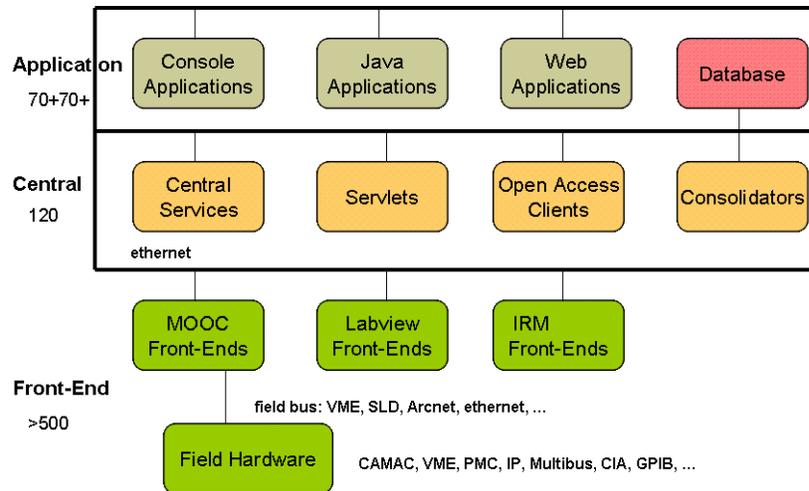


Figure 2: Control System Architecture

4.6.5 Communication Protocol

Communication among the various layers of the control system is done through a home-grown protocol known as ACNET. The ACNET protocol was created in the early 1980s and transferred accelerator data between PDP-11s and VAX systems over DEC's proprietary PCL (Parallel Communication Link.) As the decade progressed and newer hardware became available, IEEE 802.5 token ring and later Ethernet was added to the controls system's network and ACNET adapted accordingly. By the end of the decade, Ethernet and TCP/IP emerged as the dominant network technologies. So, ACNET became a protocol carried over UDP. This move provided three benefits. It allowed any TCP/IP-capable platform to use ACNET. It supported the use of commercial routers to make ACNET packets easily available across WANs, and it let the IP layer handle packeting issues. The ACNET protocol is currently supported on Linux and VxWorks platforms. The Linux implementation is written so that it should be easily portable to other UNIX-like operating systems.

ACNET was designed to be, foremost, a soft, real-time data acquisition protocol. Limitations on network bandwidth and processor memory at the time resulted in a very efficient design for returning machine data at high rates with minimal overhead. As a result, returning large data types can be awkward, but returning lots of small pieces of data (the typical case) works well.

Messages are directed at specific tasks on the target node; a daemon process receives and forwards the messages appropriately. ACNET is a multilevel protocol. At the lowest level ACNET peers communicate by one of two methods: Unsolicited Messages (USMs) and Request/Reply Messages. USMs are less "expensive" than Request/Reply transactions and are useful for broadcasting state to multiple clients.

Request/Reply communication, however, is the main workhorse of the control system. A requesting task sends the request to a replying task. This can either be a

request for a single reading, or a request for multiple periodic readings without re-requesting. A single packet may include requests for data from multiple devices. The replying task then sends one or more replies to the requester asynchronously. If the replier needs to stop the replies (due to an error, for instance), it can include an "End-of-Mult-Reply" status. Likewise, if the requester no longer wants the stream, it sends a cancel message to the replier, which shuts down the stream. Multicast requests have recently been added to the protocol, in which case the requestor will receive streams of data from all repliers in the multicast group.

Higher-level protocols atop the request/reply layer provide the specifics for data acquisition. The primary one is called RETDAT (RETURN DATA) and is used for simple data acquisition. It allows a process to receive a stream of data either at a periodic rate or whenever a specified clock event occurs. The newer GETS32/SETS32 protocol adds a more comprehensive set of event collection options and includes precise timestamps in the reply to aid in correlation of data across the complex. The Fast Time Plot protocol is used for acquiring device readings at high rates. To reduce the required network bandwidth, readings at rates up to 1440 Hz are blocked into single network packets and delivered to the requester several times per second. The Snapshot protocol specifies acquisition of a single block of up to 4096 points of data at whatever rate can be supported by the underlying hardware.

4.6.6 Timing System

ACNET makes use of several types of timing systems to coordinate operations of the accelerator complex. Overall system timing is provided via the timelines that are broadcast on the TCLK clock system. Individual accelerator beam related timing, associated with such devices as kickers and instrumentation, is supplied by the Beam Sync Clock systems. Slowly changing machine data (<720 Hz.) which is useful across the complex (accelerator ramps, machine state settings, beam intensities, etc.) is made available via the MDAT (Machine DATA) system. These timing system signals are available via both hardwire (fiber and copper) transmission and network multicast.

TCLK is the highest level clock system in the complex. It is an 8-bit, modified Manchester encoded clock transmitted on a 10 MHz carrier with start bit and parity. Clock events can be placed no closer than 1.2 μ S apart on the hardwire transmission. The network multicast of TCLK provides a 15 Hz transmission of groups of TCLK events that have occurred during the previous 67 msec period. This provides soft real-time information to user applications and central service processes without requiring a special clock receiver card. Timelines are groups of TCLK events that define machine cycles that repeat over a given time period. Timelines are made up of "modules" that define what happens within the complex (with required internal timing and machine state info) over the period specified by the module. A typical timeline module will include TCLK reset events associated with the Booster, Main Injector and the destination machine/experiment. A VME based front-end with special application software known as the Timeline Generator (TLG, described below [1]) provides a flexible user interface that allows operators to manipulate timelines as needed to meet changing operational conditions.

The various accelerator beam sync clocks (TVBS, MIBS and RRBS) are also 8-bit modified Manchester encoded clocks. However, their carriers are sub-harmonics of the given accelerator's RF frequency (RF/7, ~7.5 MHz). They all carry revolution markers

sourced from the low level RF (LLRF) systems along with beam transfer events synchronized to the revolution markers. This allows precise timing of kickers and instrumentation.

A more recent addition to the timing systems is the use of states. Machine states refer to phases of operation such as proton injection, antiproton injection, acceleration, etc. They are used for example by the LLRF systems to determine what type of RF manipulations will take place within a given machine cycle and when. The Tevatron Low Beta Sequence state is used to change magnet settings for the low beta squeeze and various other state variable transitions are used to trigger data acquisition. To trigger a state transition, any control system task may set a virtual device in a state server, and the transition is then forwarded by multicast or direct communication to other elements of the system. A few selected state values are transmitted on the MDAT network. This allows for timing flexibility beyond the 256 event limit set by TCLK. Also as state values are held in virtual devices, applications may query them at any time.

4.6.7 Supported Hardware

The ACNET control system comprises a variety of hardware and field buses that have evolved over the life time of the accelerator complex. While the functions have remained similar over the years, new hardware technology has been integrated in to the controls system whenever possible given the schedule and budget. The evolution of the ACNET system's hardware mirrors the evolution of controls hardware technology in general.

Early on, the controls system hardware included PDP-11s connected to Lockheed MAC 16 minicomputers with CAMAC as the field bus. Numerous custom CAMAC cards were developed for the timing system, data acquisition and control functions. Two notable and widely used systems, MADCs and ramp generators, are described in more detail below.

4.6.7.1 Data Acquisition

Analog signals are digitized by in house designed multiplexed analog to digital converters (MADCs) which are connected through a CAMAC interface. The MADCs allowed multiple users to sample 14 bit data from up to 128 channels per MADC at a maximum frequency of ~90 KHz for a single user/single channel. Early models allowed 6 simultaneous continuous fast time plot channels and the newer model of CAMAC interfaces allow 16 simultaneous plot channels. Data can be sampled on any clock event, external input or at a programmable rate.

The MADCs have served well for nearly three decades and are now being replaced with HOTLink Rack Monitors (HRMs). A variety of commercial and custom digitizers are used for specialized high rate applications. There are still significant systems that are controlled with CAMAC equipment and thousands of channels are connected through MADCs.

4.6.7.2 Ramp Generators

During the course of operations many power supplies must be ramped. Often they must be ramped in a different manner depending on the type of beam cycle. To satisfy this requirement we have developed flexible ramp generating hardware which can save

sets of tables locally and play the appropriate ramp on specific clock events. This allows different types of beam cycles to be interleaved seamlessly without having to reload ramp tables using higher level software for each cycle.

4.6.7.3 Field Buses

As microprocessor technology progressed, VME and VXI based designs were incorporated into the controls system and processing was distributed. MAC 16s were replaced with VME and VXI front-ends. Eventually VME became the standard control system front-end platform.

Newer power supply controls are most often implemented in Programmable Logic Controllers (PLCs) and many devices come with Ethernet connectivity. The ACNET control system has been interfaced to several popular manufacturer's PLCs.

GPIB and Ethernet connectivity to instrumentation allows for remote diagnostics including oscilloscopes, signal generators, spectrum analyzers, etc.

4.6.8 Front-End Systems

Data from hardware devices enters the Fermilab control system through the front-end computers. These computers are responsible for acquiring the data from the hardware or field bus and responding to the timing system to ensure prompt collection. These closest-to-the-hardware nodes communicate with the rest of the control system using the ACNET protocol. Another important function is to provide a mapping between the central device database and the actual hardware readings and settings. At start-up time, a front-end may have its device settings downloaded from the central database. To implement these common tasks, three different architectures have evolved at Fermilab: MOOC (Minimally Object Oriented Controls), IRM (Internet Rack Monitor) [2], and OAC (Open Access Client) [3].

The IRM software architecture provides 15 Hz hard real time performance to match the pulse rate for the Fermilab linac. It also provides synchronized data collection across all the 15Hz IRM nodes. Custom processing is possible by adding "local applications" to an IRM node. The IRM architecture is built into a standard VME crate providing multiple channels of general-purpose analog and digital I/O. This off-the-shelf I/O capability of the IRM makes it a good choice for many applications with about 185 in use and it is the standard for controls in the linac. The HOTLink Rack Monitor (HRM) [4] provides more analog channels and higher digitization rates in a more modern hardware architecture.

MOOC nodes are also VME-based, built on the vxWorks real-time operating system running on PowerPC based computers. MOOC provides more customization and varieties of acquisition schemes than the IRM. In MOOC, in object-oriented fashion, the developer writes a software class to support a type of device, and then creates an instance of this class for each device. Thus there is great flexibility in device support, while the MOOC framework provides all interactions with the timing system and the ACNET communications. There are roughly 275 MOOC systems in the Tevatron, Main-injector, and anti-proton source. Data is acquired from a variety of field buses, including VME, Arcnet, CAMAC, GPIB, and others.

OACs are front-ends that typically run on centralized server nodes using a framework written in Java, with no special hardware device connections. OACs use the same communication protocols as other front-ends, but their position in the system

gives them easy access to the central database and data from all other front-ends. Access to the timing system is via the above described multicast. Besides providing utility functions such as virtual devices, some typical tasks performed by OACs include:

- Computational combination of data from other front-ends, including database driven algebraic expressions or custom Java code to perform emittance calculations for example.
- Ethernet-based data acquisition, including commercial hardware such as oscilloscopes, PLCs, or custom Ethernet-enabled FPGA devices.
- Process control, including finite state machines, PID loops, and beam trajectory stabilization.

Additionally, the Fermilab data loggers are built on the OAC architecture. There are about 120 OACs plus another 70 data loggers in the system.

Besides these common front-ends, there are also around 25 systems running LabVIEW. They act as front-end nodes in the control system by the inclusion of LabVIEW modules programmed to follow the ACNET communications protocol. LabVIEW front-ends are typically used in instrumentation systems such as wire scanners or synchrotron light systems. GUIs developed in LabVIEW are generally used only by instrumentation experts. With the systems connected as front-end nodes, data is available to standard control system services such as data logging and alarms. Standard applications provide the required subset of the LabVIEW functionality to operators.

These front-end architectures have proven successful in fulfilling the key requirements of data acquisition, timing system response, communications protocol support, and database mapping across the diverse accelerator chain at Fermilab, supporting both legacy hardware and new systems.

4.6.8.1 Central Services

The central tier of this three tiered control system houses central services. Examples of central service functionality include alarm collection and distribution, data logging, and servlets supporting web applications.

4.6.8.2 Data Logging

Logging of accelerator data is done by tasks writing to MySQL databases distributed over 70 central service nodes. Each distributed data logger supports 18 tables having a nominal capacity of 60 devices circularly sharing a $5e8$ point data space. A full logger overwrites old points in 1446 days when sampled at a 15 second interval. Overwrite times are shorter or longer dependent on the number of devices in a logger's table and the sampling frequency. More than 50 of these loggers organized by machine or department and log data on periodic rates, TCLK events or software state event transitions. The remaining loggers are reserved for specific diagnostic functions such as TCLK event or software state transitions or device setting modifications. Other loggers provide archives of logged data. So that no data is lost when a specific logger wraps around, each day's data, currently over 5 Gigabytes, is transferred to permanent archives of several terabytes on spinning media.

4.6.8.3 Sequenced Data Acquisition

Sequenced Data Acquisition (SDA) saves scalar, snapshot, and fast time plot data during defined periods of important machine operations. The most complex example is for Tevatron Collider stores, where a typical shot collects about 25K scalars, 1K snapshots, and a few hundred fast time plots with specific collection requirements. The data is used for post-mortem as well as shot to shot analysis to study trends in Collider performance. An extensive suite of tools automatically produces summary information and plots available via the web. This facility is extremely valuable for studying trends in accelerator performance.

4.6.8.4 Save/Restore

Save/Restore services provide for operator initiated saves of the complex for future display or restore. Four times a day, automatic saves are initiated that encompass nearly all addressable devices on operational nodes. Besides providing a backup to operator initiated saves, the big saves expose lurking data acquisition problems, and the data is reflected into loggers that although is sparse provides a quick historical assessment of nearly all the operational channels of the control system.

4.6.8.5 Alarms

Each front-end is responsible for scanning its devices for alarm conditions. When detected, the front-end sends an ACNET message to a central alarm server [5]. The central alarm server supports an alarm protocol for reporting and clearing alarms from the hundreds of alarm reporting nodes and an alarm distribution methodology that includes multicasting of alarm updates to nodes that service hundreds of alarm display clients. This architecture provides excellent scalability with both the number of alarm producers and alarm clients.

4.6.8.6 Front-End Download

When a front-end system is rebooted, it needs to know the current setting values and alarm thresholds for its devices. As front-ends do not have easy access to the main database, a front-end download service provides this functionality. A separate setting service keeps the database up to date. Front-ends forward new setting values to this server that then saves them.

4.6.8.7 Accountability

Various central services record considerable information about control system activities. The data acquisition setting routines forward settings to a service that logs settings for accountability providing application access to who, what, when, and where a setting was performed. Also logged are data acquisition jobs initiated by Java programs, data acquisition errors, application usage, CPU utilization by node, and database queries. This information is made available via web based reports.

4.6.8.8 Data Acquisition Engines

Java clients do not communicate directly with front-ends but instead go through Data Acquisition Engines (DAEs). Performing all data acquisition through the central

layer allows more reliable control of security and settings logging, better isolating the front-ends from improper requests. Also the engines perform consolidation of common requests from Java clients across the control system. A single request is made to the front-end and the data are then distributed to all requesting clients. The engines also simulate data acquisition conditions which may not be directly supported by some front-end systems.

4.6.8.9 Servlets

Several Java based servlets provide control system access to web based applications including the parameter page, logger display, and SDA viewer applications. Logged data acquisition errors, CPU utilization, logger fetch, and SQL statement logging are other examples of servlet-provided access to web based displays.

4.6.8.10 Time-Line Generator

The Time Line Generator (TLG) is an ACNET client-server system that generates timelines which place 256 possible events on to the TCLK network and 16 bit states on the MDAT network. The TLG denotes a move away from flat timeline generation to the production of rule-based structured timelines. Each timeline is built on a user application and then executed on the server, a MOOC front-end. A structured timeline is represented by a set of modules and a rule set. A subset of this rule set includes the priority of each module, starting time of each module, the number of repetitions and the end time of each module. Placement of each module is governed by these rules. Each module consists of a set of events, states and its own rule set. These rules govern event and state placement, linkage between different event types and additional actions. The overall result of the timeline is the generation of a set of events and states with specific rules governing how each event and state is placed and how they react to other events and states. The move to rule-based structured timelines has resulted in a great deal of flexibility and robustness being built into the system. Users can now target a specific component of the timeline to be changed by simply swapping modules in and out of the current timeline. The rule set would then allow the timeline events and states to adjust their priority and placement based upon these changing user specifications. This methodology allows supporting very complex modes of operation of the accelerator chain. Furthermore modifications can be made and implemented very efficiently. The TLG system makes it very straightforward to quickly switch from operations that include NuMI/MINOS, pbar production, MiniBooNe and 120 GeV fixed target, to a subset of these or to a Tevatron injection timeline.

4.6.8.11 Experiment Communication

Experiments need to obtain accelerator information such as accelerator state, beam intensities and losses. Also it is sometimes useful for the experiments to send information to the accelerator, such as colliding beam luminosities and collision points measured by their detectors. As the experiments do not have direct access to most of the accelerator control system and also have a different programming environment, this is accomplished by a central service communicating via the XML-RPC protocol. APIs for XML-RPC are available in many languages. Each experiment writes its own applications to obtain desired accelerator information and they may also set virtual accelerator devices with experiment information through this service.

4.6.8.12 Databases

The Fermilab control system utilizes several data storage facilities. The predominant data storage is located in several Sybase relational databases that total over 100GB of data. There are separate databases for all device and scaling information for the control system, application data storage, lattice information, Save/Restore, and SDA data for shot analysis. Datalogger data is stored in distributed MySQL databases containing a total of over 7TB of data. A filesharing service is also available to manage shared access to historical data files used by older applications.

4.6.8.13 Application Frameworks

Applications are written using one of two frameworks. In the CLIB framework, applications are written in C/C++ for the Linux platform. Graphics is based on a custom library on top of basic X-Window calls. A newer framework using the Java language allows for development of applications that can run on any platform and provides a more modern look and feel. Both frameworks provide common functions and a common look and feel within that framework. Both capture all application code required for operations in a CVS repository and provide a place to launch them.

4.6.8.14 CLIB Framework

There are roughly 600 Linux-based applications which are used to operate the Fermilab accelerator complex. They are written primarily by people who operate the accelerator including machine physicists, engineers, and operators rather than by individuals from the controls department.

The structure of these applications is rather simple consisting basically of an infinite event processing loop. The basic events are initialization, user interrupt, periodic (15 Hz), and termination. The initialization event occurs once and allows the programmer to set up any initial conditions, and the termination interrupt occurs once at the end of the program to allow for operations such as saving files and cancelling device requests. The periodic interrupt supports updating displays while the user interrupt event allows the program to respond to a user's request. There are other events that are used to a lesser degree including the notification that a global control system state value has changed and the occurrence of a clock event as examples.

To support the writing of these programs, there is a large shared library named CLIB which stands for Console Library. This library contains approximately 1700 entry points which support such topics as data acquisition, user interface, data manipulation, program control, network messaging, error message handling, ACL (Accelerator Command Language) support, and other miscellaneous routines. CLIB is written and maintained by the controls department and is linked at runtime which allows global functionality changes as well as bug fixes to be implemented easily across the entire suite of applications.

In addition to CLIB, there are other smaller libraries which are called user libraries. Many of these libraries are written by controls department personnel while others are written by machine physicists and engineers.

To simplify the creation and modification of applications and user libraries, there is a C/C++ Software Development Environment (SDE). Its first purpose is to allow users to develop application programs without much software development expertise. Users must know C/C++ but need not understand Makefiles, compiler/linker options, and

revision control systems. A second goal is to make sure operational software is never lost. To this end, the SDE automatically places files in a revision control system, in particular CVS. Capturing the entire source in a revision control system allows the SDE to provide a retreat functionality. A retreat of an application program can be done by any user if it turns out in hind-sight that the most recent change(s) are causing problems. The SDE also provides a facility to develop new libraries to avoid code duplication between similar applications. Libraries can be either statically or dynamically linked.

4.6.8.15 Java Framework

Besides the CLIB framework, the Fermilab Accelerator Control System includes a newer infrastructure supporting user applications written in Java. These applications can run under Windows, Linux, Solaris, Mac, and FreeBSD platforms on both central nodes and user computers. There are also a number of web applications providing data to the users via the HTTP protocol. The Java infrastructure consists of three major parts: the application framework, the application index, and the building system.

The Java Application Framework [6] facilitates development of standardized control applications by providing an implementation of a uniform Swing look-and-feel and several core services. This includes authentication, logging, printing, screen capture, submission of data to an electronic logbook, and access to application properties in a central repository. Kerberos V5 is used as a common method of authentication for both standalone and web applications, via a customized Kerberos client [7].

The Application Index [8] is a central web-based database of all Java Controls applications and an application that provides for their launching via Java Web Start. For each program, the database provides a URL of a corresponding JNLP file (a standard descriptor understood by the Web Start client). The URLs are combined in a tree, according to the application's fields of use. The Index also supports searching programs by name, description, and author. The JNLP files are generated dynamically upon each request using current information in the database, such as the program's class path, initial and maximum memory heap size, required version of Java Virtual Machine, and others. This allows for changing of runtime parameters quickly from a single place. The Application Index also allows viewing of central logs and statistics on running applications collected by the Application Framework.

Currently all Java code, including locally developed code and third party libraries, is maintained in a single source tree in CVS. Developers write code using their method of choice. A custom Eclipse plug-in is available that simplifies development of accelerator applications. When ready to install new code in the system, developers commit code to CVS and request a new release via a web interface. A building system (which is a set of Perl and Ant scripts on a central server) schedules new builds, checks out relevant modules from the repository, compiles the code, creates and signs jar files, and deploys the binaries. The latest production version of Java Controls is made available through a shared drive on a file server and over the web. The former is used by various server-side processes, such as servlets and OACs, and for development. The latter is mainly for the web-startable client applications. All released jar files (including third-party libraries) are properly signed so that their origin can be verified using the department's public key certificate.

4.6.9 Key Applications

Described below are some of the more important core applications in the system.

4.6.9.1 Parameter Page

The Parameter Page (Figure 3) is a general purpose program which allows display and control of lists of accelerator devices. For each device it displays the device name, descriptive text, reading, setting, alarm limit/status, and digital status. Device settings, alarm limits, and digital control can be modified by users. Groups of devices may be combined into “knobs” that allow correlated changes to be made to members of the group.

Accelerator subsystems have been organized into a hierarchy of persistent parameter device lists called subpages by the subsystem experts and operators. Users can easily add devices to subpages at any time by entering device names.

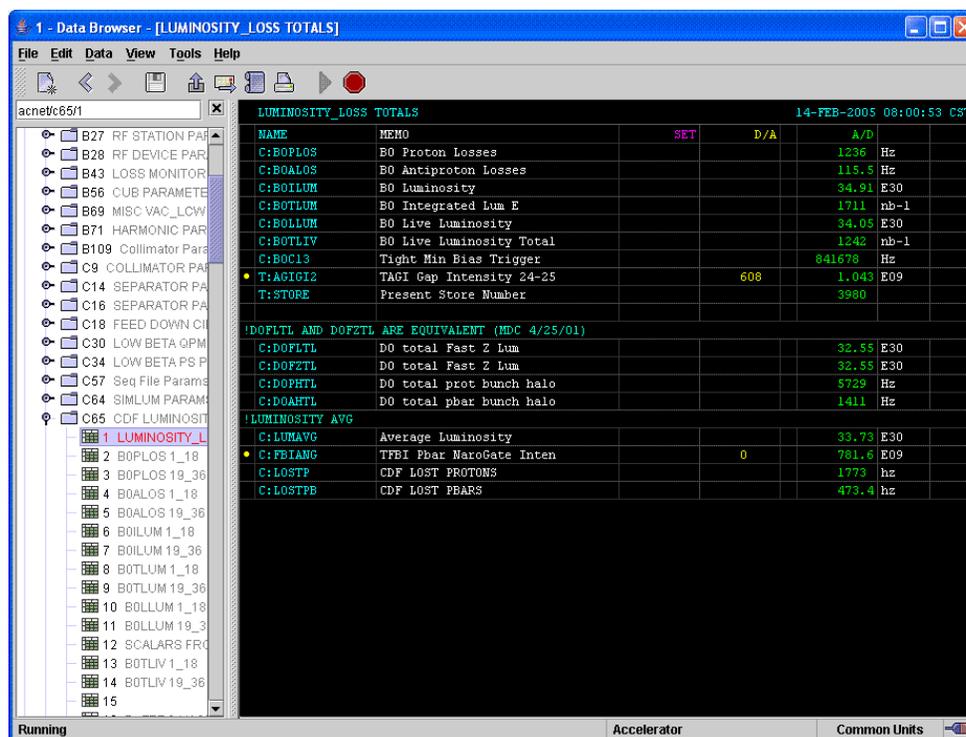


Figure 3: A Parameter Page in the Java Framework

4.6.9.2 Fast Time Plot Utility

The Fast Time Plot utility allows control system users to plot devices in real time. Data sample rates up to 1440 Hz are supported. The x axis can be referenced to any accelerator clock event, time since the plot was started, or another device. Device readings can be combined into simple expressions using add, subtract, multiply and divide operators before being plotted.

4.6.9.3 Snapshot Plot Utility

The Snapshot Plot utility supports plots with data collection rates of up to 20 MHz. Hardware determines what rates a given device supports. In addition to displaying snapshot plots for individual users, snapshot plots can be initiated by the SDA utility and the Snapshot Manager for automated data collection and analysis.

4.6.9.4 ACL

ACL (Accelerator Command Language) is a simple to use but powerful scripting language which is designed to operate an accelerator complex. It contains syntax for using control system device specifications in much the same way as variables are used in standard programming languages. There are over 160 commands ranging from controls-specific ones such as read and set to generic if and looping statements. There are also over 100 supported intrinsic functions which can be used in expressions.

The main goal when creating this language was to empower end users to encode powerful algorithms for controlling the accelerator complex. The people who best know what needs to be done to fix a problem or to add a new functionality are often not programmers by nature. They could make up a software specification to solve the problem, but this involves delay and often details are lost in translation. ACL is a solution to this situation.

Machine physicists, engineers, and operators have created many ACL scripts which are executed in the environment of the Fermilab Sequencer. These scripts have helped to make the operation of the accelerator complex more robust and efficient.

ACL scripts can be executed in a number of environments. They can be executed as mentioned above as atomic commands in the Fermilab Sequencer. They can also be executed as embedded commands in parameter pages, and they can be used to update displays and to implement machine control in Lex SA. ACL scripts can be executed directly within any application program using a library routine. In this mode, ACL variables can communicate results to the calling application. There is also a command line interface which is useful for quickly diagnosing problems with the control system.

4.6.9.5 Sequencer

The Fermilab Sequencer [9] is the primary program for coordinating the operation of the accelerator complex. It is especially important for handling the complex sequence of operations necessary for injecting the protons and antiprotons into the Tevatron Collider and bringing them to collision.

The Sequencer was designed so that machine experts could configure it. It could be thought of a simple programming environment using a custom command set consisting of roughly 60 entries. One of these commands also provides an entry into a wider programming environment by executing an ACL (Accelerator Command Language) script. These commands are organized into groups which are referred to as aggregate commands. An aggregate command is usually thought to express the execution of a major accelerator state change. Aggregates are in turn grouped into modes. Modes generally represent a portion of the accelerator complex such as the Main Injector or the Tevatron Collider.

The Sequencer supports two execution modes, command edit and command execution. The Sequencer is edited by machine experts and operators directly from the program without any formal programming taking place. Once the editing operation is

finished, the insertion, modifications, or deletions are stored to a relational database table. There are tables for each individual command type as well as tables for organizing the commands in the proper sequence.

In execution mode, users can choose to execute a single command, a set of commands, or an entire aggregate command. If multiple commands are selected, they are executed sequentially until completion for the most part. There are a few command types which have options for spawning off operations in parallel when sequential operation does not have to be strictly enforced. When an error is encountered, an alarm message is displayed to the user and execution ceases. The user can then decide at what point to resume execution.

Multiple Sequencer modes operate in concert to manage the operation of initiating a store in the Collider. These modes typically coordinate their execution by the use of global control system state values.

4.6.9.6 Lex SA

Lex SA is a user-editable synoptic graphics program based on the CLIB framework. It was created to allow end users to create displays including those which appeared like schematic drawings of various accelerator systems. It supports a set of graphic objects which include drawing primitives, scanned images, and objects which will display the readings of control system values in various ways. There are also objects which can execute ACL (Accelerator Command Language) scripts to change the display or to make device settings. This feature makes the display capabilities extensible by the end user.

Lex SA is actually comprised of two programs, an editor and a display program. In the editor program, users can drag and drop the various types of graphical objects onto the canvas and can edit their properties. Users can also create and save complex graphical objects comprised of primitive objects which they can save away for future use. The entire display can then be saved to a relational database for use by the display program. There is a table for each type of object as well as tables to store how the objects are joined together to make the display.

The display program can be launched from several dedicated applications including the editor. A Lex SA display can also be mapped to each subpage of any parameter page. Once the display is initialized and running, users can click on fields to execute ACL scripts and they can knob analog setting values. There is also a companion program that supports dynamically building a parameter page of devices by clicking objects in the display. There are also objects that when clicked will start up another display allowing individual displays to be linked together.

4.6.9.7 Java Synoptic

Synoptic [10, 11] is a client-server system for graphical data representation, similar to Lex SA and EPICS EDM screens. In addition to providing a more modern look and feel than Lex SA, it can display "live" images in conventional web browsers using very low bandwidth and, in most cases, without the need of additional software. As in the case of Lex SA there are separate display and editor applications.

In a simplest use case, a read-only Synoptic display is opened as a web page (Figure 4) in a regular browser. Upon request, the Synoptic web server makes up a layout of the display, initializes data acquisition, and periodically generates images in a Scalable

Vector Graphics (SVG) format. JavaScript code in the browser polls the server every 1-2 seconds for new graphical data and feeds it to an SVG viewer. Since the new image usually looks very similar to the previous one, it is sufficient for the client to receive only differences between successive images. The latest Mozilla Firefox, Apple Safari, and Google Chrome include embedded SVG viewers. Currently Microsoft Internet Explorer requires a third-party plug-in. All listed browsers show SVG images fairly consistently and render partial updates without flickering.

To enable settings from Synoptic to the control system, a display has to be opened in a Synoptic Viewer application. All data acquisition tasks in this case are started locally, and the display's image is rendered directly on the application's canvas. Unlike the web interface, Synoptic Viewer may only be used from certain locations within the lab for security reasons.

A specialized graphical editor called Synoptic Builder is used to create and edit the displays. Like Synoptic Viewer, this is a console Java application, normally launched through Web Start. The builder includes a library of components that can be placed on a display. There are separate components for data acquisition, data transformation, and data visualization. The first two groups are hidden at runtime. The components are interconnected with data pipes. The builder also allows static images and symbols to be included. Synoptic displays are stored in XML format either locally (for later use in Synoptic Viewer) or in a central CVS repository (for both Synoptic Viewer and the web interface).

4.6.9.8 Web Applications

A number of applications have been written that run entirely in a web browser. This includes many tools to view SDA data, and a parameter page and device database viewer. Web applications currently do not follow a standard framework. They are written using generic servlets deployed on Tomcat servers with client code written as Java Server Pages (JSPs) or Javascript.

4.6.10 Console Infrastructure

CLIB applications must run under a specialized console environment. This consists of a set of tasks that launch and manage applications, perform data acquisition, handle graphics rendering, and other functions. The console applications framework has evolved from earlier versions of the control system and thus has an older look and feel. Application programs are started from an Index Page program (Figure 5) which displays menus of programs pertaining to each accelerator. Programs have a main window for user interaction and optional windows to display graphics. Applications are run on any of 75 application server computers which contain the complete console framework environment. The framework uses the X Window system for user interaction and display. User workstations can be any internet connected computer which is running an X Window server. Also a special version of the console framework has been created that renders its graphics to a Java applet. This allows running a console from any web browser without installing special software. For security reasons, only a subset of applications may be launched in this environment and settings are not permitted. A "CLIB Peeker" facility [12] allows one to view internal information about any currently running application.

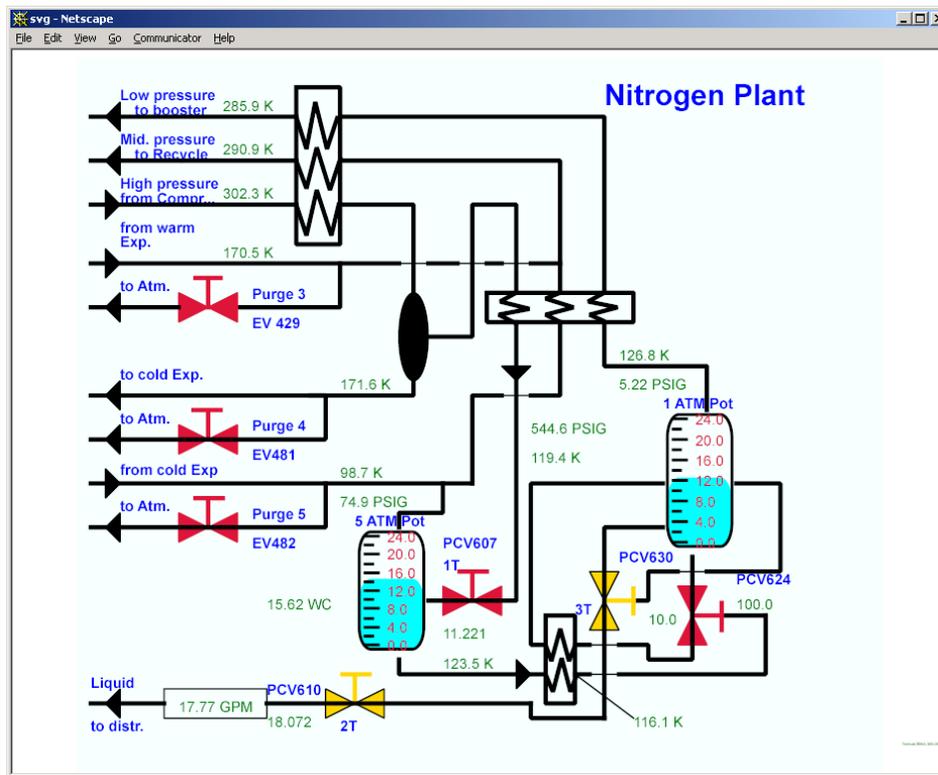


Figure 4: An example synoptic display in a web browser

For Java applications, a separate web-startable Index Page application or Java applet is used. Recently it has become possible to launch Java applications from the above consoles providing a single environment for all applications.

4.6.11 Security

The control system is on a dedicated network inside a firewall that restricts access both in to and out of it. This greatly limits the probability of computer compromise from external attacks.

In such a large and diverse system, it is also desirable to control setting capability even among accelerator personnel. To accomplish this, “classes” are defined and applied to both people and remote consoles. Specific devices can only be set by those in certain classes. Furthermore, applications run from outside the main control room start with settings disabled. They must be manually enabled by the user. These features reduce the probability of accidental settings disrupting operations.

Line	Page	Page Title
1		SDA DISPLAY
2	25	LOW BETA QUADS
3	26	
4	27	
5	28	
6	29	
7	30	LOW BETA QPM PARAM
	31	LOW B SCALERS(QPM)
9	32	B0/D0 QPM Params
10	33	
11	34	LOW BETA PS PARAMS
	35	LOW BETA CIRC BFRS
13	36	
14		SEPARATOR STAT/CTL
15	38	Separator Bakeout
16	39	TEV TUNE DISPLAY
17	40	TEK 3052 CONTROL
18	41	TEV SCHOTTKY CNTRL
19	42	TEV TUNE MEAS
20	43	TEV CHROM MEASURE
	44	TEV TUNE DISPLAYII
22	45	TEV BPM BY FRAME
23	46	BETASCAN
	48	Sequencer
	49	TeV Ramp Build
	50	TeV Orbit Pgm(TOP)
	51	
	52	
	53	
	54	
	55	
	56	
	57	Seq File Params
	58	
	60	COLLIDER SIMULATN
	61	LUM. CALC & LIFE
	62	LUMINOSITY CALC
	63	LIFETIME PLOTS
	64	SIMLUM PARAMS
	65	CDF LUMINOSITY
	66	
	67	
	68	
	69	STACK/LIFETIME SA

Figure 5: A Console Index Page

4.6.12 Past Evolution

Over the many years since ACNET has been developed enormous advances have been made in computing and hardware technology rendering many components of the original system obsolete. The control system has been able to evolve to take advantage of new technology as well as deal with the increasing operational demands of the complex.

While much field hardware remains in CAMAC, there is now a rich diversity of VME, VXI, Multibus, GPIB, and Ethernet connected hardware. The latter includes commercial scopes and spectrum analyzers as well as custom developed hardware.

Front-end systems were originally PDP-11 and Lockheed-Martin MAC-16 computers. These gave way to i386 Multibus and 68000 VME based systems running the MTOS operating system. All of these older systems have now been replaced by VME based 68040 or Power PC processors running the VxWorks or pSOS operating systems.

Console applications originally ran on PDP-11 computers with custom graphics. These gave way to VAXStations with X-Window graphics. In recent years all VAX software was ported to Linux and now runs on standard PCs.

Communication via the ACNET protocol originally used Digital PCL11-B links. These were migrated to IEEE 802.5 token ring links, and now this traffic travels exclusively over Ethernet.

Programming was originally done in FORTRAN and assembler, and later C. Now C++ is supported, and Java is supported for higher level software. The Sybase relational database was introduced, replacing older VAX based databases.

Hardware subsystems have been continually replaced as needed. Most notably the beam position and beam loss monitor systems for the Tevatron and Main Injector were

replaced in recent years. As these are large systems and there has been limited accelerator shutdown time, these upgrades had to be done in a staged manner during operational periods. Partial old and new systems had to coexist until the replacement was complete.

4.6.13 Future Directions

The Tevatron Collider is currently scheduled to end operation by October, 2010. The complex will then be upgraded to increase the intensity delivered to the neutrino physics program. The new NOvA experiment is expected to run until at least the late 2010's, and other new experiments are expected to make use of 8 GeV beam. The ACNET control system will continue to be used during this era. Obsolete hardware will be replaced as needed, and some effort will go toward modernizing the CLIB application environment.

An 8 GeV superconducting linac, known as Project X [13], has been proposed to further increase the beam intensity available for these and other new experiments. Prototype accelerators, HINS [14] and NML [15], for the Project X linac are currently under development at Fermilab. Until recently they have used the EPICS [16] and DESY DOOCS [17] control systems and been independent of the main ACNET system. A number of front-ends and synoptic display screens have been developed. Effort is currently underway to integrate these facilities back into ACNET. A prototype integration of an EPICS IOC under MOOC has been developed in the style of the EPICS2TINE interface [18]. EDM has been extended to support communication via the ACNET protocol as well as EPICS Channel Access. This allows inclusion of ACNET devices on already developed screens. EDM screens may now be launched from ACNET consoles. Clock events from these facilities have been added to the general TCLK clock event multicast and are available to data loggers and other central services. The current plan is to base the core Project X control system on ACNET, while supporting EPICS IOCs as needed. This will allow other EPICS based labs developing subsystems for Project X to work in the system with which they are most familiar.

4.6.14 Summary

Though originally developed in 1983, the ACNET control system has evolved to meet the increasingly complex operational needs of the Fermilab accelerator chain as well as take advantage of new technology developed since then. Its very modular nature has allowed both hardware and software systems to be upgraded as needed with minimal disruption to operations. The very efficient ACNET communication protocol has handled the continually increasing number of computers and associated volume of data. Solid development frameworks at all levels of the system have met the needs of developers and promoted commonality in the code. Powerful yet straightforward to use core applications such as the Parameter Page, plotting programs, Sequencer, and Accelerator Command Language make the system very accessible to operations personnel. Extensive logging of accelerator data as well as events, settings, errors etc. greatly aid diagnosis of subtle problems in this very large system. With the recent port of all application software to Linux from VAX/VMS, ACNET will be viable for the projected lifetime of the upgraded neutrino physics program and should form a strong basis for the control system of the proposed Project X accelerator.

4.6.15 Acknowledgements

The Fermilab control system has been developed by numerous people over many years. This includes not only controls department members, but machine specialists, operations and other technical support personnel as well.

4.6.16 References

1. L. Carmichael, "Automated Task Scheduling Using Multiple FSMs at Fermilab," Proceedings of the 1997 ICALEPCS, Beijing, China (1997).
2. R. Goodwin, M. Kucera, and M. Shea, "Use of Small Stand-alone Internet Nodes as a Distributed Control System," Proceedings of the 1993 ICALEPCS, Berlin, Germany (1993).
3. D. Nicklaus, "Java-based Open Access Front Ends in the Fermilab Controls System," Proceedings of the 2003 ICALEPCS, Gyeongju, Korea (2003).
4. A. R. Franck, R. W. Goodwin, P. A. Kasley, M. Shea, "HOTLink Rack Monitor," Proceedings of the 2001 ICALEPCS, SLAC, Stanford, CA (2001).
5. S. Ahn, "Fermilab Beams Division Alarms Processing System," Proceedings of the 1999 ICALEPCS, Trieste, Italy (1999).
6. <http://www-bd.fnal.gov/controls/java/framework/af-guide.pdf>
7. A. D. Petrov and D. J. Nicklaus, "Secure Client Tier for the Accelerator Control System," Proceedings of the 2005 ICALEPCS, Geneva, Switzerland (2005).
8. <http://www-bd.fnal.gov/appix>
9. T.B. Bolshakov, A.D. Petrov, S.L. Lackey, "Synoptic Display - A Client-Server System for Graphical Data Representation," Proceedings of the 2003 ICALEPCS, Gyeongju, Korea (2003).
10. <http://synoptic.fnal.gov>
11. J. Annala, "The Fermilab Sequencer used in Collider Operation," Proceedings of the 1995 ICALEPCS, Chicago, IL (1995).
12. J. Wang and B. Hendricks, "A Diagnostic Tool for Console Applications of the Fermilab Accelerator Control System," Proceedings of the 1997 ICALEPCS, Beijing, China (1997).
13. S. Nagaitsev, "Fermilab's Project X," Proceedings of the XXIV Linear Accelerator Conference, Victoria, B.C. Canada (2008).
14. R. Webber, "Overview of the High Intensity Neutrino Source Linac R&D Program at Fermilab," Proceedings of the XXIV Linear Accelerator Conference, Victoria, B.C. Canada (2008).
15. B. Chase, M. Votava, M. Wendt, "Controls, LLRF and instrumentation systems for ILC test facilities at Fermilab," Proceedings of the 2007 Particle Accelerator Conference, Albuquerque, N.M. (2007).
16. <http://www.aps.anl.gov/epics>
17. <http://doocs.desy.de>
18. P. Duval *et. al.*, "The Babylonization of Control Systems," Proceedings of the 2003 ICALEPCS, Gyeongju, Korea (2003).

4.7 THE DIAMOND LIGHT SOURCE CONTROL SYSTEM

Mark Heron, Diamond Light Source, Oxfordshire, UK
Mail to: Mark.Heron@Diamond.ac.uk

4.7.1 Introduction

Diamond, a third generation 3GeV synchrotron light source, began routine operation in January 2007. The Diamond Control System provides integrated control across the accelerators, photon beamlines and conventional facilities. It has further provided a high level of integration of the technical systems and good functionality to operate the accelerator, from day one. The scope of the Control System includes all supervisory controls, interlocking and protection of equipment and related computing infrastructure.

This paper introduces Diamond, and describes the Control System. The interface from the Control System to the accelerator technical systems, with specific focus on the diagnostics, the timing and the control of magnets is presented in more detail. Whilst the Control System extends across the photon beamlines and the associated experimental stations its use in these areas is not discussed.

4.7.2 Diamond Description

The Diamond accelerator complex consists of a 100 MeV electron Linac, a 3 GeV Booster synchrotron and a 3 GeV storage ring [1]. The Linac comprises a thermionic gun, a pre-buncher, a buncher and two S-Band accelerating structures. It provides a multi-bunch beam, up to 200 ns long with a typical charge of 1.7 nC, or a single-bunch beam with charge up to 1nC. The Booster synchrotron is a 5 Hz cycling, conventional missing-dipole 22-cell FODO lattice. This provides zero-dispersion straight sections which are utilised for injection, extraction and RF cavities. The Storage Ring (SR) is a 561 m circumference, 24-cell double bend achromatic lattice with a low emittance of 2.7 nm rad, obtained with finite dispersion in the straight sections. The SR currently uses two superconducting RF cavities for acceleration of a stored beam up to 300 mA in 800 bunch mode.

The photon output is optimised for high brightness up to 20 keV from undulators and high flux up to 100 keV from multipole wigglers. The facility currently provides twelve photon beamlines, with a further ten beamlines at various stages of conceptual design, detailed design and construction. The latter are all scheduled for operation by 2011. A further development of beamlines is now approved which will deliver an additional ten beamlines during the period 2008 to 2016. An image of the synchrotron building and office complex is shown in Figure 1.



Figure 1: The Diamond Light Source facility.

4.7.3 Control System Overview

The Diamond Control System is based on the EPICS [2] control system toolkit. EPICS is structured as a client-server architecture. The servers are called input/output controllers (IOCs) and provide the interface to the underlying technical systems. They are (usually) realised as embedded systems. The clients include the Operator Interface (Consoles) and other IOCs. The network protocol Channel Access is used to communicate between the Clients and Servers. The basic element of data over Channel Access is the Process Variable.

The process database, which resides on the IOC, defines the EPICS records which are the interface to the underlying hardware, as well as the processing to be performed by these records to map the raw data into engineering units, generate monitors, alarms and to interconnect to other records. Device Support then maps the functionality of underlying hardware on to these EPICS records, whilst Driver Support defines the interface to the hardware.

On Diamond the Input Output Controllers (IOCs) are ~260 VME systems running VxWorks, together with 204 Libera eBPMs running Linux on ARM processors and a further 10 soft IOCs running Linux on PCs. The IOCs are structured by technical system and by geographical location. Linux PCs are used as operator consoles.

4.7.4 Hardware

The interface from the Control System to the technical system is predominantly based on VME Industry Pack (IP) carriers, IP modules, transition cards and plant interface modules. The system is very modular and allows a high flexibility and density of I/O, thereby enabling most systems to be interfaced using up to four VME modules in a seven-slot VME crate. The IP module types include 16-bit ADCs and DACs, digital I/O, serial comms. (RS232, and RS422/485), a high resolution ADC (including current source for direct PT100 and thermocouple connection), a scalar and an incremental encoder. Motion control is realised using dedicated VME motion controller modules.

For interlocking and protection of equipment Programmable Logic Controllers (PLCs) are used [3]. Low-end requirements are addressed using Omron CJ1 [4] PLCs, which provide interlocking and control functionality. They are used for interlocking and control of vacuum valves and for slow machine protection by means of interlocks on water flows and equipment temperatures. High-end process control applications use the Siemens S7 [5] series of PLCs; such applications include the Linac, the SR superconducting RF cavity cryogenics and the cryogenic refrigeration plant.

4.7.5 Physical Structure

The Control System interfaces to the technical systems at 37 rooms, called control and instrumentation areas (CIAs). For the SR there is one CIA per cell, thereby ensuring that each cell is self-contained. The CIAs are air-conditioned to maintain a clean and temperature-stabilised environment for the instrumentation. The temperature stabilisation of the instrumentation helps prevent environmental disturbance from affecting the beam.

4.7.6 Network

A fibre optic infrastructure connects each of the CIAs to the Control System Computer Room and from there to the Control Room. It provides two computer networks, a Primary Network, dedicated to the Control System and a Secondary Network. For security and to enable effective management of traffic, the Primary Network is a private network with no routing to other networks. Control System information is accessed from the other networks through shared servers and application-layer gateways. Each network uses a central switch in the Control System Computer Room and a further layer consisting of edge switches at each CIA. From the central switches to the edge switches Gbit Ethernet is used, whilst from the edge switches to IOCs, servers or consoles 100Mbit Ethernet is used.

The Secondary Network is used for a range of miscellaneous computer systems which require network access around the accelerators, but which are not critical in terms of the operation of the facility. Examples where the Secondary Network is used include the streaming of video images to the control room, and the provision of terminal servers which are connected to the console port of all IOCs, providing for remote configuration and reboot even when the Primary Network is unavailable.

The fibre infrastructure is also used for timing signal distribution, for the machine protection system, and for the data communications for the beam position feedback system.

4.7.7 Servers and Consoles

All Servers and Consoles run the RedHat Linux operating system, currently Enterprise Linux 4, which is the standard Linux distribution used for operations and development.

The servers are all 1U rack mounted PCs, located in the Control System Computer Room. They provide the standard network services, NFS mounts, DNS, Print, Web, and Kick-start dedicated to the Control System Primary Network. There are also dual-homed servers including an SSH Bastion to allow remote access to the Primary

Network during shutdowns and emergency remote access during operations to fix faults, a number of EPICS Channel Access Gateways, an Archiver, and a Boot/Build server.

Operational IOCs are only able to boot from the Boot/Build Server. The use of a Build Server ensures that all target systems are built against a consistent build environment.

The EPICS Channel Access Gateways enable machine parameters to be read from the isolated Primary Network by the office networks and by each of the beamline networks. These application-layer gateways, with no direct routing of IP packets, are unidirectional (read-only) for office networks and bidirectional for the beamline network. An application of the latter is the control of ID gaps from beamlines.

There is also a computation cluster to support off-line physics modelling and simulations.

Operator consoles are provided in the Control Room, each of the CIAs and as laptops for mobile working (mainly commissioning). All run the standard Linux distribution. There are 10 Consoles in the Control Room, Figure 2, which provide dual 22" monitors and are dual-homed to both Primary and Secondary Networks, the latter of which provides access to other site networks. Laptops and CIA Consoles are standard PCs running the standard Linux distribution, which connect only to the Primary Network.



Figure 2: Diamond Control Room operator console.

4.7.8 General Applications

The main application requirements are addressed through the EPICS applications, EDM for synoptics, BURT for the backup and restore of process variables, ALH for alarm handling and Channel Archiver for historical archiving/viewing of data. Where bespoke applications [6] requiring client-side processing have been required these have been developed using the Python [7] language and the Qt [8] toolkit.

Synoptics for each technical system have been realised using the EDM display manager. This necessitated deployment of several new EDM widgets, including a Thumb Wheel control and a Video Display widget. Whilst EDM panels can be created directly for each application, they are generally built as templates and usually built using scripts from configuration data. For each IOC software module, e.g. a power supply controller, an EDM template panel is generated for all the detailed functionality. For the overall application, consisting of many power supplies a summary view is produced which allows the template panels to be called with appropriate process variable substitutions. Furthermore, the overall summary views are generated using

scripting tools, based on the number of required instances and the functionality required for each instance.

Whilst the EPICS tool BURT provides the required functionality to back up and restore signals in groups of PVs, its user interface was limited and it was recognised that the ability to group backup request files, typically by accelerator or by subsystem, and to back up and restore entire groups would be advantageous. This was realised by encapsulating the BURT within a Python/Qt application. Other examples of bespoke applications in Python and Qt include an application to control Top-up operation and some synoptics.

4.7.9 Physics Applications

To enable early testing of physics tools through the Control System, a virtual accelerator was implemented to give simulation of the lattice though the intended process variable interface. This was developed by providing EPICS device support to interface to a model using the TRACY II libraries [9].

For physics tools extensive use is made of the suite of Matlab tools for the design of accelerator physics applications, based on the Accelerator Toolbox (AT) [10], and MiddleLayer. MiddleLayer defines the abstraction of EPICS PVs to generic terms used in the AT. AT then provides a suite of physics tools for commissioning, characterisation and operation of accelerators. Figure 3 (left) shows a snapshot of the AT ‘Orbitgui’ application used at Diamond for orbit display, correction and to set orbit bumps. Most of the AT applications are used for both SR and Booster commissioning and operations, by redefining the Middle Layer for the appropriate accelerator.

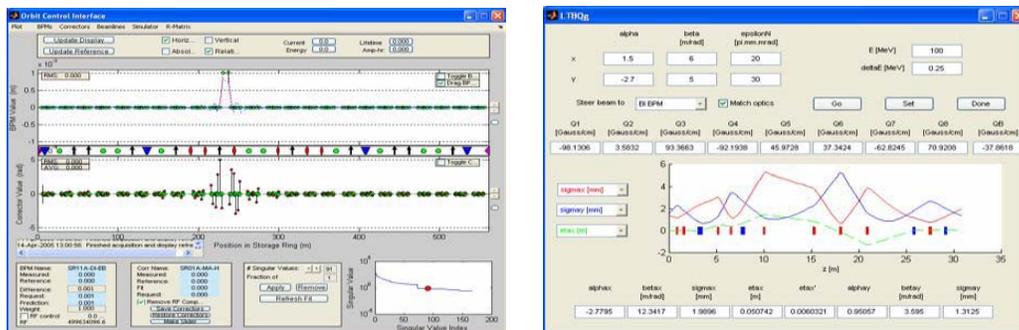


Figure 3: AT ‘Orbitgui’ application for Diamond SR (left); Matlab application to control the beam optics in a transfer line (right)

Development of Matlab physics applications was also undertaken in areas not covered by MiddleLayer, such as beam control in the transfer lines, Linac emittance and energy spread. Figure 3 (right) shows a snapshot of a Matlab graphical interface for control of the beam optics in the transfer line. Another area where Matlab applications have been developed is the analysis of turn-by-turn data [11] from the eBPMs in the SR and Booster. The eBPM turn-by-turn data mode enables experimental investigation of the non-linear beam using the SUSSIX [12] codes by post-processing of the turn-by-turn data, permitting the extraction of information on the detuning with amplitude, the frequency maps and the spectral content of the betatron oscillations.

4.7.10 Technical Systems

4.7.10.1 Magnets and Power Supplies.

The SR lattice consists of 48 main dipoles, 240 quadrupoles, 168 sextupoles, 336 horizontal and vertical steerers, and 168 skew quadrupoles. There are five pulsed magnets for injection and integral field correctors and trims for permanent magnet insertion devices, and one superconducting multipole wiggler. The main dipoles are series-connected with one power supply whilst all other magnets are individually powered. With the exception of the superconducting multipole wiggler all magnets are energised by power supplies [13] that use the same power supply controller [14]. This controller uses a Digital Signal Processor for the regulation and state machine, and an FPGA to generate the PWM for direct control of inverter modules. The inverter modules are phase-shifted to minimise ripple, and for medium and large supplies are in an N+1 redundant configuration enabling uninterrupted operation in the event of one module failing. The resulting performance is an 8 hour stability of better than 10 ppm and precision of 3ppm.

As the same controller is used across the nearly all PSUs, it provides a standard physical interface to the Control Systems with some functional differences. Each controller provides some 200 control and monitoring process variables. These include the basic control functionality of a power supply (On, Off, Set Current, Get Current) but extend to include hysteresis cycling, measurement of load impedance, measure of load temperature, version number, error states, synchronized ramping (used on the Booster) and waveform capture.

The interface from the power supply controllers to the IOCs is a 5 MHz Manchester-encoded point-to-point serial interface over a fibre optic link to a VME IP carrier and IP module. There are separate IOCs for each cell of the SR for quadrupole families and sextupole families, for steerers and skew quadrupoles, and for injection elements. Power supply controllers for IDs are interfaced by the IOC for that ID.

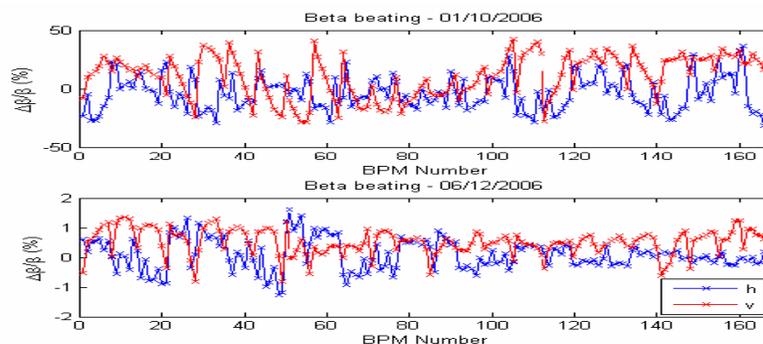


Figure 4: Initial beta beat (top) and beta beat using LOCO correction (bottom)

Having individual control of the field in each focusing element has enabled the use of Beam-based Alignment (BBA) and correction of the linear optics. BBA is used to establish the centre of each quadrupole to the eBPMs, such that a correct orbit then passes through the centre of the quadrupole, thereby preserving the optics of the lattice. BBA is realized by varying each individual quadrupole magnet and observing the effects on the orbit, from which the eBPM offsets are determined to an accuracy of

better than a few microns. The application LOCO [15] is then used to correct the SR linear optics. This uses a process of comparing the orbit response matrix and dispersion measurements made on the machine to those predicted using the model, and then performing an SVD fit to determine how to adjust the model quadrupole strengths in order to give the best reproduction of the measured data, and thereby to determine the corrections to be applied to the actual quadrupoles. The application of this on Diamond reduced the initial beta beat from more than $\pm 20\%$ to $\pm 1\%$, as shown in Figure 4.

4.7.10.2 Beam Position Diagnostics

Diagnostic instrumentation on Diamond consists of the Electron Beam Position Monitors (eBPMs), cameras and screens on synchrotron light monitors, Photon Beam Position monitors (pBPMs), and Beam Loss Monitors [16].

The eBPMs are the largest group of diagnostic instruments, consisting of 204 monitors, each of which uses a Libera Electron Beam Position Processor [17]. These provide for signal processing and detection in a FPGA with detected values being available through an embedded processor board. The processor board uses an ARM microprocessor with the Linux OS. To integrate this data into the Control System, Diamond developed an EPICS interface to run on the embedded processor board, thereby dispensing with the need for a VME interface. In effect each Libera eBPM becomes an IOC, or network attached device. This together with three timing signals provides for detailed control of the operation of the Libera eBPMs and synchronisation with the single pass beam in transfer lines and first turn modes or with the revolution frequency of the SR and Booster.

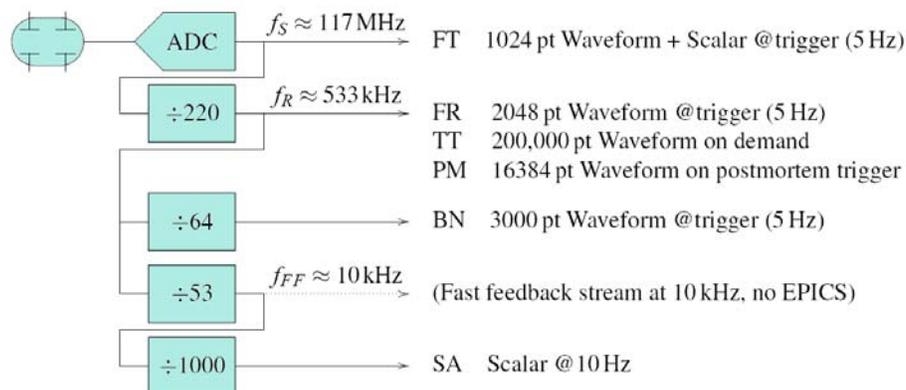


Figure 5: Libera eBPM processing chain

The data processing chain of the Libera provides digitised data at a wide range of rates, see Figure 5. These are created by filtering and decimation of the 4 ADCs at the input where the signals are sampled at 117 MHz. Data at this rate is available through a triggered 1024-sample buffer, which allows precise gating in time. This is used to measure transient beams, as in the transfer paths, or to measure individual turns in the Booster or Storage Ring. However, the ‘raw’ ADC samples have to be processed as the 500 MHz input has been transformed to a 31 MHz intermediate frequency signal by the under-sampling at 117 MHz. From this, envelope waveforms of 256 points of 34 ns

spacing are computed as well as an adjustable window with programmable width to integrate over the length of the bunch train in order to retrieve the beam position.

The First Turn (FT) process variable provides the calculated value of the beam position and intensity as well as the ADC waveforms and envelopes for every trigger. The overall trigger delay for the eBPM system, from the event receivers in the timing system, is used to measure precisely the orbit of virtually any turn. This display of individual orbits is particularly useful in establishing the closure of the injection bump in the SR, to observe not only the first turn (after passing through two of four kickers on injection) but also the second turn (passing through all four kickers during their decay) or even the turn before the first turn (of the stored beam, passing through all four kickers during switch-on).

A number of further process variables draw their data from a Libera internal circular buffer of button intensities at turn-by-turn rate. This data is continuously created by digital down conversion and decimation and stored into the buffer which holds several seconds of data. Whilst it would have been desirable to make the whole contents of this buffer available through EPICS, limitations of the available memory for buffering and of the available network bandwidth required a more structured approach.

For Free Running (FR) data, a vector of 2048 values with the beam position and intensity, updated on every trigger, is provided. This information is useful to display the early evolution of the beam after injection and real-time updated tune spectra by applying an FFT.

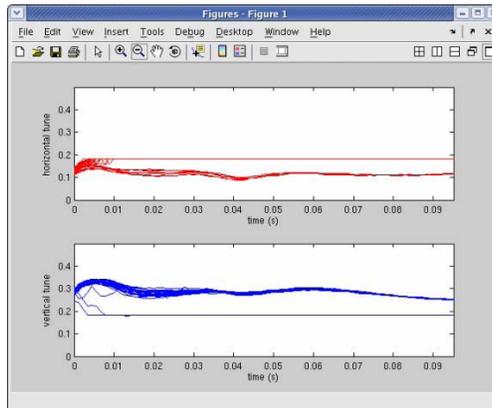


Figure 6: Fractional tunes during Booster ramp in Horizontal plane (top) and vertical plane (bottom). The 0.18 line was an artifact of the signal processing in the Libera eBPMs at that time.

For Turn-by-Turn (TT) data a vector holding at least 200,000 points is available. It is a triggered data source and needs to be enabled externally for a single shot acquisition. Transfer of this long buffer is realised using blocks of 32,768 points to reduce the amount of memory consumed by this function. This buffer is particularly interesting for accelerator physics applications [18], one example of its use being an application to extract the tune signals along the Booster ramp. For the Booster ramp 190,000 points of turn-by-turn data corresponds to the entire 100msec Booster ramp. Once the acquisition is armed, the data is acquired always from the same starting point with respect to the Linac gun trigger, e.g. in the Booster, at the beginning of the ramp. For each slice of turn-by-turn data, the highest peak in the amplitude spectrum is

detected. From this a precise determination of the betatron tune is then performed using an interpolated FFT algorithm. An example plot of the Booster tunes during the ramp is shown in Figure 6.

For Post Mortem (PM) purposes, the generation of data for the past 16384 turns (pre-trigger) is triggered by a dedicated PM input to the Liberas on beam loss and used to review the events leading up to the beam loss. The PM trigger is produced from the Machine Protection System beam loss event and distributed through the timing system. (See section Machine Protection System)

For Beam Normal (BN) a further decimated waveform is returned, containing the beam position and intensity averaged over 64 turns. The length of the vector covers 100ms and is continuously triggered. For display purposes these waveforms are also available with an additional decimation of 16 thus giving the average of 1024 turns. An application of this data is the Booster closed orbit display as shown in Figure 7.

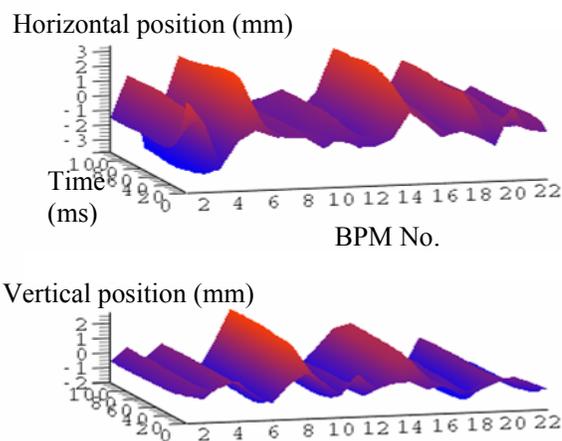


Figure 7: Booster orbit during acceleration cycle, with horizontal position (top) vertical position (bottom). Position, on vertical axis, shown for all 22 eBPMs against time.

Finally, the Slow Acquisition (SA) data gives the stored beam position updated at 10 Hz as scalar process variables.

A further data source is available at a 10 kHz update rate, but not through the processor interface and hence not directly as EPICS process variables from each Libera. This data stream is directly connected to the Fast Orbit Feedback (FOFB) network through a custom communication controller implemented on the FPGA. From a data access node on this network the data is then available as EPICS process variables (See section Orbit Position Feedback).

Throughout the injection chain, screens and synchrotron light monitors (SLM) are installed to image the electron beam [16]. These all use IEEE1394 cameras interfaced into the VME IOCs using a PMC IEEE1394 interface located on the processor boards. The cameras all update at 5 fps synchronised to the injection trigger. An EPICS driver for the cameras was developed to enable control of the exposure time, the camera gain, and digital zoom and pan. To facilitate the use of the images with the EDM display manager a Video Display widget was developed that includes the ability to add false colour corresponding to image intensity. Images are also taken directly into Matlab applications for processing. Examples of an EDM display showing the SR Synchrotron

Light Monitor and of a Matlab application for matching BTS optics are shown in Figure 8.

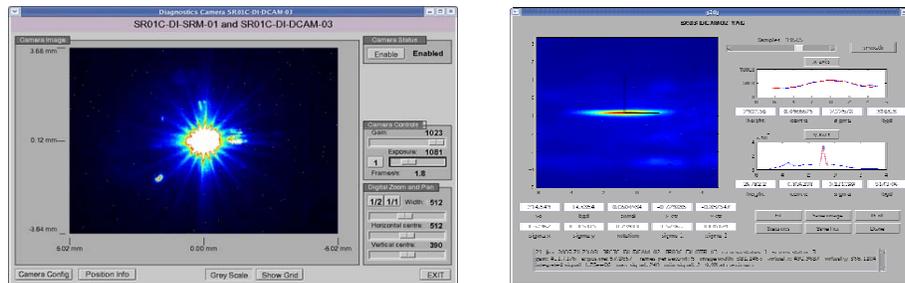


Figure 8: EDM panel and video widget showing Storage Ring Synchrotron Light Monitor (left) and a MatLab application for matching BTS optics (right).

The remaining diagnostic instruments include the Beam Loss Monitors (BLMs), pBPMs and positional measurements on all eBPM vessels. For the Linac and transfer paths the BLMs are PMT/scintillator monitors and are interfaced through ADCs triggered on injection. For the Booster and SR, PIN-diode BLMs are used which are interfaced through scalars. The positional encoders are interfaced through incremental encoder modules. In all cases these interfaces these are IP modules in VME crates.

4.7.10.3 Orbit Position Feedback

The Fast Orbit Feed-Back (FOFB) system [19] on Diamond performs a global orbit correction using the position from the 168 horizontal and 168 vertical eBPMs to 168 horizontal and 168 vertical corrector magnets at a 10kHz update rate. A custom communication controller, implemented in VHDL, is used to move X and Y positional and control data from the 168 Libera eBPMs to each of 24 computation nodes [20]. The network topology is structured as a 2D torus, with one computation node per cell, and gives a degree of independence from failure of single or combinations of eBPMs or links, as shown in Figure 9 (left). Each of the computation nodes receives data from all eBPMs, and uses a dedicated MVME5500 VME processor board, located in the Steerer IOC, to calculate the outputs for a sub pseudo inverse response matrix (Tikhonov regularisation is applied to scale the singular values [21]) corresponding to the seven correctors for that cell. The regulator, an Internal Model Controller [22], is then implemented as fifth order IIR filters on the outputs of the sub pseudo inverse response matrix followed by a series of boundary checks to trap for erroneous conditions and to shutdown the feedback process in a graceful way in the event of unrealistic corrector demands. The new PSU demand values are written over the same point-to-point links that are used to connect the IOCs to the PSU controllers for that cell. A secondary loop performs correction of the residual horizontal orbit by adjusting the RF frequency.

The FOFB provides around 20 dB of suppression at 16 and 24 Hz where most ground noise is coupled to the girders, with a crossover point at 80 Hz. It is unable to correct beam motion near 300 Hz caused by the girder cooling water flow and mechanical resonances, Figure 9 (right). It also prevents insertion device gap changes from affecting other beam users by compensating for the beam motion remaining despite the use of ID gap feed-forward tables.

Also located on the FOFB network is an additional node for data capture. This captures the 10kHz positional data synchronously sampled from all eBPMs with sizes

up to 144MSamples. An application of this is calculating the beam position power spectral density (PSD) using Welch's method of overlapped windowed FFTs. These PSD measurements are repeated with and without feedback running to measure the system performance (the sensitivity function) and also to correlate with seismometer and X-ray beam position measurements to identify the sources of beam vibration in the machine.

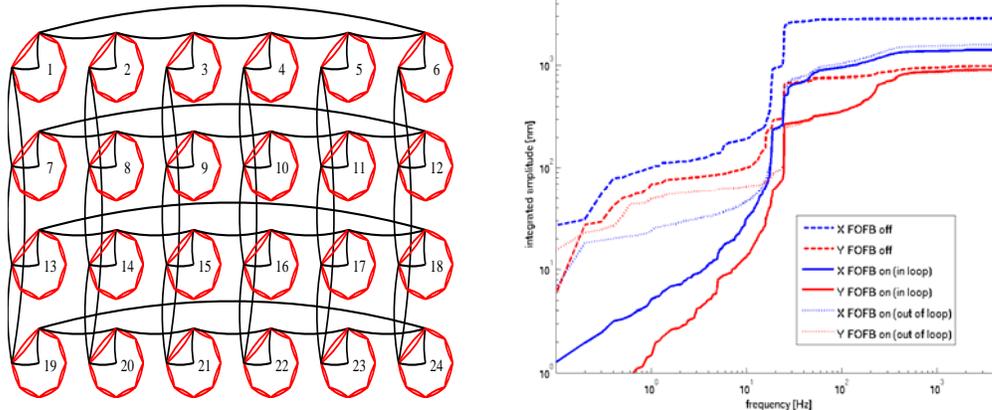


Figure 9: FOFB communication network topology, with each circle being one SR cell consisting of 7 eBPMs, and a computation node in the centre. The interconnection is as a 2D torus (left). Integrated amplitude of positional noise for Feedback Off, Feedback On, in loop, and Feedback On, out of loop (right).

4.7.10.4 Timing System

The Diamond timing system [23] is based on a master event generator (EVG) and multiple event receivers (EVRs) [24] which are installed in the IOCs.

The EVG generates events from an internal sequencer and external signals. The events are distributed over fibre optic network to multiple event receivers (EVRs). The network is structured as a two-level multi-star topology, with simultaneous delivery of event messages to multiple EVRs by using fibres of equal lengths, to a precision of ± 0.15 metre or < 1 nsec. The EVRs connect directly to the equipment using copper or fibre optic connections or through a four-channel timer when greater resolution is required.

The EVG issues event frames, each consisting of an 8-bit event code and an 8-bit distributed data bus, at a rate of 125 MEvent/sec. The event clock is derived from the 500 MHz RF signal. The events can be generated from eight external trigger events, a sequence RAM, software events or events received from an upstream event generator. The latter are not used on Diamond.

The EVR recovers the 500 MHz clock from the event stream and demultiplexes the event stream to the 8-bit distributed bus and the 8-bit event code. The decoded events are then mapped through RAMs on to one of the following: four delayed pulse outputs, with programmable width and delay, 14 pulse outputs with programmable delay and width or seven set/reset flip-flops. The processed events can produce hardware outputs, software interrupts or EPICS software events.

Currently eighteen events are produced. These define the triggers for the acceleration process, so the events include Linac Trigger, Booster Magnet Cycle, Booster Injection, Booster Extraction, and SR Injection. There are also early (or Pre) versions of some of these events for charging power supplies and diagnostics versions for instrumentation triggers. Other events include control of fast orbit feedback, signalling top-up cycles, and resetting the integration period on Radiation Monitors. The 8-bit distributed bus is used for transmitting a 1 MHz reference, a post-mortem signal, and time stamping data.

The Diamond SR has 936 buckets and a revolution frequency of 533.8 kHz, and the Booster 264 buckets and a revolution frequency of 1892.6 kHz. The coincidence of the SR and Booster revolution determines a frequency for the coincidence clock of 48.529 kHz. The Booster clock is used to increment the EVG sequence RAM, and the coincidence clock is used to reset the sequence, thereby locking all cycles to both accelerators. Event entries in the sequence RAM then generate the necessary sequence of triggers to accelerate the electrons through the Linac and Booster and into the SR.

The operation of modern light sources requires that a variety of fill patterns are realised in the SR. Using the system above it is possible to place a single bunch or bunch train anywhere in the SR by delaying the extraction from the Booster. However this can result in an unacceptably long delay. An improved scheme is to consider that the SR orbit is divided into 8 segments of 120 buckets and every fourth Booster turn is phase-locked with its SR segment. Hence, it is possible to place a single bunch or the start of a bunch train in a given SR bucket by varying the Linac gun trigger (in a range of 120 RF clock periods) and Booster extraction event with multiples of four Booster orbit clocks. So, for example, to target SR bucket number 340 the Booster extract delay will be 8 periods of Booster orbit clocks and the Linac delay will be 100 RF clocks. This sequence of modifying the Linac trigger times and Booster extraction on a cycle-by-cycle basis is controlled through an EPICS application running on the IOC. Control System inputs to this application determine the fill pattern for the SR. Currently pre-prepared patterns are single bunch repeat, bunch train repeat, single bunch one shot, bunch train one shot, SR 2/3 fill and SR fill of all buckets. It is also possible to download an arbitrary list of SR buckets to fill to the EPICS IOC application. The application then puts the charge into those buckets on subsequent cycles. By comparing the measured SR charge against a desired charge profile this process is then used to maintain charge in the required SR buckets at the desired level and forms the basis for top-up control [25].

4.7.10.5 Machine Protection System

The protection of equipment is realised through the Machine Protection System (MPS), which provides fast, sub 600 μ sec, beam dumps by switching off the RF when a critical interlock fails. These critical interlocks include the 672 invalid-orbit interlocks calculated in the Libera eBPMs and other critical interlocks where components are exposed to high beam powers, up to 30 kW (for a photon beam generated by a superconducting wiggler with a 300 mA electron beam). A further 456 slow interlocks from water flows and temperatures are processed through PLC sub systems and feed into the MPS on a cell-by-cell basis. For each SR cell a local MPS module generates a pulse stream [26] to the global MPS module which, subject to all local modules being good, generates a pulse stream to the source of the energy, RF amplifiers or PSUs. The local and global MPS modules are realised as VME64x transition cards and are

monitored by a VME DIO board. The VME IO board and EPICS software are not part of the protection process, but serve to manage the process. The PLCs are supervised over a serial link.

The MPS beam dump signal is used to produce a post-mortem event, which is distributed through the timing system to each of the Libera eBPMs and other IOCs when beam is lost. This stops logging in the Libera buffers and triggers the buffer to be read out. A beam dump on invalid orbit, exceeding ± 1 mm limits, is shown in Figure 10.

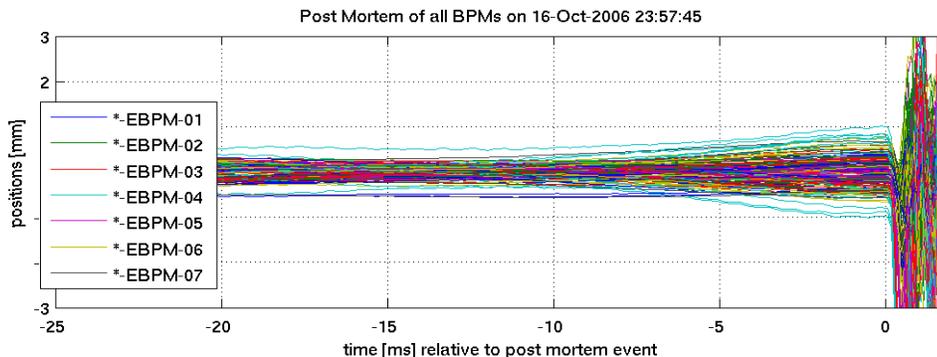


Figure 10: Post Mortem dump of all horizontal eBPMs showing beam loss at time 0 ms due to the beam position exceeding the ± 1 mm orbit limit.

4.7.10.6 Linac, RF, Vacuum and Other Systems

The control of the modulators for the Linac RF is performed by a number of Siemens S7 PLCs. These interface to the IOC over a network connection. All other hardware associated with the Linac, including PSUs, Vacuum, Timing etc, is managed using the standard interfaces used elsewhere on Diamond, thereby maintaining consistency.

The Booster RF amplifier is a commercial unit using a commercial proprietary amplifier controller and interfacing into the IOC over a serial link. The Booster RF and low level RF use analogue signal processing and are controlled using DACs, ADCs and digital IO. Control of each of the SR amplifiers is performed by an associated IOC, which interfaces to the amplifier through standard VME IO.

All vacuum gauge controllers and ion pump supplies are interfaced to IOCs via serial connections in a point-to-point configuration. Vacuum valves are controlled and protected by standard valve control units each encapsulating a PLC and interfacing to the IOC using a serial connection. RGA units, which are capable of being operated in a stand-alone or network mode, are also interfaced to the Control System, to allow a number of predefined mass scans to be viewed and archived through the Control System.

The 72 girders which make up the SR each have 5 degrees of freedom driven by motorised cams which are controlled by OMS VME58 controllers. Other hardware associated with the girder positions is managed through the standard VME IO.

Each of the first nine permanent magnet IDs is also controlled by OMS VME58 motion controllers, together with a PLC subsystem to define a safe operating window

and hence protection of the structure. Other instrumentation associated with the IDs is managed through the standard solutions.

The Personnel Safety System (PSS) [27] on Diamond is based on a hard-wired dual guard line, relay based system. Whilst the protection logic is entirely realised through the hard-wired relay logic, the inputs, intermediate states and all outputs are monitored by the Control System.

4.7.11 Conclusions

The Control System for Diamond has met its requirement in terms of technical performance, and importantly, initial availability and subsequent stability. Even with some level of overlap of commissioning between the Control System and systems being controlled it was there as a service to support commissioning of the accelerators and beamlines.

The use of modular design, re-use of code and implementation by configuration, made for a robust system design, and efficient subsequent management. The use of the vertical tests and simulation early in the design process gave high levels of confidence in the final application software, all of which has benefited from the stability and reliability of the underlying EPICS toolkit.

4.7.12 Acknowledgements

The realisation of the Diamond Control System was the work of many people, including those in the Diamond Control Systems Group, other groups in Diamond, the EPICS collaboration and other laboratories and projects who have so willingly assisted us.

I am further grateful to Dr Günther Rehm for contributions to the Diagnostics sections of this paper.

4.7.13 References

1. R. P. Walker, "Overview of the Status of the Diamond Project", EPAC 2006, Edinburgh
2. www.aps.aml.gov/epics
3. S. C. Lay, P. Amos, P. Hamadyk, M. T. Heron, H. Shiers, "The Selection, Development and Application of PLC Solutions for the Diamond Light Source", ICALEPCS 2007, Knoxville
4. <http://www.omron-industrial.com/uk/home/products/automationsystems/ProgrammableLog/ModularPLCSeri/CJ1M/default.asp>
5. <http://www.automation.siemens.co.uk/main/extra/literature/files/Automation%20Brochures/Safety%20Systems/6ZB5310-0FW02-0BA6%20-%20Hotstandby%20PLC.pdf>
6. T. M. Cobb, M. G. Abbott, I. J. Gillingham, N. P. Rees, M. T. Heron, "Diverse Uses of Python on Diamond", PCAPAC 2008, Ljubljana
7. <http://www.python.org/>
8. <http://trolltech.com/products>
9. M. Boge, "Update on TRACY-2 Documentation", SLS Internal Note, SLS-TME-TA-1999-0002 (1999); <http://slsbd.psi.ch/pub/slsnotes/>
10. A. Terebilo, "Accelerator Modelling with MATLAB Accelerator Toolbox," PAC'01, May 2002, pg. 3203.
11. R. Bartolini, I. Martin, G. Rehm, J. Rowland, "Turn by Turn Data Analysis at the

- Diamond Storage Ring”, EPAC 2008, Genoa
12. R. Bartolini and F. Schmidt, CERN-SL Note 98-017 (AP), (1998).
 13. R.J. Rushton, C. A. Abraham, J. A. Dobbing, F Cagnolati, G. Facchini, M. Pretelli, V. Rossi, L. Sita, C. Rossi, “Diamond Storage Ring Power Converters”, EPAC 2006, Edinburgh
 14. <http://arxiv.org/pdf/cs.SE/0111019>
 15. J. Safranek, NIM A388, 27, (1997).
 16. G. Rehm, M. Abbott, “ Performance of Global Diagnostics Systems During the Commissioning of Diamond”, EPAC 2006, Edinburgh
 17. <http://www.i-tech.si/>
 18. R. Bartolini, M. Abbott, I. P. S. Martin, G. Rehm, J. Rowland, ”Turn-by-Turn Data Acquisition and Post Processing for the Diamond Booster and Storage Ring”, EPAC 2006, Edinburgh
 19. M. G. Abbott, J. A. Dobbing, M. T. Heron, G. Rehm, J. Rowland, I. S. Uzun, “Performance and Future Development of The Diamond Fast orbit Feedback Systems”, EPAC 2008, Genoa
 20. I. S. Uzun, R. Bartolini, G. Rehm, J. A. Dobbing, M. T. Heron, J. Rowland, “Initial Design of the Fast Orbit Feedback System for Diamond Light Source”, Proceedings of ICALEPCS 2005.
 21. A. Neumaier, “Solving Ill-Conditioned and Singular Linear Systems: A Tutorial on Regularization”, SIAM Review, 40, 636-666, 1998
 22. W. P. Heath, A. G. Wills, “Design of Cross-Directional Controllers with Optimal Steady State Performance”, European Journal of Control, 2004.
 23. Y. Chernousko, A. Gonias, M. T. Heron, T. Korhonen, E. Pietarinen, J. Pietarinen, “The Timing System for Diamond Light Source”, EPAC 2006, Edinburgh
 24. <http://www.mrf.fi/>
 25. R. P. Walker, R. Bartolini, P. Bonner, F. Burge, Y. Chernousko, C. Christou, J. A. Dobbing, M. T. Heron, V. C. Kempson, I. Martin, G. Rehm, R. Rushton, S. J. Singleton, M. C. Wilson, Diamond Light Source, “Preparation for Top-up Operation at Diamond”, EPAC 2008, Genoa
 26. http://www.jlab.org/accel/ssg/papers_Pres/technotes/TN-0193.pdf
 27. M. C. Wilson, K. A. R. Baker, E. V. Buckley, M. T. Heron, J. R. Alexander, “Diamond Personnel Safety System”, ICALEPCS 2003, Gyeongju

4.8 The Accelerator Control System at DESY

Agababyan, T. Bruns, P. Gessler, G. Grygiel, O. Hensler, R. Kammering, H. Keller, V. Kocharyan, A. Petrosyan, G. Petrosyan, L. Petrosyan, V. Petrosyan, K. Rehlich, V. Rybnikov, E. Sombrowski, M. Staack and T. Wilksen
 Deutsches Elektronen-Synchrotron, Germany
 Mail to: kay.rehlich@desy.de

4.8.1 Introduction

The Free-Electron Laser in Hamburg, FLASH, located at DESY, is the first of its kind worldwide to produce femtosecond pulses of soft X-rays [1]. In 2013, the planned European XFEL facility [2] will deliver hard X-ray pulses far shorter than those from any other X-ray source and with a peak brilliance of six to eight orders of magnitude higher. The unprecedented shortness and intensity of these X-ray pulses as well as their coherence open entirely new fields for photon science research.

4.8.2 Requirements for a FEL Control System

In contrast to the continuous and steady beam delivery of conventional synchrotron radiation sources, FLASH delivers up to 800 shots 5 times per second. Hence it needs a very precise synchronization between accelerator and experiments. Furthermore experiments at FLASH sample with very high frequencies – up to several GHz - to get a proper time resolution while investigating fast processes with photon beams. A huge number of high resolution and very fast acquiring digital cameras are included in the data collection at the experiments end stations. The camera acquisition requires to be precisely triggered with the machine pulses.

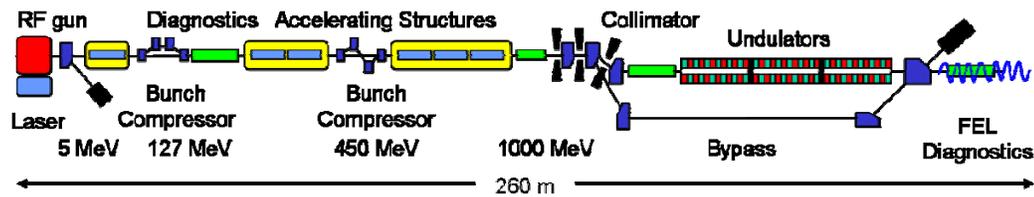


Figure 1: Schematic overview of FLASH

Even though FLASH has only a total length of 260 meters, the number of diagnostics components is comparable with that of much larger machines like third generation synchrotron radiation sources. The densely packed equipment and highly sophisticated diagnostics are needed to meet the high demands of controlling the SASE process (for details about the SASE process see [1]). Furthermore this puts strong requirements on the radio frequency (RF) control and regulation. E.g. a phase stability of at least 10^{-4} is needed to establish a proper peak current for the SASE process. Various different generations of low level RF (LLRF) control and regulation hardware for the superconducting cavities, utilizing DSP and modern FPGA technologies, have been successfully developed and deployed at FLASH. Therefore the FLASH facility serves as a perfect test bed for XFEL as well as for ILC research and development.

Besides the standard beam diagnostics like BPMs, wire scanners, toroids and optical transition radiation monitors (OTR), numerous experiments diagnostics e.g. gas monitor detectors (GMD), photon BPMs, optical replica synthesizers (ORS), a transverse deflecting cavity (LOLA) or a femtosecond synchronization system, have been implemented at the accelerator and the photon experiments end stations. Nearly all of these devices must be remotely controllable to allow e.g. adjusting mirrors, slits or attenuators. This leads to a huge number of actuator devices to be integrated into the control system.

Since the beginning of FLASH operations a new approach to record all machine relevant data in a central archive has been deployed and proven to be very useful for various purposes. Originally more focused on accelerator-based operations and research, this data acquisition system (DAQ) has attracted now more and more photon experiments users. Besides diagnostics information from the accelerator also the experiments data is recorded. This allows an easy correlation of machine shot data with the user data, especially for users who run their experiments only for short periods of time.

To accommodate the complicated and sophisticated tasks of slow feedback loops, monitoring of complex system variables etc., high level applications are integrated into

the DAQ system, since all synchronized machine data is available here. For example, complex algorithms to compute the beam energy - developed in MATLAB - can directly run within this level.

FLASH operates as a user facility since more than 3 years and has a strong collaborative character. Besides the challenging demands it has to meet for standard operations, being a test bed for ILC superconducting cavity technology puts extra requirements to the control system. Remote access to nearly all control system variables, also for external users, has been a must since the first years of FLASH operation. Tools like VNC, SSH and nowadays SUN secure global desktop, allow all members of the collaboration full access to the control system and therefore to nearly all components of the FLASH accelerator directly from their home institutes from all around the world.

4.8.3 Implementation in DOOCS

In 1993 a project was started to develop and deploy superconducting cavities at a test linac, the TESLA test facility (TTF), with the goal to design a large linear collider. For the control system it was decided right from the beginning to use state-of-the-art technologies. The programming language C++, utilizing an object-oriented approach, was just invented during the first days of the project. A distributed system, with weak dependencies between components using object-orientated implementations at all levels of the system, was envisioned. This was the birth of DOOCS - the Distributed Object-Oriented Control System [3]. The system has evolved and matured since then. FLASH has now about half a million values and parameters to be controlled by DOOCS. 30000 of these channels are archived as floating point or status variables.

4.8.3.1 *The DOOCS Architecture*

The front-end servers represent the lowest layer or tier in the control system. These servers are directly connected to the hardware by different kind of busses: VME, SEDAC (a DESY-developed field bus), CAN, PROFIBus, GPIB, Ethernet-based busses like MODBUS etc.). Simple input/output controllers as well as complex ones like PLCs are connected to the device servers within this layer.

The middle layer or service tier is used for global services: the equipment name server (ENS), services for automation (e.g. finite state machines), the DAQ, databases and web services. A design goal was to implement complex functions and automation in this tier and not in the client layer.

The client layer is used to present the data of the control system to the user. Client programs do not require complex logic or processing and do not hold states of the control system.

In general, the object oriented approach implements multiple devices of the same type in a device server process on the lowest tier. A certain device is represented as an instance of a device class of its type. All required abstractions of a device are implemented in a single piece of code. If another device is added to the system cloning during run-time can just generate it. The data of a device, that is accessible as a property of the device from the network, is implemented as data class in a server library. By this means devices and the network interface have a lot of common elements. This allows on one hand to change general functions by changing the server library instead of

modifying all individual device servers. On the other hand it is the base for generic programs to talk to all different device and data types.

A further design goal was to create a transparent system. A name server supports writing ad-hoc routines or finding data channels. This name service provides an online list of all devices and properties, including a short description of the function, on the network. Since data is transferred as data objects, a transformation of data types is handled internally. Archives of values are defined in device servers and therefore no further configuration work is required.

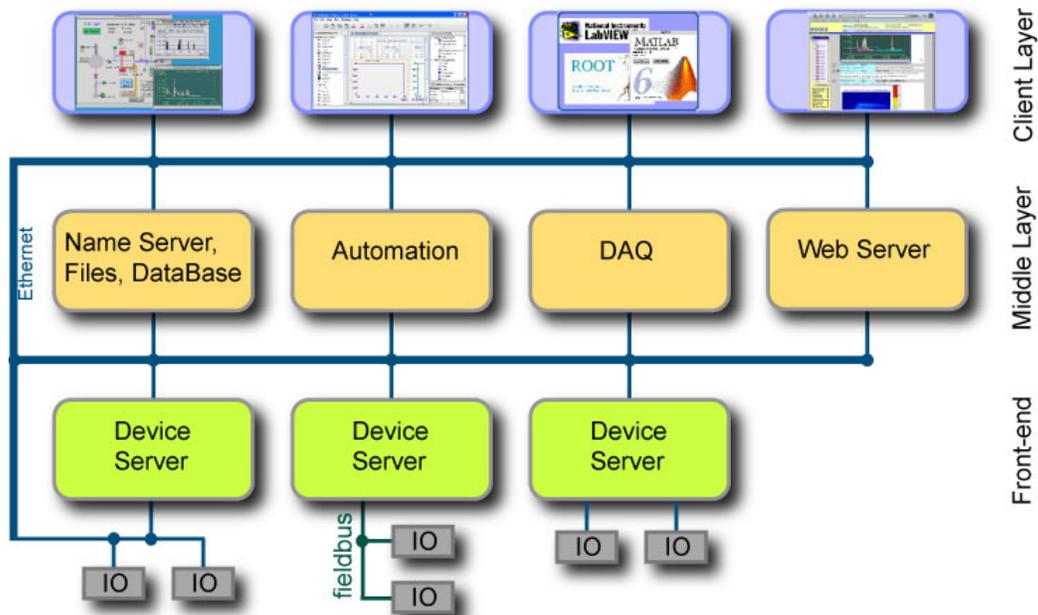


Figure 2: The DOOCS Architecture

With this device abstraction on the front-end layer the services for multiple devices, like automation, are located on the middle layer. This gives a clean hierarchical organization of the controls. Automation is implemented in separated processes as a finite state machine or is attached to the DAQ system. As explained later, the DAQ provides a fast data path to the front-ends. A measurement procedure attached to the DAQ can make use of the synchronized data flow of all beam diagnostics instruments.

Since all complex operations are implemented in the two lower tiers the programs on the client layer do not require providing significant processing of the data. Therefore most of the user interfaces can be created by the help of a graphical editor (ddd [7] or jddd [9]). In addition, applications to browse all data in the system, i.e. rpc_test and jDTool, are available. A Save & Restore tool allows to read, save and restore all simple parameters of the control system. For measurements and ad-hoc procedures an interface to MATLAB is provided.

4.8.4 The DOOCS Application Programming Interface

Applications access any signal known to DOOCS via a special software layer called application programming interface (API). An API defines rules, data types and functions used to deliver requested data. Every signal known to DOOCS is called a channel. A channel is identified by its unique address. An address consists of 4 components of the form Facility/Device/Location/Property. This name space can be represented as a 4-layer tree. The components of an address define the property name (Property) of the signal known to the unit (Location) attached to the server process (Device) running on some computer in the context of a physical machine (Facility) like FLASH or XFEL. The collection of all addresses is forming the DOOCS name space. Name query commands are available to read the actual entries of the three tree levels, and user interfaces display the full dynamic name space in a graphical way.

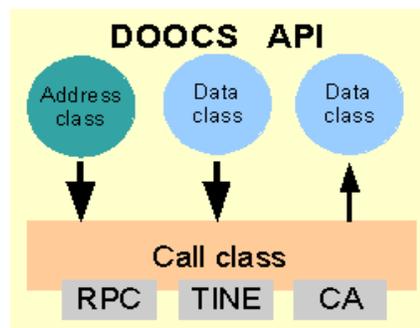


Figure 3: Class layout of the DOOCS API

An intrinsic data type represents the value of every channel, e.g. ADC signals are mostly floats and interlock signals are bits. There are also complex data types like archives of values, images and waveforms. The device server defines that data type. With the help of the data transport class an automatic conversion of data type can be performed on the client or the server side.

An application program interacts with the API by means of three basic class objects - address class, data class and call class.

An address class stores the channel address and provides some set of functions to handle different components of the address like changing property or location, merging address components and so on.

A data class is a container of the channel's value in the format corresponding to the type of a signal. Apart from that, this class has some extra miscellaneous functions that can extract an actual value or store a new one in any other format preferred by user.

A call class is the most important and critical part of the API as it delivers data to/from channels and deals with the low level protocols talking to server processes that host the requested signal. For each addressed channel it maintains a record in internal table. The typical sequence of actions is the following:

address resolution → data exchange → error processing

At the first stage it resolves the address of the channel supplied in the address class object. The resolution is carried out by means of contacting ENS (Equipment Name

Server). The first information to resolve is the protocol to talk. In case of the RPC protocol the requested server process has to be localized before this class enters the second stage. Now a new value supplied in data class object is sent to the channel or an actual value of the channel is received and stored in the data class object provided by the application. The API organizes the data exchange as either synchronous or asynchronous transfer. In the first case the application waits until the server delivers the response. In the second, it simply triggers the transaction. Apart from that, the call class handles all error situations and reports status, monitors the state of a connection link to the channel and tries to recover it after errors. All this activities are completely hidden from the user. This class has also some set of management functions used to tune the API for a specific channel. With the help of the protocol information from the name server the API provides also a path to other control domains. At present it recognizes address domains of control system software packages TINE [4] and EPICS [5] address domains allowing applications to directly access these control systems. The Java version of the API supports TANGO [6], too. Address and data class objects are always parameters for functions of call class objects.

The API is implemented in C++ as a UNIX shared library, called client library. A Java version is available as a jar file. To improve the efficiency of the library the concept of multithreading was used.

4.8.4.1 Object-Oriented Server Design

A DOOCS server is a UNIX process that consists of objects from a C++ server library. An actual server consists of several entries of some device type at different locations in the system. It may contain different device types also. Every instance of a device defines a set of properties. These named properties are the access points on the communication network and are implemented as data objects.

Some of the fundamental object oriented concepts, like encapsulation, inheritance and polymorphism, are used in the design of DOOCS device servers. Obviously, the inheritance concept is used for the device defining classes. The main device class contains a set of standard properties. All the device classes are derived from this base class. Polymorphism is used at creation of all the DOOCS data function classes to ensure the same programming interface to access different kinds of properties.

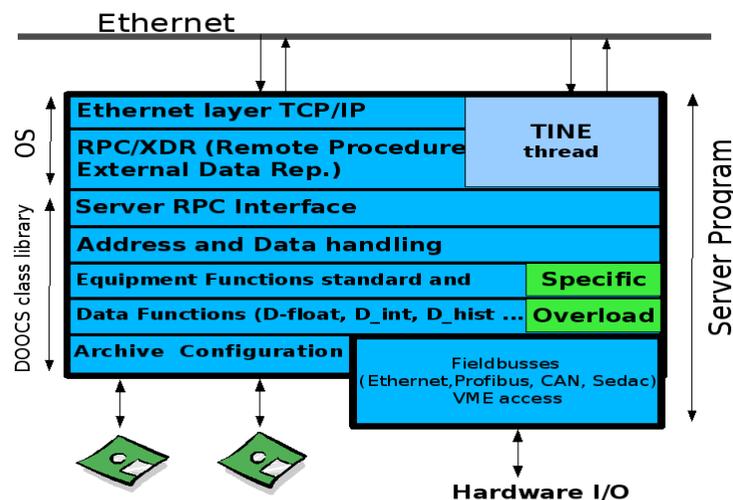


Figure 4: DOOCS server structure

To represent a device in a DOOCS server, two main objects were identified and specified in classes. The first class implements a device, the second one defines a data property of a device.

A server process has a container of device classes. Each objects of these classes represents one instance of a certain device type. E.g. the toroid server contains one instance of the toroid class for each toroid in the machine. The physical position of the device in the machine defines its location. So, the server keeps a list of devices at different locations.

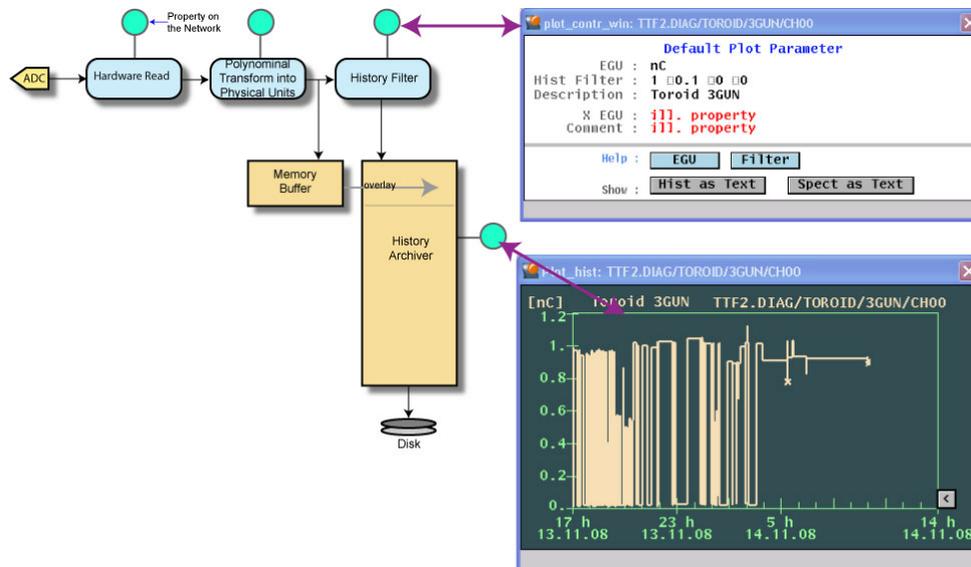


Figure 5: DOOCS Standard Archiving Utility

Each device server class is composed of a number of data property classes. Data classes represent the attributes of a device that are accessible on the network. All data objects inherit the `D_fct` base class. Standard data classes in the library are integers, floats and structures for filter settings, arrays and complex data types like historical trend data or spectra.

Actual servers overload the standard device class and just add the device specific data functions. A DOOCS device always contains some standard parts to handle the device errors and status information. This standard part is identical in all locations and sums up to around fifteen properties. Since a client application can query all device and property names during run time, it is possible to assign device names dynamically. A moving pump station has been implemented in this way for instance. It was a design goal to run a device server autonomously. Therefore it keeps its configuration locally to be independent of the network.

4.8.4.2 Archiving

To archive a value in DOOCS, one just has to add the `D_hist` data class to `D_float`, or use the combination of both, the `D_float_hist` class, which provides a ring-buffer-like storage of float or status data. These classes automatically add two special properties to control the history filtering and engineering units for the plot display. If the ring-buffer is full, the oldest data will be overwritten.

4.8.4.3 Server Location

Every server must have one standard location to control the server updates, read and write the configuration file, configure it and collect statistics about its runtime behavior. Browsing from the watchdog panels, one may reach the following standard server overview panel.

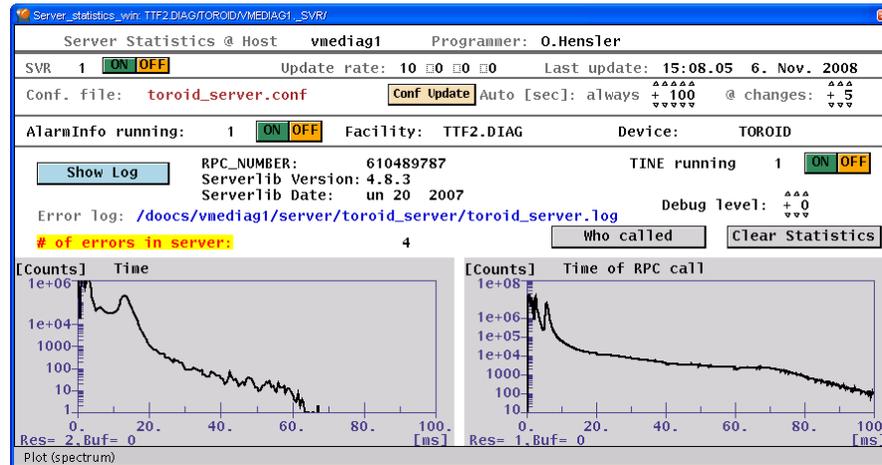


Figure 6: Property panel of a DOOCS server (Toroid)

4.8.4.4 Related Libraries

In addition to the device and network data classes, the server library contains a lot of helper classes, like archiving for historical trends, reading and writing of the configuration files, alarm server interface and error logs.

The server library is extended by many additional communication libraries derived from a `D_fct` class to access VME, Ethernet, and Modbus over Ethernet (Beckhoff), CANbus, PROFIBus or SEDAC devices. Another powerful class is the `D_ADCscope`, which facilitates a scope-like reading of a fast ADC channel with many configuration, calculation and archiving possibilities. Based on this class, many diagnostics servers like toroids, BPM or LLRF, are realized.

4.8.4.5 Toroid Example

The DOOCS server used for reading out the toroid signal gives a good overview of many features of the DOOCS server library and beyond. The timing system is used to synchronize the server with the repetition rate of the FLASH machine. The analog signal of a toroid is split to two fast ADC channels to sample the signal baseline and the peak. So, two instances of the `ADCscope` class are used to read the channels and build the difference to suppress the baseline variations. In order to see the complete bunch train, a standard `D_spectrum` type class is overloaded to calculate the charge. To get a history trend, the first bunch is always calculated and stored over time.

The charge of every bunch is important for middle layer measurement and feedback servers. Therefore the DAQ libraries are attached to the toroid server to send out the data of a whole macro pulse to the central DAQ server with the repetition rate of the accelerator.

4.8.5 Control System Hardware and Infrastructure

The FLASH control system runs on SunOS/Solaris, Debian GNU/Linux and Ubuntu operating systems on either SPARC or x86/64 hardware.

4.8.5.1 Front-End Hardware

On the front-end devices an embedded SPARC CPU system with VME interface running Solaris 8 is used. Housed in VME crates these are monitored and controlled via CANbus read-out. As front-ends for the camera system, 19" PCs are used and remotely controlled through an additional IPMI interface. There are mainly SUN SPARC servers operated as middle layer services for NFS support, backup, web services and user login. 19" PCs running Debian Linux is used as a simulation cluster.

4.8.5.2 Timing System

The purpose of the timing system is to trigger devices on certain events of the machine operation, to provide synchronized clocks for ADC sampling and to distribute further reliable information. Typically distributed devices like the gun, kickers, klystrons, data acquisition modules etc., need to be triggered with configurable delays from a common source. Some of them require in addition a further synchronization for ADC sampling. For the trigger a jitter of the order of a nanosecond is sufficient, ADC clocks might require a much better stability. Timing signals have to be stable and are therefore implemented in hardware. The parameters of the timing system are controlled and read out by the control system.

DESY upgraded a timing system originally developed by Fermilab. It is a flexible and distributed system based on a modulated 9 MHz pulse train (See Figure 7).

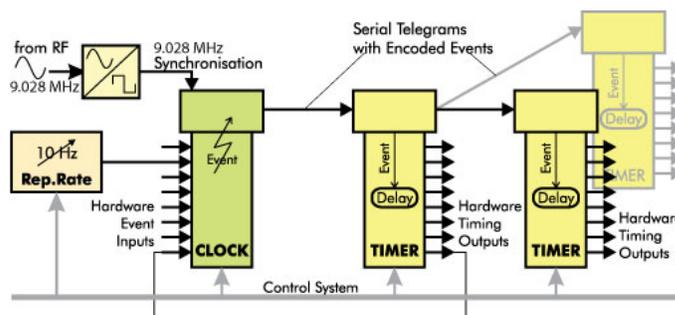


Figure 7: Block Diagram of the FLASH timing system

The 9 MHz is synchronized with the RF master oscillator and gives a stable frequency reference. The telegrams contain coded events and a 32-bit event number, which allows identify the pulse. A single wire transports the signals from the event-generating module to the event-receiving units. Tree-like topologies allow a very flexible distribution of events. Event type and delay times can be programmed in the receiving modules to generate trigger signals or variable gates. The 9 MHz allows for a 110 ns resolution; the jitter is in the order of 0.5 ns.

4.8.5.3 *The Machine Protection System*

A further key component of the FLASH hardware is the machine protection system (MPS). It is based on a PLC with the addition of a collector for fast signals of the order of microseconds. The system checks about 250 slow signals and 100 fast interlock conditions. The FLASH MPS uses e.g. error conditions of magnet power supplies, beam loss detection by multipliers and differential beam current measurements of toroids to switch off the beam within a few microseconds. The system guaranties a safe operation of the linac independent of control system software. The control system just monitors the MPS and brings the status with all details to the operators.

4.8.5.4 *Server Hardware*

The core of the data acquisition server hardware is a multiprocessor SUN Fire E2900 hosting 32 GB of memory and a SUN Fire X4500 providing 22 TB RAID file space using SUNs Zeta byte File System (ZFS). For the necessary network connections the existing infrastructure of the FLASH accelerator (Gigabit Ethernet) is used.

4.8.5.5 *Software Packaging Systems*

On Linux systems, nearly all DOOCS software is packed into Debian packages. The package build process scans automatically for shared library dependencies; therefore all software dependencies will be resolved automatically during installation or upgrade. The “smart” conflict resolution system interface of the package manager is able to solve almost all conflicts.

For Solaris the package system of the operation system is used. The DOOCS server package contains the server binary, an example configuration file and the actual libraries. During initial installation of a server, an entry into the watchdog server configuration is added automatically. For upgrading the server version, a special script is provided.

4.8.5.6 *Remote Installation, Configuration and Maintenance*

Jumpstart is used to install Solaris on SPARC-based computers. It is an automatic installation process, which allows categorizing machines and automatically installing systems of the corresponding category. Scripts called at the end of the installation process will set up the basic DOOCS configuration.

A Fully Automatic Installation (FAI) [14] named schema is used to install and maintain Debian and Ubuntu on x86/64 hardware. It is based on a class concept and is highly configurable and changeable. Configurations are defined in classes. A class is like a system type that can be combined with others. The installation and configuration is completely automated, it does everything a system administrator has to do before a user can login to a new system. It can install and configure the whole operating system and all software packages. FAI is scalable, it allows installing and maintaining any number of client systems. It is also a rescue and disaster recovery system - after a re-installation the system has the same state as before. Furthermore it is used to update running systems without completely re-installing them via the softupdate functionality - so that there is only one single point of configuration for installation and updates. A typical installation of a DOOCS server takes less then 15 minutes. The current DOOCS installation at FLASH contains about 200 Solaris- and 170 Linux-based computers.

4.8.6 Data Acquisition System

One of the main tasks of the DAQ is, to collect all linear accelerator relevant data providing it to (slow) feedback and monitoring services as well as archiving it for offline analysis. The FLASH DAQ can simultaneously record data from accelerator or photon experiments and store this together with accelerator related data in a separate stream. This allows for an easy way to correlate measurements done by the experiments with any collected linac information. The DAQ is fully integrated into the DOOCS control system and its core, the bunch-synchronized data storage, provides a suitable way of serving higher-level applications like the aforementioned (slow) feedback loops and other beam-related calculations.

4.8.6.1 The DAQ Architecture

FLASH is a complex machine containing about 1000 fast ADC channels distributed over tens of VME crates. These channels sample information at rates from 1 MHz up to 8 GHz. The DAQ records data from more than 900 fast channels at rates of 5 Hz and from more than 500 slow channels (HV, pressure etc.) at about 1 Hz. More than 40 fast cameras are being used both for diagnostics and observation. In order to understand the machine behavior one has to be able to correlate any diagnostics channels on the bunch-by-bunch level. The current FLASH DAQ is capable of writing all required data with the full machine repetition rate at 5 Hz with up to 800 bunches per shot to archived data files.

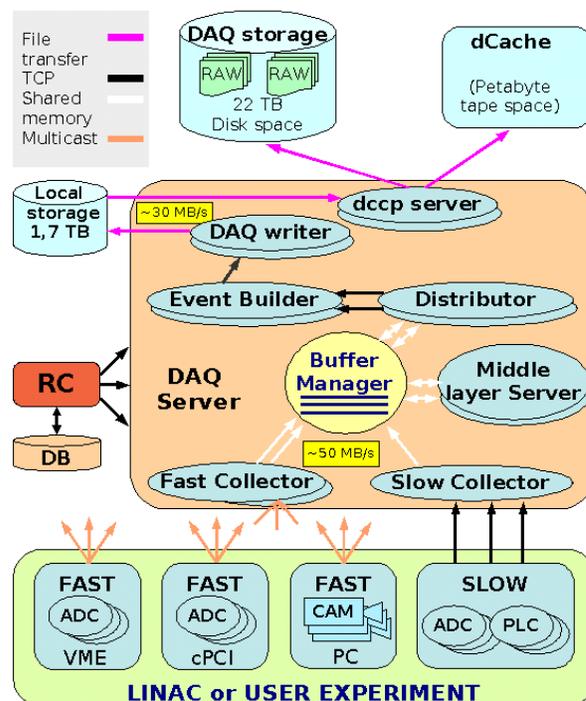


Figure 8: DAQ Architecture

The FLASH DAQ architecture is very flexible and scalable to satisfy the requirements of the machine diagnostics subsystems as well as the experiments. It allows integration of “short live time” experiments. Since the front-ends send the data

as multi-cast packets, several central DAQ systems can run in parallel. Configurations with one to three independent DAQs on one computer or on several computers have been successfully tested.

The architecture of the DAQ system is shown in Figure 8 with the device server layer at the bottom delivering the data to the collector processes, which are located in the middle layer. The central core of the DAQ is the Buffer Manager receiving the data from the collectors and distributing it via Distributor, Event Builder, Writers and dcp servers to the main file and data storage server on top of the figure.

4.8.6.2 The Front-End Hardware

Most of the fast hardware is connected to modules in VME crates controlled by SPARC CPUs running Solaris OS. The cameras are connected to PCs with a Linux operating system. The experiment specific hardware (e.g. compact PCI crates) is controlled by CPU boards running Linux. All hardware is interfaced to the control system via DOOCS device server processes.

For the adoption to the DAQ system the front-end servers read the data from VME (cameras, etc) into the CPU memory after an interrupt from the timing system. The waveform data is then converted and corrected to real physical units and copied to output buffers. To ship the data efficiently to the main DAQ server the multi-cast UDP protocol is used. It is the task of a sender library to push the buffers (server blocks) to the network. This approach allows scaling the whole system to run multiple DAQ servers, which could see the data coming from the front-ends without additional load on them.

Before the server sends data out to the DAQ it has to create a server block (SB). Every server block consists of data from the channels collected by the server. It is headed with additional information containing: time stamp, event ID, server block name (part of the DOOCS name of the server), status, trigger mask, number of channels in the block and the total block size. Every channel included into the server block consists of a channel header containing the channel name, channel type, channel status and channel data length. The channel data format depends on the device type (beam position monitor, toroid, camera, etc.).

4.8.6.3 Data Collectors

There are two types of collector processes: a slow collector for polling at slow rates (max. 1Hz) data from systems like PLCs, etc and multiple instances of fast collector processes listening to the multi-cast messages sent by the fast front-ends. In case of packet losses, fast collectors can request the corresponding front-end sender to re-transmit them. Measurements show that only about one percent of packets have to be re-transmitted. It mostly depends on the load of the front-end CPU.

After receiving packets the server block data are inserted into the buffer manager that makes sorting by the time or the event number.

4.8.6.4 Buffer Manager

The buffer manager [13] is the heart of the system and provides temporary data storage in shared memory, its synchronization on the shot-by-shot basis and a fast access for data reading to several data consumers simultaneously.

The data units the BM deals with are server blocks produced by the front-end servers. All server blocks belonging the same machine shot are merged into an event. There are two event types in the FLASH DAQ system: fast and slow. Fast events contain server blocks either collected by the fast collectors or generated by the middle layer servers. In both cases the data corresponds to one accelerator shot. The slow events consist of server blocks produced by the slow collector.

The BM is a combination of shared memory segments and a library used by the buffer manager clients. There are two types of segments in the BM: control segments and data segments. The control segment is built of a client control and an event control part. The client control segment consists of several slots and is dedicated to the registration of all processes willing to make use of the BM.

Every process has to register with the BM for every event type and a list of server blocks within the event of interest either as a writer (producer) or a reader (consumer) or both. One client can use as many slots as required to register all its requests.

4.8.6.5 Middle Layer Processes

Since all beam relevant data is passing through the buffer manager's main memory, this is the perfect place for attaching slow feedback and monitoring processes. A number of high-level control and monitoring programs have been implemented this way already.

The electron energy measurement in a dogleg of FLASH is a good example. The calculation requires all bunch information of three beam position monitors and a few magnet readings. The algorithm is implemented in MATLAB (see also the chapter: Feedbacks and Automation). To ship the data to MATLAB a special framework has been developed. The library provides input, filter and output channels. All channels are online configurable to select data from the buffer manager, filter data and prepare a structure to a MATLAB procedure. This procedure creates the result in a structure passed to the output channels. The output channels can produce data for another procedure, for the buffer manager or to be sent to any channel of the control system.

With this schema an implementation of slow feedbacks is straightforward. Adding further control loops does not create any additional load on the front-ends because the data is usually available in the buffer manager. If the processors of the central DAQ server run out of free CPU cycles, one more CPU box can be added. This makes the system quite scalable.

4.8.6.6 Distributor and Event Builder

Distributor processes fulfill two main tasks: creating data streams and pushing the data to the event builder. It is taken into account that not all data collected in the buffer manager might be of interest for further analysis. On the other hand for the analysis convenience the relevant data should be stored in the same file. Therefore the distributor makes a selection of events at two levels: on event type level and on data channel level, according to the configuration from the run control. Once the distributor produces an event with preselected data channels, it is sent to the event builder via a TCP/IP connection. The DAQ foresees up to 32 data streams produced simultaneously.

The event builder receives the data streams from the distributor via a TCP/IP link, and therefore can run on a different computer. It sends data to writer processes that produce data files in a data access time optimized format. To compress the data standard

algorithms like ZLIB or LZO are used. The data files are written to a local disk in the corresponding directories according to the experiment name, stream name and some other parameters obtained from the run control. Along with writing data files, some additional files are stored on the local disk (e.g. run catalog, streams descriptions and index files). They contain the information required for the off-line analysis. With their help one can find out the list of channels in the selected stream, names of files to analyze for a certain time period.

4.8.6.7 Data Storage

The writer processes write the files to a local of 1.5TB disk. Copy jobs are used to write the data files to dCache and to the local disk of the storage server with 22 TB disk space. dCache is a hierarchical disk and tape storage system developed for the HEP experiments at DESY and Fermilab [11]. For FLASH it is mainly used as a long-term archive for user experiments. 20 TB of the space is used for keeping the accelerator data for the last 30 days. The remaining space is used to temporary store the data of the experiments for fast data processing.

4.8.6.8 Run Controller

The run control process is responsible for configuration of DAQ parameters. The run controller uses an Oracle database to store all required configurations. The whole information is distributed among several tables. Relations between fields of diverse tables are heavily used to avoid information duplication.

The RC allows preparing the configuration parameters for a new DAQ run as the following. The first step is the selection of the DAQ mode. Every DAQ mode defines a list of hardware and software entities that have to be configured. Every DAQ mode can contain several versions of those parameter sets, meaning the same hardware but with different properties values. One run parameter is usually associated with a group of hardware or software properties. This gives a very flexible way to change a lot of parameters in the system by setting once the corresponding run parameter via the RC.

To prepare the system for a DAQ run the RC takes the following steps: chooses a run mode, selects the run mode version and adjusts some run parameters if required. The RC also provides the possibility of more detailed software and hardware selection for the DAQ usage. This is important if some subsystems have problems to run properly. All those settings are done by means of the run controller GUI. Once a new run mode is set up, the run controller configures all front-end and DAQ processes via the distribution of the parameters. The parameters are sent as XML strings or numerical values. After that the run controller makes use of one DAQ process (not shown in Figure 8) that drives the corresponding Finite State Machines of DAQ processes to step them through “initialized”, “configured”, “ready”, “prepared”, “paused” and “run”. In the “run” state the DAQ starts to collect data according to its configuration.

4.8.6.9 Control System Integration

The DAQ system is fully integrated into DOOCS and therefore control system tools have access to the DAQ parameters. Figure 9 shows the DAQ status overview.

4.8.6.10 DAQ Graphical User Interfaces

A number of Graphic User Interfaces was developed to configure, control and monitor the DAQ system.

The run control DB user interface is written in Java and provides access to the DAQ configurations stored in the Oracle database.

The run control Java based user interface is the main tool to control the DAQ both for the accelerator and for the user's experiments. It allows the selection of all required parameters for the DAQ. It provides the status control of the RC FSM.

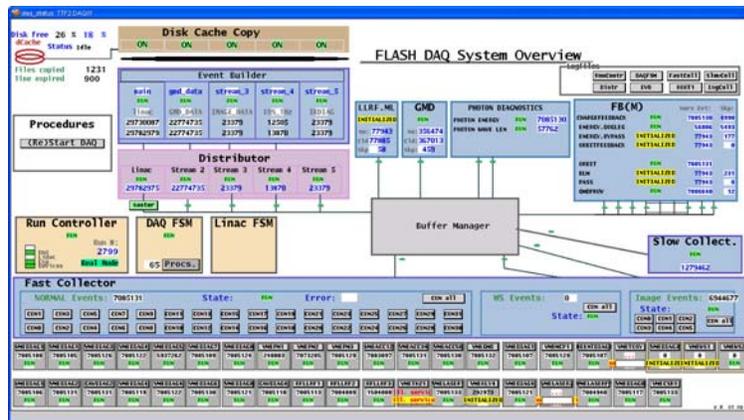


Figure 9: Status display of the DAQ system

A web-based interface is used by the experiments to configure their data streams. The interface allows including and excluding the channels collected by the DAQ to the required data stream in order to have all required data in the same DAQ files.

The DAQ file manager tool written in Java allows users to have an overview of all taken DAQ runs and written DAQ files. It also provides the possibility to stage the files stored on tape to a local disk.

The MATLAB based GUI for the DAQ data access and visualization allows to select the channels and display them as function of time as well as to make some statistics calculations on the selected data.

The Java based tool for the fast access and visualization of stored data is in preparation.

4.8.6.11 Current DAQ Status

With respect to original expectations, it showed that not the machine physicists but instead the photon experiment users at FLASH are making more heavily use of this new possibility for mass data storage. Here the huge data volume produced by various camera systems and some giga-sample ADCs need to be stored and methods for comfortable data retrieval must be provided.

Most experimenters are just doing one experiment at this facility, because beam time in this new wavelength regime is very limited and unique. So experiments are not running that routinely like at common synchrotron facilities and are mostly not prepared for the huge data rates they accumulate within short beam times at FLASH. Therefore it is of great benefit for the photon experiments if the necessary infrastructure for storing hundreds gigabytes is available on-site and can be used right away.

The second major impact the DAQ system has regarding machine operation is the new opportunity to access the full data of the last machine shots in a synchronized way at the stage of the buffer manager. A number of applications need a lot of data from the machine with full resolution and at high repetition rate. Some of these are now already part of the standard machine operation. An API to easily attach C++ and MATLAB based applications has been developed and is already in use since more than two years, proving its practicability and reliability.

Table 1: Data Volumes Stored in the DAQ System for Experiments at FLASH

2005	2006	2007	2008
0.7 TB	4.0 TB	2.5 TB	7.2 TB

The DAQ system is now running since summer 2005 reliably at the FLASH facility. Table 1 shows the volumes of data stored by the DAQ for the experiments since summer 2005.

The novel combination of high energy physics (HEP) techniques and an accelerator control system successfully shows that such an integrated framework can solve the requirements demanded by the high data rates and huge number of components.

After this stage of implementation and commissioning of the DAQ system we need to extend the efforts to ease the access to the stored data. Here it turns out that every specific user group as its own favorite data analysis tool (machine physicists love MATLAB whereby the photon experimenters are more using tools like Igor or Origin).

The chosen design did show its scalability needed for the upcoming generation of FEL facilities.

The FLASH DAQ system was developed in a collaboration of three institutes: DESY Zeuthen, DESY Hamburg and LEPP Cornell University, Ithaca, NY, USA.

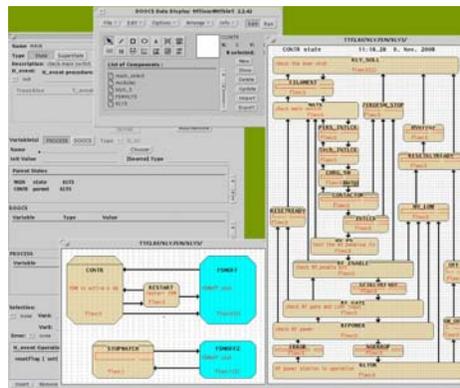


Figure 10: The DOOCS graphical FSM designer

4.8.7 Feedbacks and Automation

Many complex processes for operation but also for measurements done at the FLASH facility require simply “walking” through a number of “states” to get a certain task accomplished. A common way to automate this work is to use so-called finite state machines (often one uses here the abbreviation FSM). The DOOCS API offers a

graphical editor to create these FSMs simply by drawing a sequence of steps and applying the logical connections between them (see Figure 10). In this approach, every rectangle symbolizes a condition of the process chain for which the transition to the next state (determined by the arrow) will occur.

Many complex subsystems like e.g. Klystrons are automated using the DOOCS FSM.

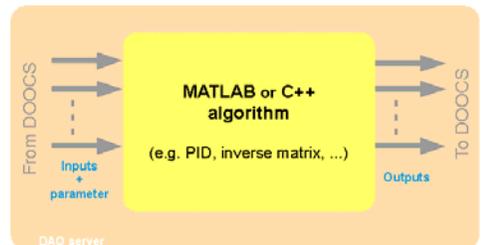


Figure 11: DOOCS generic feedback API

Among the control and automation of well-known process chains that can be driven by FSMs, there are also a number of correlations within the machine that can only be properly controlled using feedbacks. Here one can divide the application areas into two distinct classes: Fast hardware controlled loops and slower software driven feedbacks. The hardware loops are widely used in e.g. the low level RF regulation and will not be discussed here. Many software feedbacks have been developed and commissioned using MATLAB (see chapter MATLAB). This is due to the excellent rapid development cycle one can achieve by this approach. Feedbacks (but also high level monitoring processes) used for standard operations are better controllable if they are more closely integrated into the control system (e.g. to prevent concurrent versions of one program running at the same time). The DAQ offers a good platform for attaching these high level processes on this level, since here all machine data passes by in a synchronized way. For this we developed a dedicated API to attach and operate such software loops.

The general idea is to provide a configurable number of input and output channels to the programmer (see Figure 11). Between these input and output channels either C++ or MATLAB algorithms can be placed to manipulate and add data and provide it to the DOOCS API.

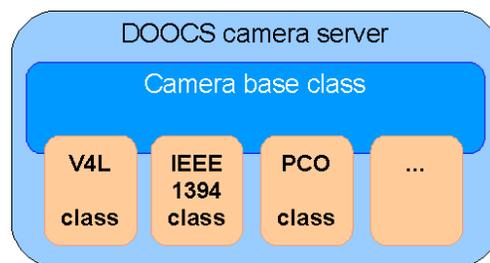


Figure 12: The DOOCS camera server

4.8.8 Camera Controls

FLASH operates more than 100 cameras of different types. To develop a common interface for different camera types and frame grabbers a class library was developed. A camera base class implements all common functions in C++. The DOOCS camera server makes use of camera specific classes to provide the access to a specific camera type. The server controls all basic and some high level imaging operations (histograms, background subtraction, region of interest, etc.)

All Camera server computers are equipped with timing modules. The timing system distributes a programmable event number incremented with every linac shot. Images from triggered cameras include this event number as well as a time stamp required for synchronization with other linac diagnostics data.

All camera parameters can be controlled remotely via the DOOCS server. The initial camera settings are stored in a camera server configuration file. Therefore parameters are preserved after a server restart. All cameras connected to the camera server are detected during the start-up process.

The camera server provides the access to the recent grabbed image via a DOOCS property. The property contains a header and the image byte array. The header includes an event number, a time stamp, and image parameters such as resolution, binning, image format, rotation, etc.

All camera independent features are implemented in the common camera base class. Like e.g. Image X/Y flipping, calibrations, background subtraction, offsets. Image pixel histogram calculation, X & Y Spectra with calculations of min, max, mean, sigma, center of mass etc. The camera server takes care to convert values into physical units.

To reduce the data and to get a better signal to noise ratio a software-based region of interest with different shapes is implemented. More than one region of interest can be defined.

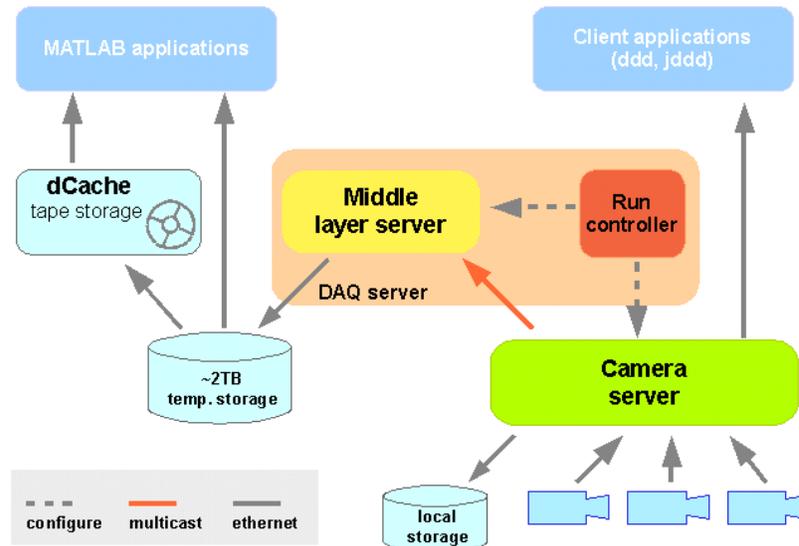


Figure 13: Data flow for the DOOCS camera server

For the display different visualization modes are implemented. The user can select between raw-data formats to JPEG representation (to save network bandwidth). Central storage of all image raw data is done via the DAQ System.

Camera specific features like hardware region of interest, hardware binning, loops (summing up images from sequential triggers), multiple or sequential images from one trigger with specific cycle, delay and exposure tables are implemented in the camera type specific class.

Software regions of interest calculated by the camera servers can be also stored in the DAQ if required. The data flow from the camera servers to the DAQ is shown in Figure 13.

The camera servers send images to the DAQ via IP multicast packets. One image is usually built of several packets. The DAQ server collects the packets belonging to one image including the header information. The full image information is made available to the middle layer and feedback servers by means of a shared memory facility, as described in the previous chapter.

The camera server also provides an interface for writing images to a local storage. The file is written in the device-independent bitmap (BMP) format. An image can either be overwritten or a series of image files with time stamps can be generated.

Camera systems are in operation for linac diagnostics and measurements of the FLASH FEL experiments: laser beam lines, various electro optical beam diagnostics, transverse deflection bunch length measurements and other bunch diagnostics, beam position measurements, infra-red spectra, beam energy measurements, etc. More than 100 cameras are installed in FLASH and eight different camera types are implemented so far:

- Cameras based on standards like v4l (Video for Linux) compliant hardware (digital and analogue cameras, TV and video cards), IEEE1394 (fire wire) based and Gigabit Ethernet cameras.
- Special cameras e.g. *Andor iStar/iDus*, *PCO dicam/sensicam* and *uEye*.

4.8.9 MATLAB

At FLASH many projects (e.g. automation, diagnostics) started with experimental hardware and small scripts driving and controlling this piece of hardware. The most commonly used software for accomplishing these tasks is and was always MATLAB. MATLAB is a numerical computing environment and programming language with strong capabilities in the matrix and algebra sector. This allows experts as well as operators to easily read and manipulate many variables at once, which is a common task if one works with an accelerator. Here tasks like reading all beam position monitors X and Y values and displaying these in a plot, or reading a camera image and manipulating it pixel-wise to e.g. apply some background subtraction, are often used standard operations.

Due to the popularity of this software package, the control system group implemented an interface to directly read and write all control system variables from the MATLAB API. This interface offers full access to the variety of DOOCS data types and thus allows creating scripts fulfilling sequences of operations on control system hardware.

Besides hundreds of small scripts for tasks like e.g. measuring resolutions, stability or standard statistical variables there is also a number of bigger projects realized in

4.8.10 Tools: Alarm, jDTool, Save & Restore

4.8.10.1 Alarm & Info System

One important part of the control system is an alarm tool that gathers all warning and error reports from different controlled components of the machine. The central place to collect errors in DOOCS system is the central Alarm&Info Server, which receives XML-based messages from all device servers and sends them to a Java-based Alarm&Info GUI. An error or device information message contains fields with the error location (DOOCS address), timestamp, severity, the error text itself and some other criteria, which allow later examination of the failure origin. The Alarm Server parses the incoming XML strings and stores them in its local memory. In parallel, it sends the messages to the GUI and indexes them for the further search requests. The Alarm GUI shows the current status of error reports from the device servers on the known DOOCS hierarchical device tree. The errors are passed from the lowest level (properties) up to

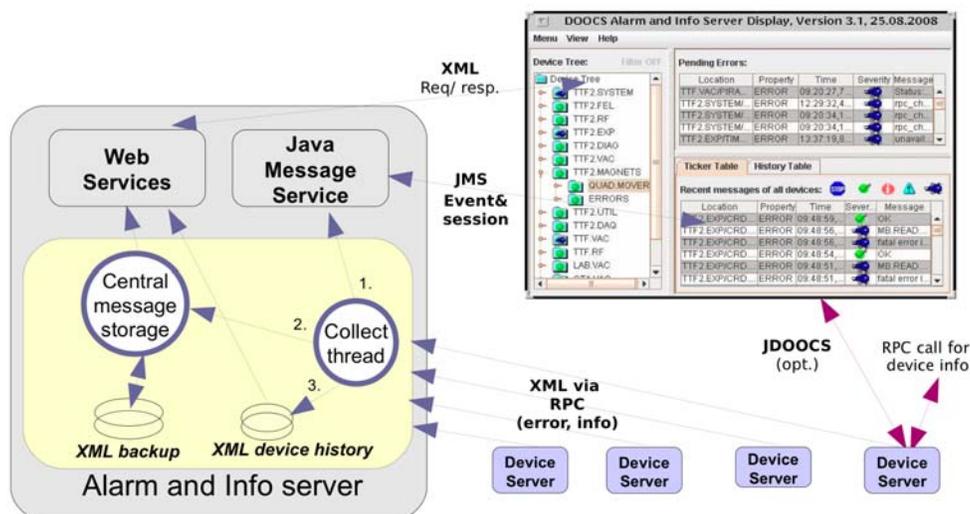


Figure 15: Architecture of the Alarm & Info Service

the facilities, so that locating the errors is easily done browsing from the top of the tree. The GUI displays also the current messages, pending errors, message histories from all sublevels of the tree. It provides also the possibility to define and save in the properties file the user's individual settings, like location or severity filters, length of the history entries, number of stored messages. The following picture shows some details of the Alarm&Info System implementation:

4.8.10.2 The *jDTool* application

In the control system there is a big number of locations of the same device types. Typical examples are pumps, magnets or sensors. Every device of some type, e.g. pumps, has the same set of properties. Especially for the initial setup but also to modify the configuration at runtime it is convenient to change the same property in several locations simultaneously. This is where the *jDTool* comes into play.

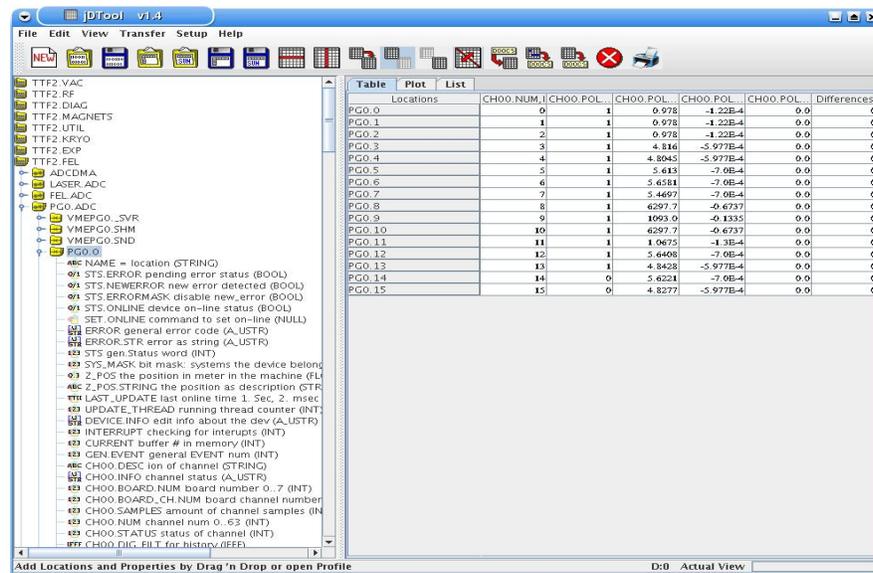


Figure 16: Snapshot of the *jDTool*

jDTool is the abbreviation for “Java DOOCS Tool”. Its main goal is to help the user to handle a bigger number of devices of the same type. This tool shows the complete view of the DOOCS device tree, where the properties of each device can be selected and dragged to the right-side spreadsheet table (see Figure 16).

The values of the properties could now be edited, saved and reloaded, as well as read from or sent to a DOOCS, TINE or TANGO server. The program also compares the values in the table with those values, which are read from the server. If there are differences, they will be highlighted, and the number of deviations (per location and total) will be counted.

For editing the values there are different options available. They apply one of the following schemes to one of the selected properties in all selected locations:

- Set all to the same value
- Build a series

In the first case, all selected values will be set to the value of the first selected one. In the second case, the difference between the first two selected values will be calculated and used as an increment to extend the first two values to a series in all selected cells. If the first two values are 1 and 2, the series will be extended to 1, 2, 3, 4, etc. If they are 4 and 6, the series will be 4, 6, 8, 10...

Like highlighting of the distinguishing values, also edited values are highlighted, but in a different colour. Since the user can choose to send either all or just the edited values to a server, the highlighting helps easily check the changes before sending them.

Besides the previously described main functions the program also offers specialized windows for handling array data types like spectrum or history. Those values could be shown either as plots or as lists.

The program offers some special functions and tools to simplify the location and property selection as well as storing and retrieving of the values, tuning the setups and printing of the tables.

The selection of locations and properties is mainly done by drag and drop from the DOOCS tree (on the right side of Figure 17). Also functions like filtering, sorting and automatic search of similar locations in different devices within a facility are supported.

For file handling two file types are supported: on one hand, the data of the shown table could be saved or loaded in a CSV format, which later can be read and modified by programs like MATLAB or Microsoft Excel for further processing or generating of the data. This feature is a replacement for the older DOOCS-based Save & Restore application. On the other hand, profiles are used to hold all information about the chosen locations and properties. The profile could be saved and reloaded as a file or placed in a versioning system.

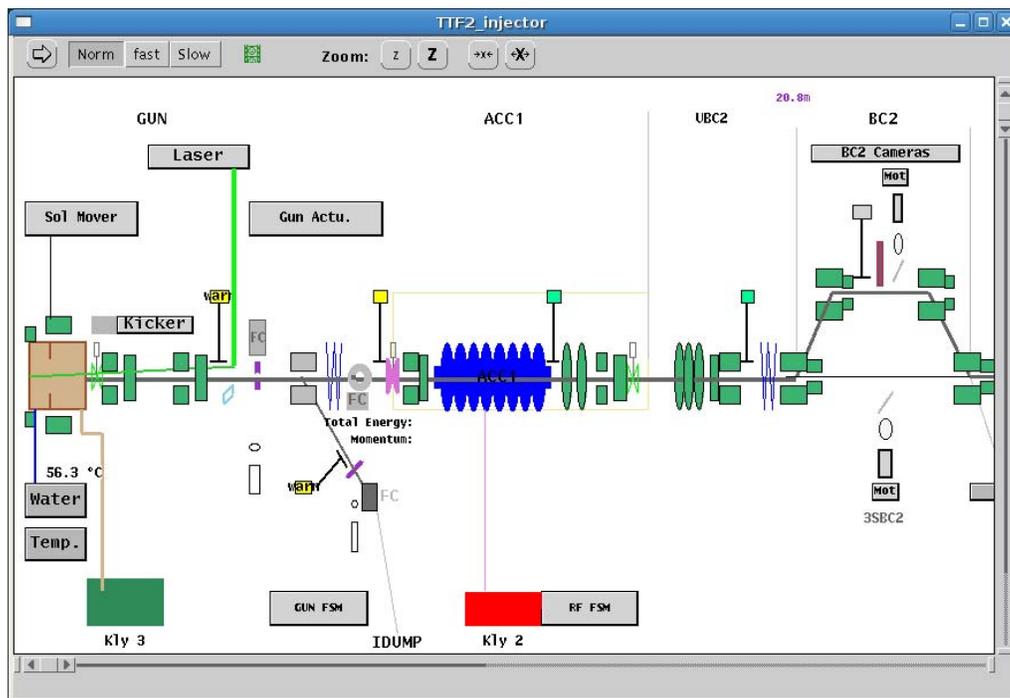


Figure 17: Screenshot of a ddd display in run mode.

4.8.11 The DOOCS Data Displays ddd and jddd

4.8.11.1 The ddd Application

The commissioning of the FLASH accelerator at DESY required the creation of numerous control panels. For a quick and easy creation of these panels the DOOCS

Data Display *ddd* [7] has been developed. *ddd* is a graphical editor for designing and running synoptic displays. It offers a set of reusable components/widgets like labels, buttons, plots and an “IF” component providing logic functionality. With the *ddd* editor graphical displays can easily be designed without programming. This concept has proved to be a good choice. FLASH operators and experts have designed more than 1300 control panels until today. Figure 1 shows one of these displays in run mode. All graphical elements are animated to indicate the device status by color or form. Device specific sub-panels are displayed on mouse click.

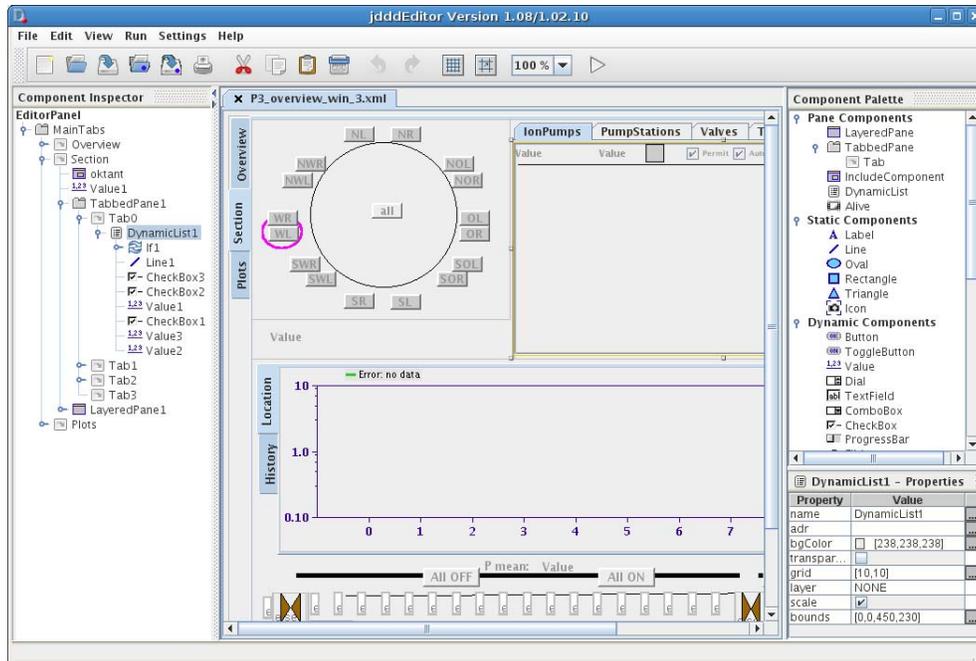


Figure 18: The *jddd* editor.

4.8.11.2 The *jddd* Application

The 12 years old *ddd* doesn't match future requirements any more, some libraries are no longer supported and new features are difficult to implement. For this reason *jddd*, the new Java DOOCS Data Display [9] [10], has been developed as a replacement of the old *ddd* program. The existing Java DOOCS API [8] was reworked to provide access to multiple control systems, currently DOOCS, TINE and TANGO. For compatibility reasons *ddd* panels can be exported to the new *jddd* XML format.

4.8.11.3 The *jddd* Editor

Like *ddd*, the new *jddd* has a graphical editor with a set of ready-made components/widgets for control panel design. The main goals for the editor design were stability, extensibility, high performance, a standard, ergonomic and intuitive machine interface and a modern look & feel. Because of the platform independence it was decided to use Java and standard Swing components. For all reusable components like labels, buttons, etc. the Java Beans technology was used.

The jddd editor layout was designed similar to other existing standard IDEs like Netbeans or Eclipse (see Figure 18). It consists of an *Editor Window* (center) for a design-time view of the control panels, a *Component Inspector* (left) displaying a tree hierarchy of all components in the currently opened editor tab, a *Component Palette* (upper right) containing a list of all available components that can be used for control panel design and a *Component Properties table* (lower right) displaying the editable settings for the currently selected component(s).

4.8.11.4 jddd Components and Widgets

jddd offers all components needed for control panel design. There are five different component types:

- Pane components, which help structuring the control panels (i.e. tabbed panes).
- Static components like labels, icons and other graphical components.
- Dynamic components, which are used to display and change control system values (i.e. buttons, values, checkboxes, sliders...).
- Logic components, whose appearance depends on a user defined control system value (i.e. the “If” and the “Switch” component).
- Plot components, which display one or multiple data channels.

In the following paragraphs a couple of useful jddd components will be introduced.

Plot Components

The plot components are designed for high performance data presentation. Four different plot types are currently available in jddd: time domain, history, location and XY plots. In the plots multiple channels can be displayed and new curves can be added by drag and drop. Some regions of the plot can be selected with the mouse to analyze the data in these limits, e.g. using Fourier transformations, linear fits, displaying RMS, min/max values, showing histograms or making data correlations.

Dynamic Lists

The dynamic list is (as the name implies) a component used for dynamic panel creation. It consists of a panel where different kinds of jddd components, displaying control system values, can be placed. At runtime jddd automatically fills the lists with components for all available locations. For example, a dynamic list in run mode is shown in the upper right corner of the panel in Figure 19. All available pump stations with corresponding pressure values are displayed in this list. A special feature of the dynamic list is the location filter. If the location of the list is set to “SOL*”, only locations starting with the letters “SOL” will be displayed.

Buttons

A jddd button can be used not only to set one or multiple control system values, but also to perform various other functions like:

1. Open a new window, replace or override an existing one.
2. Set component property (to set one or multiple property values)
3. Execute a shell command
4. Print the panel
5. Open a related help page in a web browser

The most interesting function for creating interactive displays with jddd is the “Set Component Property”, which can be used, e.g. to change in the run mode the component's content of a Dynamic List or the reference to an XML file.

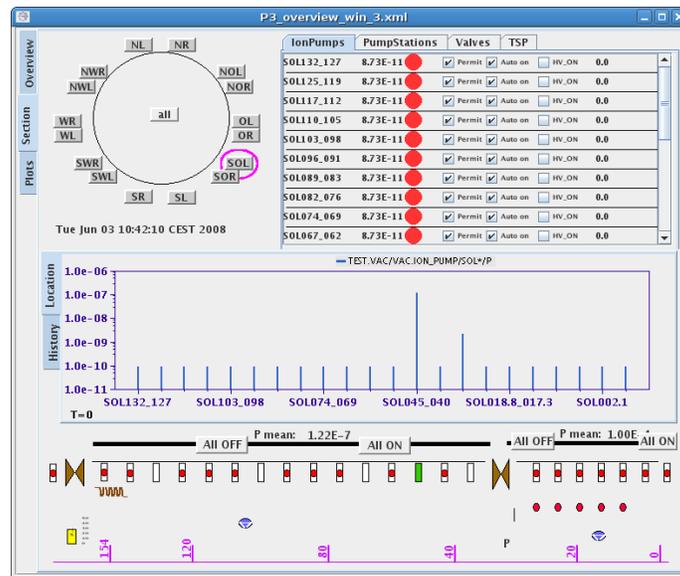


Figure 19: Running jddd panel for PETRA vacuum controls.

4.8.11.5 *jddd Edit and Runtime Mode*

The jddd program has two different execution modes. The first one is the edit mode for drawing and testing panels. The created panels are saved in a XML file format. The second one is a runtime mode, where these XML files are parsed and executed.

The data update rate can be set for each component. The archived data are accessed by a mouse click on the component.

The displays can be organized in a hierarchical manner. Buttons or sensitive areas can be defined to open new panels at runtime mode for more detailed information on certain devices or subsystems.

A drag & drop functionality enables the operator to drop additional data into a plot and e.g. to compare spectra at runtime. For documentation the displays can be printed directly or else a screenshot can be saved in PNG file format.

4.8.11.6 *jddd Editor Features*

All displays may be used as generic library components and may be added to other jddd displays. For example, if there is a panel displaying a steerer, multiple instances of these steerers can be included in a new panel displaying the beam pipe. Then each steerer gets its individual device address.

Library components help to create control panels in an efficient way, because similar components which differ only in the device address have to be created only once and can be reused afterwards. In addition, changing the original component changes all its clones. This saves a lot of work, because instead of editing every single component, only the “template” has to be modified.

The development of high-level applications is simplified with different export possibilities of jddd displays (see Figure 20):

The first one is to save a jddd display as a Java JFrame or JPanel. This way complex displays are easily designed with the jddd editor and complex functionality can be added to the Java source code. The only disadvantage of this method is that panels cannot be revised with the jddd editor after external modification of the Java code.

The second way is therefore more flexible: jddd displays may be reused as Java Beans in other applications. Only a few lines of codes are needed to add a jddd display to an existing Java application. The properties of all components in this jddd display can easily be accessed from the application. This way the jddd XML files can be modified any time without affecting the applications source code.

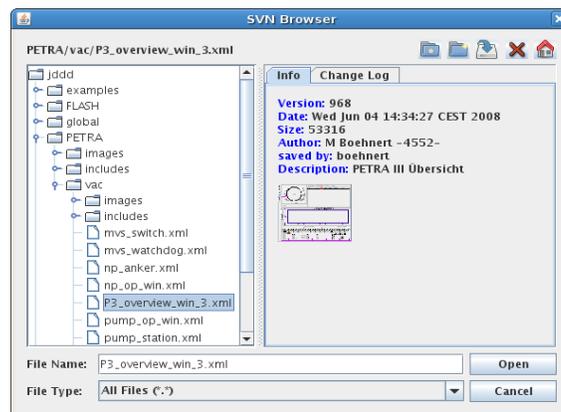


Figure 20: The subversion browser.

The panels are saved in an XML file format and kept either on local disk or in a centrally hosted subversion repository, which allows proper bookkeeping and version tracking. A new Java subversion browser was developed using an open source SVN library (svnkit). With this browser, files are committed to, loaded from, updated, moved or removed from a user specified subversion repository. For file browsing there are two different views available:

- In the default “Info” view all information about the most current file version is displayed. For jddd files these information includes version number, date, file size, author name, description text and a thumbnail preview (shown in Figure 20).
- In the “Change Log” view older file versions can be viewed and loaded.

Many people have spent a lot of time to develop high-level graphical Java beans with very special functionality. These beans may be imported to jddd using a plugin interface. Only a *plugin.jar* is needed containing the wrapper classes for each Java bean. If the *plugin.jar* path is set in the jddd preferences, the beans, which are included in the jar, will be automatically visible in the component palette and may be added to a panel using drag and drop. All bean properties that are specified in the wrapper classes are editable in the jddd component properties table using the standard property editors. The plugin technology allows other institutes to develop their own widgets and to use these widgets in the jddd editor.

4.8.12 DOOCS Related Web Services

For informational and documentary purposes of FLASH operations the web is of high importance. One essential tool in the control room is the electronic logbook [12] developed by the DOOCS group. This tool offers a comfortable way for entering and searching any operation related information through a well-known interface: a standard web browser. The DOOCS version of the electronic logbook has gained worldwide a very high popularity. Many other institutes like e.g. the SLAC National Accelerator Laboratory, the Paul-Scherrer Institute or the Australian Synchrotron Radiation facility to name only a few, have either adopted the technology or simply run copies of the FLASH logbook.

4.8.13 Conclusion and Outlook

More than a decade ago the DOOCS project started from an object-oriented approach connecting only a handful of servers for a test facility. The modular design, based on libraries, has been proven to be a very solid and scalable system well suited for today's accelerator requirements. Although the core system and principles are kept, DOOCS has meanwhile evolved into a full control system framework including a high performance DAQ and Java-based client applications replacing the original ones. Hence, DOOCS is well prepared to accommodate the challenging requirements of a control system for the XFEL. Development continues to integrate support for state-of-the-art technologies like the new Advanced Telecom Computing Architecture (ATCA and μ TCA) hardware standard, or the extended use of modern Web-based technologies to offer secure and easy-to-use access for all collaborating institutes.

4.8.14 References

1. T. Åberg et al., "A VUV Free Electron Laser at the TESLA Test Facility at DESY - Conceptual Design Report", DESY Report TESLA-FEL 95-03, Hamburg, June 1995. And TESLA Collaboration, J. Andruszkow et al., "First Observation of Selfamplified Spontaneous Emission in a Free Electron Laser at 109 nm Wavelength," Phys. Rev. Lett. 85, pp. 3825-3829, 2000. See also the project page at <http://flash.desy.de>.
2. XFEL project page: <http://www.xfel.eu>.
3. G. Grygiel, O. Hensler, K. Rehlich, "DOOCS: A Distributed Object-Oriented Control System on PC's and Workstations," PCaPAC'96, Hamburg, Germany. See also <http://doocs/desy.de>.
4. Piotr Bartkiewicz and Philip Duval, "TINE as an accelerator control system at DESY," Measurement Science and Technology, 18, 2007, p. 2379. See also <http://tine.desy.de>.
5. Dalesio, L. R., Kraimer, M.R., Kozubal, A. J., "EPICS Architecture," in Proceedings of International Conference on Accelerator and Large Experimental Physics Control Systems, C. O. Pac, S. Kurokawa and T. Katoh, Eds., Tsukuba, Japan, 1991, pp. 278-282. See also <http://www.aps.anl.gov/epics/index.php>.
6. A. Götz et al., "Tango – A CORBA based Control System," in Proceedings of International Conference on Accelerator and Large Experimental Physics Control Systems, ICALEPCS'03, Gyeongju, Korea, 2003. See also <http://tango-controls.org>.
7. K. Rehlich, "An Object-Oriented Data Display for the TESLA Test Facility," ICALEPCS'97, Beijing, China, June 1997.
8. K. Rehlich and V. Kocharyan, "JDOOCS – a Java Library for DOOCS," PCaPAC'02, Frascati, Italy, October 2002.

9. E. Sombrowski, A. Petrosyan, K. Rehlich, P. Tege, "jddd: A java DOOCS Data Display for the XFEL," ICALEPCS'07, Knoxville, Tennessee, October 2007. See also <http://jddd.desy.de>.
10. E. Sombrowski, K. Rehlich, "First Experiences with jddd for Petra Vacuum Controls," PCaPAC'08, Ljubljana, Slovenia, October 2008.
11. M. de Riese et al., "The dCache Book", <http://www.dcache.org/manuals/Book>. See also <http://dcache.desy.de>.
12. R. Kammering et al., "The Electronic Logbook @ the TTF-VUV FEL – Making the Next Step," PCaPAC'05, Hayama, Japan, March 2005. See also <http://tesla.desy.de/doocs/elogbook/elogbook.html>.
13. V. Rybnikov et al., "Buffer Manager Implementation for the FLASH Data Acquisition System" PCaPAC'08, Ljubljana, Slovenia, October 2008.
14. Fully Automatic Installation, FAI, see <http://www.informatik.uni-koeln.de/fai/>.

4.9 TINE: The Control System for PETRA3 and the DESY Pre-accelerators

R.Bacher, P.Bartkiewicz, F.Brinker and P.Duval
 Deutsches Elektronen-Synchrotron, Germany
 Mail to: reinhard.bacher@desy.de

4.9.1 Introduction

4.9.1.1 The DESY Accelerator Chain

The accelerators LINAC2 and DESY2, the accumulator ring PIA and the synchrotron light facilities DORIS and PETRA3 comprise the accelerator chain at DESY. LINAC2 is a linear pre-accelerator delivering electrons or positrons which are injected into the fast cycling synchrotron DESY2. From DESY2, the beam can be transferred either to DORIS or PETRA3. Presently all accelerators with the exception of DORIS are undergoing extensive renovation and upgrade, with a major impact on both front-end hardware and controls software.

Control and supervision of the various accelerator subsystems such as RF systems, magnets, beam diagnostics etc. is mainly performed through PCs running Windows or Linux. The distributed control system components communicate via the Ethernet and with attached hardware elements via fieldbus networks as well as data bus links. Classic UNIX workstations, such as Sun workstations running Solaris, Hewlett Packard machines running HP-UNIX, etc. are also in use and still play a prime role in accelerator control at DESY. A number of subsystems make extensive use of VME standard based computers running VxWorks. More recently, much more use is being made of embedded system solutions, such as PC104 running ELINOS (an embedded Linux distribution) or ALTERA with NIOS II core. In some rare instances (DORIS RF) there are some legacy systems using MS-DOS.

The various control system computers represent a large-scale distributed control system controlling and supervising many thousands of front-end controllers in the field. The principal control system in the DESY accelerator chain is the TINE system [1]. However, in these modern times, one is often required to seamlessly integrate elements from other systems. In the DESY accelerator chain, this means interfacing with EPICS

iocs (facility infrastructure), TANGO (beam-line control elements at HASYLAB), and with DOOCS (vacuum elements).

4.9.1.2 *Requisite Features of the Control System for the DESY Accelerator Chain*

Now that we have highlighted some of the details and requirements concerning the integration and control of the subsystems and components found in the DESY accelerator chains we are able to list a set of criteria, which must be available in the control system:

- Generally:
 - *Multi-platform*: A common library and interface should be available on various hardware platforms and operating systems.
 - *Multi-protocol*: Since network infrastructure has developed with the accelerator chain growth, the system should integrate various Ethernet technologies.
 - *Multi-architecture*: Here we are referring to ‘data-transport’ or ‘data-exchange’ architecture. There are several reasons why this feature must be present. One is a need to provide compatibility with different ‘local’ control systems which are to be integrated. Another is the data transfer efficiency for certain cases. This is an important point, especially where scalability is concerned, and will be discussed later.
 - *Capability of transferring various data types*: Any set of primitive data types (simple integer or float arrays, for example) or complex data types (e.g. mixed doublets of two primitives, or user-defined structures) between different platforms should work seamlessly, without concern for byte-swapping between dissimilar platforms (little-endian versus big-endian).
 - *Clear and easy method of application specific code integration with the system*: This involves an easy to understand interface where it is clear to a developer when he/she should integrate the own code (if necessary) into the system.

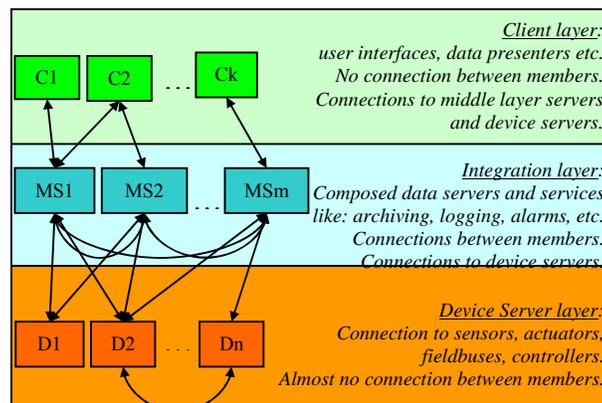


Figure 1: Three layers of a control system

- Device layer (Figure 1):
 - Capability of interfacing to various field busses: The most popular field busses at DESY are SEDAC (DESY proprietary field bus), CAN, ProfiBus, RS232, and GPIB.
 - Small footprint: The control system kernel library should have a small footprint in order to fit inside embedded systems, where resources are frequently limited.
- Integration layer:
 - Naming service: All servers and their services are addressed only by names. The huge number of servers excludes physical addressing.
 - Well-defined integration strategy for other control systems existing at DESY, such as EPICS or DOOCS.
 - System for filtering and archiving data, events, alarms etc.
- Client layer:
 - Client browsers for the easy access of data provided by any existing server.
 - Generic tools offering access to archived data, alarms and event viewers.
 - Client widgets which can connect to any control system endpoint and can be used in rapid application development in both simple and rich clients.

Among many modern control systems, commercial or open source, we shall see below that only TINE can cope with the above requirements.

4.9.2 TINE: The Control System Expert's Perspective

4.9.2.1 *Three-fold Integrated Network Environment*

The key word in the TINE acronym is “Integrated”: this is a multi-platform system, supporting currently: MS-DOS, Win16 (Windows 3.X), Win32 (Windows 95, 98, NT, 2K, XP), UNIX (Solaris, HP-UX, OSF, SGI, Linux, FreeBSD), MACOS, VAX and ALPHA VMS, VxWorks, and embedded systems such as ELINOS, ALTERA-NIOS II and most recently Window CE. TINE is a multi-protocol system, where data exchange among the participants can occur via any of the UDP, TCP/IP, or even IPX protocols, as well as shared memory. TINE is also multi-architecture, where data transfer can follow any of the strategies listed below:

- **Client-Server:** This is a traditional transaction-based data exchange mechanism, available in most control systems. It is pure, synchronous client-server data exchange, where a client makes a request and waits for the completion of the request. This is also a necessary mechanism for sending commands to a front end, where the next action to take depends on the outcome of the command. The client-server approach has however two disadvantages: if the server for any reasons goes down, or a network problem occurs, the client application will wait until the timeout mechanism considers a transaction to be aborted. This problem can be overcome to some extent by incorporating threads. However, the second disadvantage appears, when several clients want the same information (regular updates, i.e. synchronous polling of control data for instance), a server will see each request from each client separately. Imagine a client asking a server “is everything okay” once per second

because that is the only way to obtain the information. This can become a burden to the server if many clients (say 50 or more) are all getting a kilobyte's worth of data at 1 Hz, as the server will have to deliver the data at 50 Hz.

- **Publisher-Subscriber:** For many cases, a much better approach is the publisher-subscriber data exchange: client (the subscriber) communicates its request to a server (the publisher) and does not wait for a response. Instead it expects to receive a notification within the timeout period. This can be a single command, or for regular data acquisition it can be a request for data at periodic intervals or upon change of data content or event. In this format, the server maintains a list of clients and what they are interested in. Now if many clients all want the same kilobyte's worth of data at 1 Hz, the server must acquire this data set only once per second, and notify the clients on its list. This is much more efficient than the client-server model in such circumstances. Imagine a client asking a server to inform him only if "everything was okay but is suddenly NOT okay" or vice versa.
- **Producer-Consumer:** A third alternative for data exchange is the Producer-Consumer model. In this case a producer transmits its data via broadcast on the control system network or via multicast to a multicast group. Consumers simply listen for the incoming data. For most control systems, there are certain parameters which are of system-wide interest. In the accelerator world it might be beam-energies, beam-currents, beam-lifetimes, or accelerator states etc.
- **Producer-Subscriber:** A hybrid between the above two modes is also possible under TINE, in which subscribers request data to be produced on the network (a "network subscription"). This is an especially useful method of data transmission for either large scale machines with many clients needing the same data or large amounts of data sent to multiple clients (such as video). Using the Producer-Subscriber architecture makes the most efficient possible use of the available network bandwidth, and is an excellent solution when scalability problems arise.

Each of above modes of data exchange could be independently used to define the control system architecture. Very small systems with a manageable number of users can run well with simple client-server data exchange. Larger Facilities, however, will frequently want to incorporate all data-exchange modes in some capacity. Video signals for display purposes for instance are best sent via multicast (Producer-Subscriber) data exchange.

For simplicity, in the next sections of this article, the term '*server*' will mean a data producer in general, not only a server in a 'client-server' approach context. The same refers to the client and data consumer: the term '*client*' will be used.

4.9.2.2 *Control System Services*

TINE offers several central services (which are themselves running as TINE servers). It should be noted that these services are all optional components to the control system, although some are strongly encouraged. The most important of these are:

Equipment Name Server: In TINE as in any control system, user application programs (clients) make use of the services provided by device servers or middle-layer servers. Finding a service in the system amounts to matching a human-readable device server name to its network address (which will be in most cases an IP address and a listening

port). TINE addresses are resolved by contacting the equipment name server (ENS). The TINE ENS uses a “*plug & play*” mechanism for servers, which, upon start-up, ‘plug’ themselves into the ENS database, thus obviating the need for a control system administrator to add new server addresses by hand. Any address changes are verified by the ENS, which can refuse to update its database and issue “address in use” messages when warranted. Due to this feature, the TINE ENS can operate without intervention by a control system administrator. Nonetheless, a control system administrator can browse the database and make changes via a graphical user interface tool (Fig. 2).

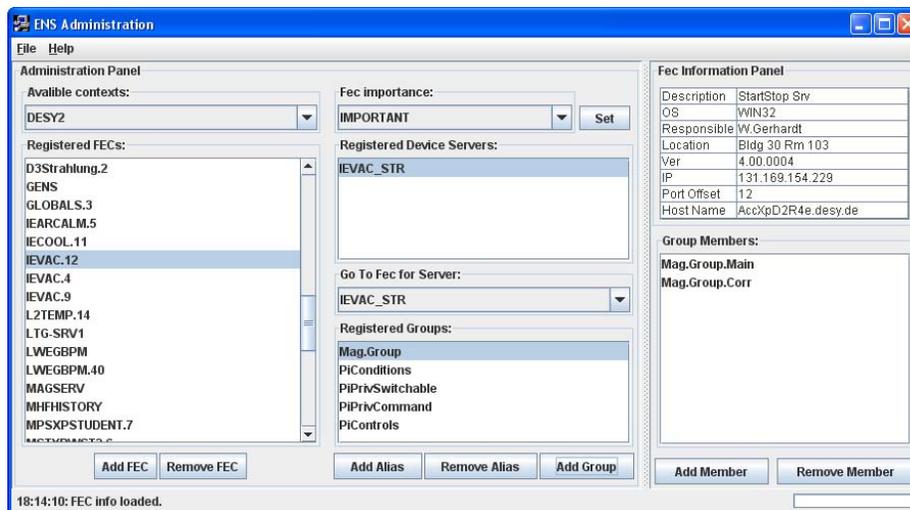


Figure 2: ENS browser

Archive services: TINE offers three different systematic ways for archiving data:

- **Central archiving.** Central archiving consists of acquiring the specified data, passing it through the relevant filters and writing it to disk if warranted. Here, the ‘filtering’ can be adjusted to store data based on state and/or tolerance criteria. For archived data manipulation and display a set of client viewers is offered. These offer a wide range of analysis features, allowing among other things, fast and easy data browsing, comparison and correlation of data sets, movie playback, and multi-channel array analysis (a TINE specialty).
- **Local archiving.** All TINE servers contain a built-in local history server (LHS), which can be activated by request. This means, that local archiving takes place at the server, where the server can be configured to maintain an archive of specified properties both on a short-term basis (in main memory) or on a long-term basis (on disk). In this case, the filtering can only accommodate tolerance specifications. Queries for archived data are managed on an independent thread (when threads are available) so as not to disturb other device server functions.
- **Event-driven archiving.** In contrast to the case of central archiving, where data is collected periodically, applying defined filtering rules, event-based archives are archive snapshots which are based on scripts specifically targeting the event in question. In the accelerator world the event can be a message such as: “the beam was lost due to a magnet quench” (i.e. a post-mortem event), or “proton injection complete” (i.e. a normal operational event). In such cases it is desirable

to store a snapshot of the relevant machine parameters (like machine optics, current orbit, etc.) at that time. A scan of 'injection' events over a time range (say, the past week) would then specify precisely when beam was injected and the stored information could easily be examined at those times. Event scripts can also trigger other activities by issuing commands to other servers, transient recorders, etc. delivering data sampled with higher rate.

Alarm Services: TINE alarms are processed at two levels. The first level is located directly at the front end and is known as the Local Alarm Server (LAS). The LAS is a standard part of all TINE servers, and it is here that it is most easily determined whether an alarm is oscillating (coming and going), is persistent, or has terminated, etc. or whether an 'alarm storm' is brewing. The second level of processing occurs at a dedicated middle layer TINE server known as the Central Alarm Server (CAS), where alarms are collected, filtered, sorted, and made available for client-side review. The CAS also performs several additional tasks more appropriate to central level processing. It can determine whether 'server-down' alarms need to be issued in the case of a non-responsive server (something an individual server of course cannot do). It can also take 'actions' following the receipt of particular alarms. For instance, it can issue event triggers for archiving, send e-mails, or write reports, etc. The TINE Alarm Viewer receives data primarily directly from the CAS.

Synchronization Service: The TINE timeserver is a server operating in producer-consumer mode and produces one quantity, namely the current TINE data timestamp, which it sends via multicast at one second intervals. A TINE server can then intercept these multicasts and apply an offset to its own clock when time-stamping its data. Optionally, a "Cycle" server can likewise issue a systematically applied "cycle number" via multicast. This is an additional integer data stamp applied to all TINE data objects (in addition to the data timestamp) and can likewise be used as a cycle 'trigger' when needed.

4.9.2.3 *Special Features*

From the very beginning, TINE was designed to be multi-platform and to have a small footprint. Consequently, the kernel is written in straight-C (as are most operating systems) and makes use of the Berkeley Sockets API, which is available on essentially all platforms. The exception to this is the java kernel, which is written in native java. There are no third-party dependencies whatsoever. This is not true, for instance, of any control system based on CORBA or SunRPC.

TINE offers address 'redirection', a feature which can be used by any TINE server to offer devices or properties which physically reside on other TINE servers. This is an extremely useful feature which can be used by specific device servers, where clients can be blissfully unaware that individual devices might live on small embedded servers, or which can be used systematically, where, for instance, histories of properties might be kept on a completely different server. It is also the principle behind the Grouped Equipment Name Server (GENS) which manages "logical" device servers whose devices are spread among one or more physical servers.

TINE servers can also schedule 'events' when a property's value has changed, a feature which is sometimes used to eliminate latency as much as possible. For instance,

using the scheduling feature, and TINE multicast (Producer-Subscriber) ‘real-time’ video (not streaming!) can be delivered to as many clients as required.

4.9.3 TINE: User’s Perspective

A “Control System” is by definition a system for controlling a machine. The pure “user” of a control system expects to find all the tools he needs to both run and examine the machine. An automobile driver expects the steering wheel, clutch, and brakes to be in working order and to have diagnostic displays informing him of speed and fuel consumption. In the case of an accelerator (turn-key or not), a “pure” user likewise expects a set of easy-to-use-and-understand display applications, which seldom require his intervention and offer useful diagnostics should the need arise. This is particularly important with today’s light sources which generally operate on a 24/7 basis. The “pure” user might even be an operator contracted from an outside firm, who is trained in using the control applications only. So the quality of these end-user controls applications is of course important in itself. However, life is generally not so simple, as both hardware and software problems will inevitably arise, as will the need to fine-tune the machine. So there are many categories of “users” of the control system beyond the operators of the machine. In practice, the hardware engineers must be able to monitor their equipment and to be able to locate and repair problems. The software engineers will want to respond to feedback and make all necessary improvements to both front-end and end-user applications. The accelerator physicists will want to investigate and improve the machine operation.

Each of these control system users will have different programming and mathematical skills as well as a different perspective and understanding of the machine. The control system should provide an application programmer interface (API) for each of these skill levels and for each of these categories of users. Users with more extensive programming skills sometimes make use of the C, C++ or java TINE API for more elaborate applications. Those with more modest programming skills but with nonetheless the need of a high level language often use the Visual Basic API, offering a truly interpretive language, ideal for the part-time programmer.

4.9.3.1 Accelerator Physicists

Optics Server: Ring-wide or local transverse coupled and uncoupled orbit correction, slow transverse orbit feedback (<10Hz) and dispersion correction is an integral part of the controls application software. Local bump tuning or orbit correction and automatic orbit feedback corrections do not influence each other and can be applied independently.

A clear separation between the console application and the software package for standard online data analysis and beam optics calculations with precisely defined interfaces simplifies the development and maintenance of the programs. The console clients which might be a MATLAB client, Visual Basic client, java client, or whatever, are responsible for the collection of the data needed, for the parameter changes as well as error detection and handling.

The algorithms for the optical calculations are implemented as a server program which does not need to talk to any hardware devices but gets the data from the calling client, thus the programmer can concentrate on the solution of the scientific problem. The data transfer is done by the TINE protocol which allows the transfer of arbitrary data structures compatible with e.g. C-structures or FORTRAN common blocks. These

consist typically of a header structure with data naming the accelerator and its status as well as parameters for the requested calculation plus an array of structures describing the involved devices. The server has access to all data files relevant for describing the accelerator such as the lattice structure, magnet and monitor calibrations, and magnet circuits, and it delivers these on request to any client. A variety of algorithms is implemented for calculating orbit and dispersion corrections (such as singular value decomposition or MICADO), local bumps with different constraints, of the calculation of magnet settings for a given optic (or vice versa). Improvements and new functionality can be added at any time and are immediately available to all client programs within the control system.

MATLAB: For machine physics studies or complex measurements of beam optics parameters machine physicists often like to construct their own programs using interactive environments such as MATLAB. The TINE MATLAB API has been already widely used in the past to optimize the performance of the HERA storage rings. For PETRA3, the so-called MATLAB “middle layer package” will be used by the beam physicists to perform beam-based alignment procedures or to measure the orbit response matrix. The package will be integrated into the PETRA3 control system allowing direct access to measured orbit data or to corrector magnet set values.

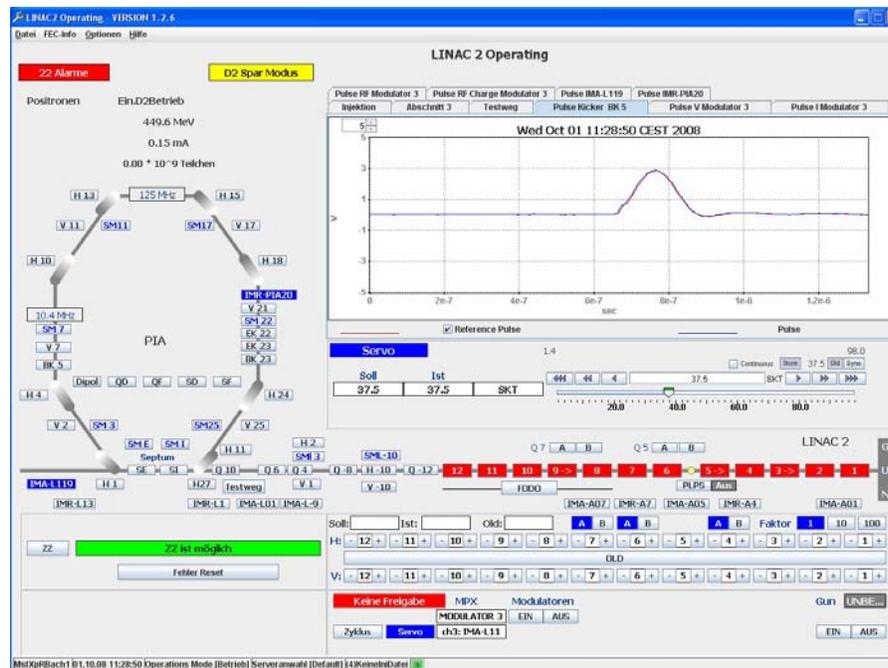


Figure 3: JAVA-based main operation panel for the pre-accelerator LINAC 2

4.9.3.2 Control Room Operators and Client Application Programmers

Client Applications: According to our experience, control room applications based on the rich-client model are best suited for providing optimal visualization and performance. Examples for rich client applications are complex orbit displays which

might combine position monitor readings and orbit correction tools or an archive data viewer with the option to correlate different data channels in various contexts. However, the operators are also free to create their own display panels, by using simple-client display frameworks such as jDDD, Control System Studio, or COMA [2].

Various Java server and client applications provide access to device control and support accelerator operation. Laymen and expert views and web links to help documents are integral functions of each application. The status of all permanent client-to-server connections of an application is routinely supervised and link errors are notified to the operators. For example, Fig. 3 shows the complex control room panel for operating LINAC2, summarizing all relevant information and actions for the operators. Other applications include live video image streams from optical diagnostic devices.

ACOP Widget Toolbox: Framework classes for client and server applications are provided to ensure design conformity and to handle initialization data. To facilitate coding, the ACOP (Accelerator Component Oriented Programming) toolbox [3] is used for simple data access and rendition. The original ACOP components were ActiveX controls, suitable only for programming in Windows. Recently, ACOP components are now available as java Swing components following the Java beans standard. Besides ACOP chart, a slider, wheel switch, dial knob, animated label, gauge, and video component have all been implemented. All ACOP components support design-time and run-time customization such as adding a channel to a chart or dragging and dropping channel metadata to another application instance. .NET ACOP components are planned but not yet realized.

Alarm Viewer: An alarm viewer panel (Fig. 4) informs the operator about critical device states. All alarms are processed and archived by the central alarm system which is an associated service of the TINE control system. The alarms are sorted by various categories and rated according to a predefined alarm severity scheme. Both open and closed alarms can be displayed simultaneously if requested. Alarm histories and detailed metadata are likewise available upon request.

Measurements: All measured signals from bunch current monitors, pulsed RF devices and injection or ejection components etc. are digitized at the front-end level and available as scope-like traces through the control system (Fig. 5). The capability of the control system to generate, distribute and visualize read-back values with a repetition rate of $6\frac{1}{4}$ Hz synchronously with the DESY2 accelerator cycle has been successfully proven.



Figure 4: Alarm viewer

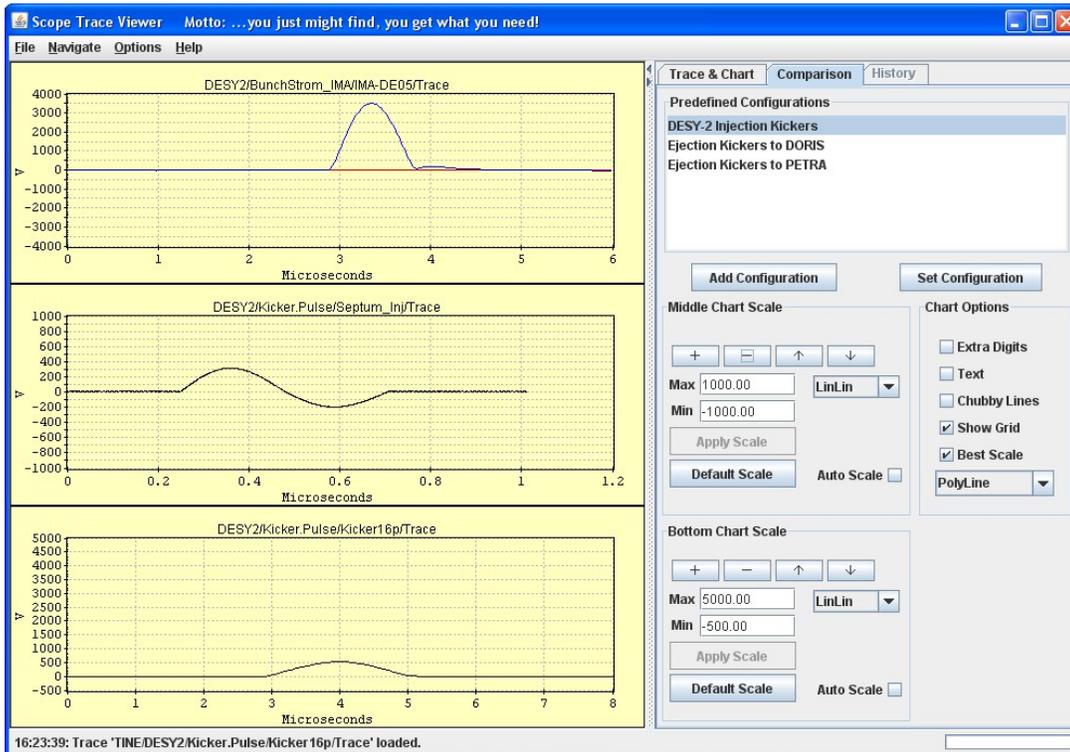


Figure 5: Generic viewer for scope-like traces

LabView and Other Laboratory Automation Workbenches: Many hardware engineers are adept at using laboratory automation workbenches such as LabView or AgilentVee. To this end, TINE provides an API for both LabView and AgilentVee, allowing the choice of the programming language that is best suited for the engineering task to be done. Control and data acquisition logic of the PETRA3 RF system is completely implemented with LabView. LabView is also successfully used for experiment automation and beam diagnostics.

Oscilloscopes and Other Test and Measurement Instruments: Specific stand-alone and off-the-shelf test and measurement instruments such as oscilloscopes or spectrum analysers do not in general fit seamlessly into control system architectures. Proper instrument integration often imposes an undue burden on the application developer. In addition to instrument specific drivers and to simplify this task, we use IVI-foundation compliant instrument drivers. The Interchangeable Virtual Instrument (IVI) standard defines types of instruments and interfaces to generic virtual instrument drivers in order to avoid vendor-specific incompatibilities. In addition, the VISA (Virtual Instrument Software Architecture) standard is used to become data bus independent. Generic LabView virtual instrument applications for oscilloscopes and digital multimeters are available.

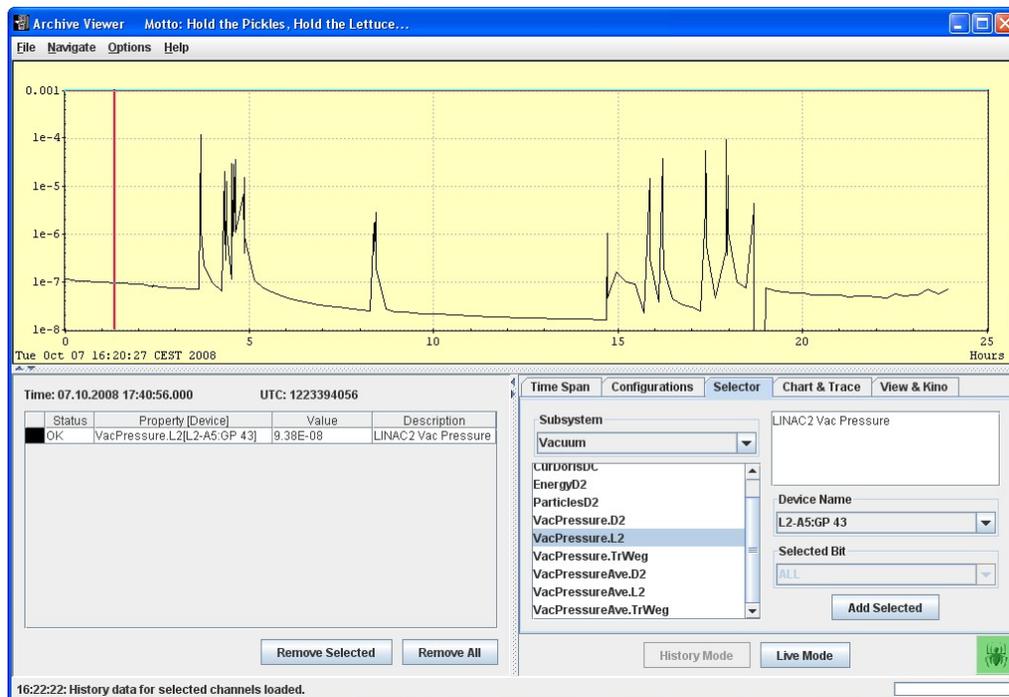


Figure 6: Archive viewer for history like machine parameter

4.9.3.3 Accelerator Coordinators

Archive Viewer: Accelerator coordinators or management personnel ask for information on operation performance and/or data for off-line analysis. Operation overview panels and statistical performance data are provided. Based on the integrated

archive systems, accelerator and beam parameter can be retrieved in various ways as histograms, histories (Fig. 6) or snapshots.

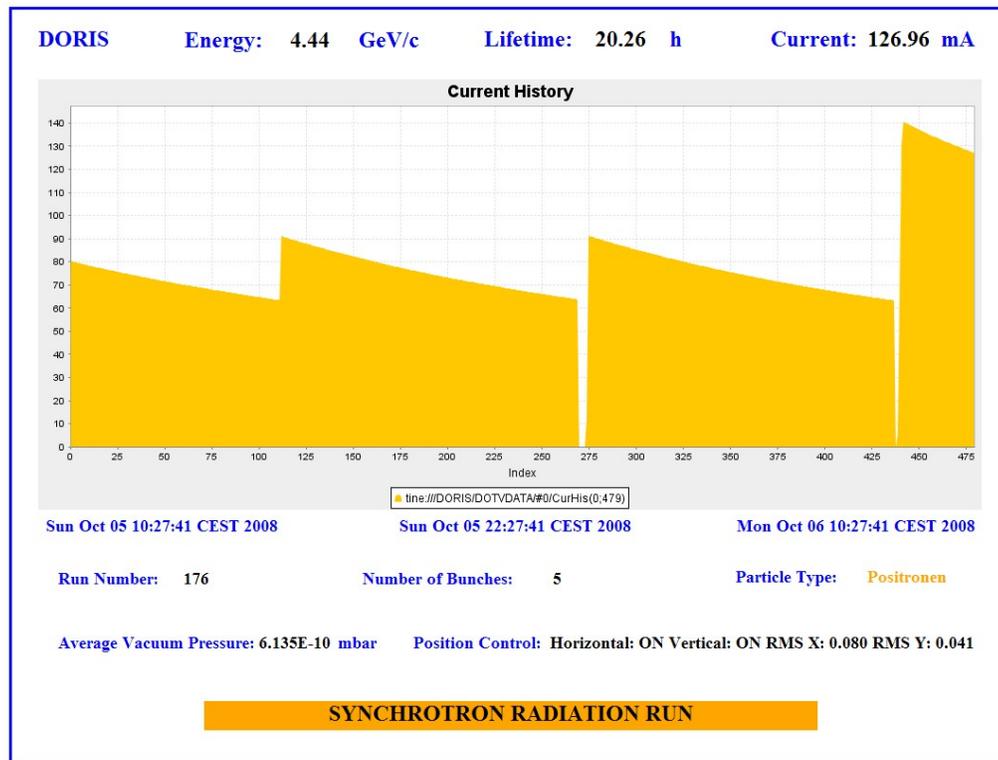


Figure 7: Web2c synoptic display page assembled by Web2cHisto, Web2cLabel, Web2cText, Web2cValue, Web2cTime and Web2cSection elements.

4.9.3.4 TINE: Remote Access with the Web2c Toolkit

The Web2c toolkit [4] is a collection of web-based tools designed to configure and visualize customer-specific client applications for accelerator control. It consists of a synoptic display, a corresponding graphical editor and a specific viewer for retrieving and rendering archived accelerator parameters.

The Web2c applications run as server-side JAVA applications and are published as dynamic web pages to the clients. Web2c pages contain only HTML tags and JavaScript snippets. The usage of the Web2c toolkit requires no specific programming skills by the users. By design, Web2c-based pages are accessible from every valid network address if not otherwise restricted. User authentication is based on a user name and a user password (encrypted), and each user has to be registered at the web server prior to login and page downloading. Web2c supports three different user roles (layman, expert and supervisor).

Web2c applications are not bound to any specific control system implementation. Presently, a plug for TINE-based control systems has been implemented. Through additional gateway processes integrated in TINE, seamless access to DOOCS-based, TANGO-based and EPICS-based control systems is feasible. However, native plugs for those control systems are not yet provided.

Web2c pages are simple client applications. The pages are generated from pre-configured, individual and user-application specific configuration files, describing the attributes and methods of the widgets embedded in the requested page. The typical use cases are overview or information panels, service panels for remote maintenance or user forms to access stored data for off-line analysis. The Web2c toolkit has been developed within the scope of the PETRA3 controls project at DESY. It is used both for accelerator controls and to control equipment of synchrotron light beam lines at DORIS and PETRA3.

4.9.3.5 Synoptic Display

The Web2c synoptic display is a frame for simple controls applications. It provides web pages with active and passive widgets to visualize and control accelerator equipment. Examples are labels, value lists, animated images, histogram charts etc. Each synoptic display page updates periodically with a rate as specified in the page's configuration file. Short histories of displayed read-back values are available on mouse click. Figure 7 shows a sample page.



Figure 8: Web2c archive viewer page displaying sample data

Archive Viewer: The archive viewer is a web-based form to request data from a control system archive and to display the retrieved data as a plot chart or a table. The user can select individual data channels or groups of data channels as listed in the archive viewer configuration file. Besides the pre-configured “last 12h”, “last 24h” and “last 48h” options, each time interval can be selected as required. On-line zooming capability via mouse click is provided retrieving fresh data for the region of interest from the archive system. Figure 8 shows the archive viewer page displaying sample data.

4.9.4 References

1. P. Duval, <http://tine.desy.de>
2. R. Bacher, P. Bartkiewicz, P. Duval, The Run-time Customization of Java Rich Clients

with the Coma Class, ICALEPCS 2007, Knoxville, USA

3. P. Duval, H.G. Wu (DESY) and I. Kriznar (Cosylab), The ACOP Family of Beans, ICALEPCS 2007, Knoxville, USA
4. R. Bacher, <http://web2ctoolkit.desy.de>

4.9.5 DOOCS - TINE: A Common Future

The DESY accelerators are controlled and supervised by two complete control systems, DOOCS and TINE. On the one hand, both systems are driven by specific requirements of the accelerator for which they were developed and have their own specific, unique features. On the other hand, both systems are quite similar. It is envisaged to merge the two systems in a way that they can profit best from each other. As an initial step the two different communication systems were consolidated with regard to data types and with the protocol integration of TINE in the DOOCS clients and servers. With this prerequisite the two control systems will become fully transparent in the next step. It does not matter whether a DOOCS client talks to a TINE server or vice versa. Rich-client displays and synoptic displays based on jDDD have the potential to complement each other. If applicable, services such as alarm or archive systems for slow varying accelerator parameters will be able to communicate without restriction, while the DOOCS DAQ system is unique for fast data acquisition tasks. Automation and beam physics applications are further good candidates for a merged approach. First implementations steps are under way within the context of the PETRA3 project.

4.10 Future Trends

Matthias Clausen and Phil Duval
DESY, Notkestrasse 85, D-22607 Hamburg, Germany
Mail to: Matthias.Clausen@desy.de

4.10.1 Introduction

Each accelerator is special and a one of a kind. Therefore a special control system is needed to run this 'unique' equipment. Following this argument many different control systems have been developed with no interest in sharing developments between them. International conferences like the ICALEPCS opened the floor to share experience and to find partners for future developments. Collaborations such as the EPICS collaboration started from here. On the other hand partner institutions started to share their implementations. This way the ISOLDE project from CERN found its partner implementation at DESY and the TANGO collaboration began. Java and scripting languages made their way into the control room. Open software interfaces are key for software sharing. Control applications are migrating from front end computers to front end controllers and even into the controlled equipment itself. The front end control shelf has been VME for decades. What will be the 'shelf of the next decade'? This and other questions of future control systems will be discussed in this article.

4.10.2 Control System Design

In the 1980s it was an important argument for modern control systems to be of the DCS type. Where DCS stands for distributed control system and basically defines that the I/O is not directly connected to the operator console but rather remotely connected to distribute front end controllers.

In today's control systems it is not a question any more whether a system is of the DCS type but rather how many layers are used between the operator interface and the I/O in the field.

4.10.2.1 *The Three Tier Architecture*

Regardless how many layers are actually implemented between the operator console and the remote I/O the architecture is in most cases defined as a *three-tier architecture*. This has basically become a synonym for a distributed control system (DCS). In the future this will truly change into multi layer architectures. The demands for richer functionalities on the network level will require additional service layers and services to provide the required functionality or to handle the huge amount of front-end controllers in future installations.

4.10.3 Front-end Controllers

VME crates are still dominating the majority of the installed front-end controllers. This also defines the type of operating system and control system software which can be run on the hardware. This uniformity has changed in today's control systems. There are no restrictions for control hardware and the software can run on any CPU and made it even into FPGAs.

4.10.3.1 *Software*

Most control systems can be distinguished by the software implementation running on the front-end controllers. A wide range from direct I/O access, channel based implementations, component approaches to control object class libraries is available. There is a trade off between complexity-hiding into components or control objects and the ability to share implementations in different installations which leads towards more generic approaches like channel based implementations. A combination of channel-based portability with specialized component-based control devices seems to be the future trend.

4.10.3.2 *Hardware*

Control software has made its way into all levels of control hardware. Power supplies are no longer controlled by a front-end controller. The controller software itself is running in the power supply. The same applies for LLRF controls where the control software is running deeply embedded in the local FPGA.

Dedicated control hardware has also been available for several years from industry. It is used to precisely read beam position monitors, to distribute timing information or to provide general purpose I/O controllers. What is new in this area is the fact that customers can not only purchase the hardware, they can also order the hardware with

‘their’ control system software installed and tested, as in many cases the drivers for the hardware are available – at least for the widely installed control systems.

The latest developments in front-end hardware show a trend towards using ATCA and/or microTCA crates. This new standard from the telcom industry comes with a wide variety of new features. High serial throughput in the backplane and the mandatory shelf management are two of the main arguments for using the more expensive TCA standard instead of the well established VME or compact-PCI (another popular choice). Due to the fact that TCA has been designed for the telcom business there are no I/O boards available which are commonly used in accelerator controls. This will certainly change over time. Some of the next big projects will indeed begin using the TCA technology. Is this the new standard after several decades of VME installation?

4.10.4 Networking

The Ethernet is today’s control system’s workhorse on the network layer. Except precise timing on the pico-second level the available Ethernet speed was and is always one order of magnitude higher than the demanded bandwidth in control systems. This trend will probably continue in the future.

The basic protocol on the Ethernet is uniformly the Internet Protocol (IP) with its two variants TCP and UDP. All control system protocols including commercially available protocols are based on IP.

4.10.4.1 Control System Protocols

Front-end software and control system protocols always go together. The dependencies of these two are too tightly bound to separate them. Thus it has been nearly impossible to discuss control protocols without discussing the control system as a whole. New trends in control system collaborations and discussions during recent conferences and workshops indicate that there is growing interest to agree on a common control protocol. One of the potential candidates for such an approach is the TINE protocol designed at DESY. TINE provides the necessary functionality to gather many control systems under this umbrella.

4.10.4.2 The TINE Control System Protocol

Originally a spin-off of the ISOLDE control system, TINE is now itself a mature control system, where a great deal of developmental effort has gone specifically into the control system protocol, offering a multi-faceted and flexible API with many alternatives for solving data flow problems. As the TINE kernel is written in straight C and based on Berkeley sockets, it has been ported to most available operating systems. A notable exception is java TINE, which is written in native java. Other platforms such as LabView, MatLab, .NET, etc. all make use of interoperability interfaces with the standard TINE kernel.

When we speak of a control system “protocol” we are generally referring to the methodology behind passing data and issuing instructions among the three “tiers” alluded to above. There are, of course, many ways to do this, and we should consider both data transfer and data types when we discuss them.

Synchronous “Client-Server” transactions and asynchronous “Publisher-Subscriber” monitors have become standard fare for all modern control systems. In addition to these

well-known data-exchange mechanisms, TINE also offers, as a way to further save on network bandwidth in large systems, a “Producer-Consumer” and a “Publisher-Consumer” data exchange. These might require a few words of explanation. In pure “Producer-Consumer” mode, a server can simply supply the control system with relevant data via multicast. Here, “relevant data” might be the machine state, beam energy, beam current, etc. In this case, there is no “connection table” and the server is simply “producing” the data whether or not there are any “consumers”. “Publisher-Consumer” mode is nominally equivalent to “Publisher-Subscriber”, with the notable exception that the “subscriber” has requested that the monitored data be sent via multicast and consequently has changed his role to an anonymous consumer. This turns out to be an ideal way, for instance, of sending video frames from a single server to multiple clients.

TINE has long been a “bottom-up” control system, where much of the functionality found in the kernel stems from the needs and desires of the control system developers. This is particularly true concerning data types. Besides allowing control system data to be sent as any of the standard “primitive” data types (or arrays thereof) common to all high-level languages, TINE also support numerous composite data types, such as value-status pairs, or name-value-status triplets. In addition, a highly prized feature of TINE is the ability of the developer to define and register his own “data type” structure.

Worth mentioning at this juncture is the TINE kernel Applications Programmer Interface (API). The TINE API uses a narrow interface and refers to “links,” in loose analogy with Dynamic Data Exchange, but in the end more reminiscent of remote procedure “calls”. These “calls” can easily collapse to the more traditional “get”, “set”, and “monitor” methods frequently offered in control system APIs, but nonetheless offer the ability to issue atomic “set/get” operations and to send completely different data types to a server as those being received. While this latter point is “unusual” it is nonetheless sometimes very useful. For instance, the TINE orbit server is in this sense nothing more than a server in the control system, but if sent the current optics and orbit and desired orbit (user-defined data type) it can return the new optics (different user-defined data type).

4.10.5 Operator Tools

In general one may state that two different types of operator tools have been developed of the last decades: Generic applications written in ‘C’ or ‘C++’ running on Unix type operating systems and implementing X-Window type graphics on one side and more specific applications written in Visual Basic running on Windows operating systems on the other side. Interestingly enough - there is a trend for both types to be converted or rewritten in Java. The pluggable approach of eclipse based applications combines the two types. Generic operator tools which typically consist of a configuration (or edit) part and a runtime part can be implemented this way as well as custom applications. The overhead of writing individual applications can be reduced by applying plugin APIs and making use of the plugin ecosystem. One candidate following this paradigm is the Control System Studio (CSS) which is based on OSGi and eclipse. CSS applications (or plugins) are accessing the control system through a generic Data Access layer (DAL). Applications written for one control system will also work in another control system environment – or mixed infrastructures. The same applies for the access to archive data or message based alarm systems. The core part of CSS also

provides interfaces and sample implementations for authentication, authorization and logging.

The trend to define open interfaces to access different control systems can already be found in several implementations. In conjunction with a pluggable infrastructure it eases collaborative software developments in the future.

4.10.6 High Level Applications

The users of a control system will largely consist of the operators, the machine physicists, the hardware engineers, and the software engineers. Each of these will have different programming and mathematical skills as well as a different perspective and understanding of the machine. The control system should provide an API for each of these skill levels and for each of these categories of users. Users with more extensive programming skills might want to program directly in C, C++ or java or C#, even at the client GUI level. Those with more modest programming skills but with nonetheless the need of a high level language for writing rich clients might be more comfortable with something along the lines of Visual Basic, Qt, or Lazarus. Finally, those with little programming experience (or time to program) might want to make simple client GUI panels, where applications are “configured” more or less on-the-fly.

It is tempting to claim that most machines physicists gravitate towards mathematically oriented packages such as MatLab or ROOT, the hardware engineers primarily use LabView, and the operators will put together application panels using whatever panel builder they have been given. While this may be true in general, there is frequently a good deal of “cross-over” (owing to a individual’s programming skill and motivation), so that such blanket statements do not categorically apply. In any case, expedients for high productivity must include a good API and control-system aware displayer widgets, components, or “beans.” Panel-building frameworks such as the Synoptic Display in Control System Studio(CSS) or jDDD now offer rather advanced (and control-system independent) features.

Recently, much effort has gone into providing common operational procedures within high level applications frameworks. This might include orbit correction, on-line modeling, sequencing, etc. Currently there are initiatives using e.g. MatLab and XAL. Porting applications from one site to another is still a major issue even though it’s been written with portability in mind.

4.10.7 Collaborations and Institutions

The likelihood that a new control system will be designed from scratch is under the given budget constraints of current projects very low. If not using an existing custom control package the tendency is to pick an existing implementation from the two control software collaborations EPICS and TANGO. So far there has been no impetus for the two collaborations to work together. But a driving force is now coming from the user base. The experiments on beam-lines of current and future synchrotron light sources are coming from institutions from all over the world. The technology used may vary from day to day. Therefore the local organization must make sure that it can deal with this situation. Openness is the key word. Integration the two key players in to each others control system is part of the story.

Common approaches on the application level and on the network level will ease this overlapping collaboration – which is on the way.

4.10.8 Globalization

The next big projects like ITER and ILC will no longer be installed by a single institution. Many heterogeneous control components will be delivered as in kind contributions from project partners. Remote operation for commissioning and finally for remote machine controls of the whole complex will be required. An agreement on common technologies will be necessary to keep the experts remote while the expert's equipment is controlling the machine far away. Such a scenario has been described during the last ICFA workshop [1].

4.10.9 Software Sharing

The desire to share control system software has been expressed for over twenty years. The different philosophies and technical implementations – which manifest itself in the various control system protocols – were sometimes given as good reasons why software sharing is technically not possible. An agreement on common protocols (like TINE++) and common interfaces (like DAL) in conjunction with pluggable software components will open a new regime of software sharing. The future will show whether the given chances will be well accepted or discussed away like in the past.

4.10.10 Conclusions

Financial restrictions and limited manpower will drive a concentration process in the area of control system software. It is not desirable to follow the idea of one or two types of implementations. There are and will be good reasons to keep a greater number of control systems implementations. But there must be ways to share knowledge and implementations between the systems. Beamline instrumentation and new big projects will be the driving force for common approaches like common protocols and open interfaces. This will open the door to 'choose from the best of all' and to collaborate on future developments.

Ideally, any tool or control system element (a GUI application, say, or a front end controller) from a "foreign" control system could be integrated into a given control system implementation with only minor adjustments to the "view" (e.g. changes in a database or naming convention). This would be much a long the lines of an operating system supporting any number of hardware "peripherals" or applications, provided certain "ground rules" are met.

4.10.11 References

1. M. Clausen, Global collaboration on accelerator operations and experiments, 9th ICFA Seminar, SLAC, http://www-conf.slac.stanford.edu/icfa2008/Clausen_102908.pdf

5 Forthcoming Beam Dynamics Events

5.1 45th ICFA Advanced Beam Dynamics Workshop: Energy Recovery Linacs (ERL09)

Georg H. Hoffstaetter, Cornell University
Mail to: Georg.Hoffstaetter@cornell.edu

This workshop (<http://www.lepp.cornell.edu/Events/ERL09/>) will take place at Cornell University (USA), from Monday June 8 through Friday June 12, 2009.

This workshop is the third international ERL workshop, following ERL05 in TJNAF and ERL07 in Daresbury. It will address fundamental challenges related to the generation of high brightness and simultaneously high average current electron beams with Energy Recovery Linacs, including high brightness electron guns, orbit stability, beam-quality preservation during acceleration and energy recovery. Specific subjects of interest are:

- * Design and development of high average current, low emittance, polarized and unpolarized injectors
- * Beam halo formation and control of beam loss
- * Ion accumulation in high current CW beams
- * Optimized lattice design and start-to-end simulation
- * Beam stability and multibunch, multipass instabilities
- * Superconducting RF systems for CW, high-current applications
- * Higher order mode damping
- * RF control and stability under very large loaded Q
- * Synchronization challenges
- * Impedance modeling and control
- * Gas and Intra Beam Scattering
- * Shielding and component protection for high power beams
- * High current, low emittance, and short pulse diagnostic techniques.

Organizing Committee: Georg Hoffstaetter (chair), Ilan Ben-Zvi, Bruce Dunham, Rodney Gerig, Ryoichi Hajima, Geoffrey Krafft, Mike Poole

5.2 ICFA Mini-Workshop: Phenomenology Workshop on Advanced QED methods for Future Accelerators

Ian Bailey, Cockcroft Institute, Lancaster, UK
Mail to: ian.bailey@stfc.ac.uk

With the advent of high-luminosity electron-positron accelerators working at the TeV energy scale and with the resulting extreme environments expected during bunch-bunch collisions, urgent further attention must be given to the impact of quantum effects on the design of such machines.

In particular, formalisms for describing radiative processes and the methods of calculation in the future strong-field environments must be reviewed and the boundaries of validity of higher order perturbative calculations determined.

Thus the central aim of the workshop is to make progress on the development of exact theoretical methods for evaluating beam induced effects (beamstrahlung, bremsstrahlung, etc.) and spin dynamics for the beam conditions at TeV colliders.

Methods of calculation to be reviewed include, but are not limited to, inclusion of the fields at Lagrangian level, the “Operator” method for ultra-relativistic particles and QED calculations using light-cone coordinates.

Reports on the status of the models implemented in existing simulation codes will be given, and discussions about ways to improve these models and estimate the remaining theoretical uncertainties are planned.

Planned dates: 3-4 March, 2009

Location: Cockcroft Institute, Daresbury, UK

Organisers: Gudrid Moortgat-Pick, Cockcroft Institute / IPPP / Durham, Ian Bailey, Cockcroft Institute / Lancaster, Antony Hartin, DESY, Swapan Chattopadhyay, Cockcroft Institute

5.3 ICFA Mini-Workshop on Laser-Assisted H^- Beam Stripping

Stuart Henderson, SNS, ORNL
Mail to: shenderson@ornl.gov

This mini-workshop will focus on the physics and technology of laser-assisted stripping of H^- beams for use in high-power proton accelerators. Recent theoretical and experimental work suggests that lasers may be employed in the stripping of H^- beams that is required for multi-turn charge-exchange injection in high power proton facilities. Laser-stripping methods hold the promise of eliminating limitations associated with conventional stripping foils, namely short foil lifetime at high beam powers, radio-activation of nearby accelerator components, beam loss associated with multiple foil traversals, and complications associated with foil inefficiency. This workshop will bring together experts in laser-particle beam interactions, laser-stripping modeling, high power lasers and laser beam optics.

- Dates: February 18-19, 2009
- Venue: ORNL
- Hosted by: Spallation Neutron Source, Oak Ridge National Laboratory
- Contact person: Stuart Henderson (shenderson@ornl.gov)
- Attendance is by invitation only.

6 Announcements of the Beam Dynamics Panel

6.1 ICFA Beam Dynamics Newsletter

6.1.1 Aim of the Newsletter

The ICFA Beam Dynamics Newsletter is intended as a channel for describing unsolved problems and highlighting important ongoing works, and not as a substitute for journal articles and conference proceedings that usually describe completed work. It is published by the ICFA Beam Dynamics Panel, one of whose missions is to encourage international collaboration in beam dynamics.

Normally it is published every April, August and December. The deadlines are 15 March, 15 July and 15 November, respectively.

Categories of Articles

The categories of articles in the newsletter are the following:

1. Announcements from the panel.
2. Reports of beam dynamics activity of a group.
3. Reports on workshops, meetings and other events related to beam dynamics.
4. Announcements of future beam dynamics-related international workshops and meetings.
5. Those who want to use newsletter to announce their workshops are welcome to do so. Articles should typically fit within half a page and include descriptions of the subject, date, place, Web site and other contact information.
6. Review of beam dynamics problems: This is a place to bring attention to unsolved problems and should not be used to report completed work. Clear and short highlights on the problem are encouraged.
7. Letters to the editor: a forum open to everyone. Anybody can express his/her opinion on the beam dynamics and related activities, by sending it to one of the editors. The editors reserve the right to reject contributions they judge to be inappropriate, although they have rarely had cause to do so.

The editors may request an article following a recommendation by panel members.

However anyone who wishes to submit an article is strongly encouraged to contact any Beam Dynamics Panel member before starting to write.

6.1.2 How to Prepare a Manuscript

Before starting to write, authors should download the template in Microsoft Word format from the Beam Dynamics Panel web site:

<http://www-bd.fnal.gov/icfabd/news.html>

It will be much easier to guarantee acceptance of the article if the template is used and the instructions included in it are respected. The template and instructions are expected to evolve with time so please make sure always to use the latest versions.

The final Microsoft Word file should be sent to one of the editors, preferably the issue editor, by email.

The editors regret that LaTeX files can no longer be accepted: a majority of contributors now prefer Word and we simply do not have the resources to make the conversions that would be needed. Contributions received in LaTeX will now be returned to the authors for re-formatting.

In cases where an article is composed entirely of straightforward prose (no equations, figures, tables, special symbols, etc.) contributions received in the form of plain text files may be accepted at the discretion of the issue editor.

Each article should include the title, authors' names, affiliations and e-mail addresses.

6.1.3 Distribution

A complete archive of issues of this newsletter from 1995 to the latest issue is available at

<http://icfa-usa.jlab.org/archive/newsletter.shtml>.

This is now intended as the primary method of distribution of the newsletter.

Readers are encouraged to sign-up for electronic mailing list to ensure that they will hear immediately when a new issue is published.

The Panel's Web site provides access to the Newsletters, information about future and past workshops, and other information useful to accelerator physicists. There are links to pages of information of local interest for each of the three ICFA areas.

Printed copies of the ICFA Beam Dynamics Newsletters are also distributed (generally some time after the Web edition appears) through the following distributors:

Weiren Chou	chou@fnal.gov	North and South Americas
Rainer Wanzenberg	rainer.wanzenberg@desy.de	Europe ⁺⁺ and Africa
Susumu Kamada	susumu.kamada@kek.jp	Asia ^{**} and Pacific

⁺⁺ Including former Soviet Union.

^{**} For Mainland China, Jiu-Qing Wang (wangjq@mail.ihep.ac.cn) takes care of the distribution

with Ms. Su Ping, Secretariat of PASC, P.O. Box 918, Beijing 100039, China.

To keep costs down (remember that the Panel has no budget of its own) readers are encouraged to use the Web as much as possible. In particular, if you receive a paper copy that you no longer require, please inform the appropriate distributor.

6.1.4 Regular Correspondents

The Beam Dynamics Newsletter particularly encourages contributions from smaller institutions and countries where the accelerator physics community is small. Since it is impossible for the editors and panel members to survey all beam dynamics activity worldwide, we have some Regular Correspondents. They are expected to find interesting activities and appropriate persons to report them and/or report them by themselves. We hope that we will have a “compact and complete” list covering all over the world eventually. The present Regular Correspondents are as follows:

Liu Lin	Liu@ns.lnls.br	LNLS, Brazil
Sameen Ahmed Khan	Rohelakan@yahoo.com	SCOT, Oman
Jacob Rodnizki	Jacob.Rodnizki@gmail.com	Soreq NRC, Israel
Rohan Dowd	Rohan.Dowd@synchrotron.org.au	Australian Synchrotron

We are calling for more volunteers as Regular Correspondents.

6.2 ICFA Beam Dynamics Panel Members

Name	eMail	Institution
Marica Biagini	marica.biagini@lnf.infn.it	LNF-INFN, Via E. Fermi 40, Frascati 00044, Italy
Yunhai Cai	yunhai@slac.stanford.edu	SLAC, 2575 Sand Hill Road, MS 26 Menlo Park, CA 94025, U.S.A.
Swapan Chattopadhyay	swapan@dl.ac.uk	The Cockcroft Institute, Daresbury Laboratory, Daresbury, Warrington WA4 4AD, U.K.
Weiren Chou (Chair)	chou@fnal.gov	Fermilab, MS 220, P.O. Box 500, Batavia, IL 60510, U.S.A.
Yoshihiro Funakoshi	yoshihiro.funakoshi@kek.jp	KEK, 1-1 Oho, Tsukuba-shi, Ibaraki-ken, 305-0801, Japan
Miguel Furman	mfurman@lbl.gov	Center for Beam Physics, LBL, Building 71, R0259, 1 Cyclotron Road, Berkeley, CA 94720-8211, U.S.A.
Jie Gao	gaoj@ihep.ac.cn	Institute for High Energy Physics, P.O. Box 918, Beijing 100039, China
Ajay Ghodke	ghodke@cat.ernet.in	RRCAT, ADL Bldg. Indore, Madhya Pradesh, India 452 013
Ingo Hofmann	i.hofmann@gsi.de	High Current Beam Physics, GSI Darmstadt, Planckstr. 1, 64291 Darmstadt, Germany
Sergei Ivanov	ivanov_s@mx.ihep.su	Institute for High Energy Physics, Protvino, Moscow Region, 142281 Russia
Kwang-Je Kim	kwangje@aps.anl.gov	Argonne Nat'l Lab, Advanced Photon Source, 9700 S. Cass Avenue, Argonne, IL 60439, U.S.A.
In Soo Ko	isko@postech.ac.kr	Pohang Accelerator Lab, San 31, Hyoja-Dong, Pohang 790-784, South Korea
Alessandra Lombardi	Alessandra.Lombardi@cern.ch	CERN, CH-1211, Geneva 23, Switzerland
Yoshiharu Mori	mori@kl.rri.kyoto-u.ac.jp	Research Reactor Inst., Kyoto Univ. Kumatori, Osaka, 590-0494, Japan
Chris Prior	c.r.prior@rl.ac.uk	ASTeC Intense Beams Group, STFC RAL, Chilton, Didcot, Oxon OX11 0QX, U.K.
Yuri Shatunov	Yu.M.Shatunov@inp.nsk.su	Acad. Lavrentiev, prospect 11, 630090 Novosibirsk, Russia
Junji Urakawa	junji.urakawa@kek.jp	KEK, 1-1 Oho, Tsukuba-shi, Ibaraki-ken, 305-0801, Japan
Jiu-Qing Wang	wangjq@mail.ihep.av.cn	Institute for High Energy Physics, P.O. Box 918, 9-1, Beijing 100039, China
Rainer Wanzenberg	rainer.wanzenberg@desy.de	DESY, Notkestrasse 85, 22603 Hamburg, Germany
Jie Wei	wei1@bnl.gov	Institute for High Energy Physics, P.O. Box 918, 9-1, Beijing 100039, China

*The views expressed in this newsletter do not necessarily coincide with those of the editors.
The individual authors are responsible for their text.*