CERN COMPUTER NEWSLETTER

CERN - IT Division
CH-1211 Geneve 23
SWITZERLAND

January – March 2001

CERN-CNL-2001-001
Vol. XXXVI, issue no 1

35th Anniversary Issue

Contents

Editorial Information

Editorial Board

General Information

The CNL Web home page is at URL: `http://cern.ch/ref/cnl`

The policy is to have approximatively 4 CNL editions per year (Spring, before and after the Summer, and before Christmas shutdown).

Issue number	Deadline
Summer edition	Friday 1st of June 2001.

Information for contributors

The CNL is processed using XML technology and templates. Articles marked up in HTML are preferred, but plain text is also acceptable. A Web form to be used for article submission (contributions) is provided at the CNL "Home" page (`http://cern.ch/ref/cnl`), link "Article Submission form" (`http://cern.ch/ref/cnl/submit`).

Help in using this form is provided via the "Help" link (`http://cern.ch/ref/cnl/help/submit`) at the top bar.

Distribution

- An electronic mail is sent to the "CNL Announcement Distribution List" (`http://cern.ch/ref/cnl/help/announcement`) as soon as a new issue of the CNL is available on the Web.
- A paper version is sent to the users who have subscribed to the "CNL Paper Version Registration" (`http://cern.ch/ref/cnl/help/register`) list.

All *subscriptions to our distribution lists* can be made through the CNL "Help&Info" page at the URL `http://cern.ch/ref/cnl/help`.

You can also cancel your subscription in either case by filling in the appropriate form.

The compressed *PostScript file* containing the complete printable version of this CNL is available via the Web at the URL `http://cern.ch/ref/cnl-2001-001/main.ps.gz`. It is possible that, in order to optimize typesetting, the PostScript printable version differs slightly from the master HTML version. In any case the HTML version on the Web must be considered as the reference.

If you need Help

FRONT LINE Services

	Phone	Location	E-mail Address
Computing Help Desk	**78888**	**513/R-052**	`Helpdesk@cern.ch`
Opening hours: 8:30–17:30 (Monday–Friday)			
Satellite Help Desk	**78888**	**52/1-004**	`Helpdesk@cern.ch`
Opening hours: 8:30–12:30 (Monday–Friday)			
Administrative Applications (EDH, BHT, etc.)	**79933**	**5/R-021**	`Ais.Support@cern.ch`
Opening hours: 8.30–12:30 and 13:30–17:30			
Central Computer Operators (24/24 hours on call)	**75011**	**513/R-049**	`Computer.Operations@cern.ch`
General Network Operational Problems	**74927**		`Network.Support@cern.ch`

Contacts at CERN

The "Computing Help Desk" is the recommended central point for help on all computer-related issues. Following is a non-exhaustive list of individual people responsible or assigned to a given service. If you do not find a pointer to the service you need, then contact the "Computing Help Desk" for submitting your request. For all the services listed below please use the generic electronic mail address whenever possible. These mailboxes will be read even when the usual specialist is absent. It is helpful to use relevant keywords in the subject field of your message. Contact specialists directly only for very urgent cases or for very general consultations. Please note:

- In the *Web version* of this page the "Service Definition" can be a *link* to the Welcome page of this service: we invite you to read carefully all the information given in this page before submitting a question/problem.
- In addition you might find useful to search in the User Support "Questions&Answers" database for possible answers to your query (`http://consult.cern.ch/qas`).
- Many generic e-mail addresses for support lines are just an automatic re-direction to the Computing Helpdesk (this is the case, for instance, for `Mail.Support` or `Www.Support`). Those addresses have been kept mainly for backward compatibility, and also to help the Helpdesk to escalate the query towards the right specialist whenever this is needed.

USER SUPPORT

Service Definition	Name	E-mail Address
User Relations and Service Manager	M. Marquina	`user.support@cern.ch`
Help Desk Manager	R. Woolnough	`user.support@cern.ch`
User Registration	M.C. Perler	`user.registration@cern.ch`
Computing BookShop	J. Megies	`bookshop@cern.ch`

CENTRAL SERVICES

Public Services (PLUS batch & interactive services and CSF/PCSF Simulation farms)
See the complete and up-to-date list of contacts at URL:
`http://cern.ch/it/pdp/is/main/experiments.html#PUBSERV`

SHIFT (Reconstruction & Analysis Clusters)	T. Smith	`tim.smith@cern.ch`
CORE Operations	M. Vergari	`mario.vergari@cern.ch`
Work Group Servers	T. Smith	`tim.smith@cern.ch`
Distributed File Services (incl. AFS)		`afs.support@cern.ch`
NAP and Parallel Applications	E. Mcintosh	`eric.mcintosh@cern.ch`
Tape Allocations and Vault Space	R. Minchin	`richard.minchin@cern.ch`
Stager, tapes and TMS support		`tape.support@cern.ch`
Central Data Recording (CDR)		`cdr.support@cern.ch`
CASTOR HSM software		`castor.support@cern.ch`
HPSS Services		`Hpss.Support@cern.ch`
CERN-wide Print Service	J.L. Vosdey	`printer.support@cern.ch`

Support for Experiments
See the complete and up-to-date list of contacts at URL :
`http://cern.ch/it/pdp/is/main/experiments.html`

DESKTOP COMPUTING

Service Definition	Name	E-mail Address
NICE (Networked PCs)	F. Hemmer	`nice.support@cern.ch`
PC Desktop (Repairs and sales)	C. Ball	`pc.admin@cern.ch`
MACintosh Support	F. Ovett	`mac.support@cern.ch`
MACintosh Sales	W. Hug	`waltraud.hug@cern.ch`
X Terminals Support		`xterminal.support@cern.ch`
ASIS Support		`asis.support@cern.ch`

Operating System Support

Service Definition	E-mail Address
HP 700 s/w:	`hp.support@cern.ch`
SUN s/w:	`sun.support@cern.ch`
Solaris s/w:	`solaris.support@cern.ch`
RS/6000 s/w:	`aix.support@cern.ch`
Digital Unix s/w:	`dunix.support@cern.ch`
Linux:	`linux.support@cern.ch`

SCIENTIFIC APPLICATIONS & SOFTWARE ENGINEERING

Service Definition	Name	E-mail Address
CERN Program Library	L. Besson	`heplib.support@cern.ch`
PAW Support	O. Couet	`paw.support@cern.ch`
Anaphe and GEANT		`heplib.support@cern.ch`
Oracle Databases	N. Segura	`oracle.support@cern.ch`
Objectivity DB Service		`objectivity.support@cern.ch`
Software Development Tools		`sdt.support@cern.ch`
Adobe FrameMaker		`docsys.support@cern.ch`

PaRC Service (Engineering Cluster):
for each application see the “`mailto`’s” E-mail address as given by the URL:
`http://cern.ch/it/ce/support_index.hmtl`

COMPUTING FOR ENGINEERING

All comments, problem reports, and suggestions can be reported via the “Feedback” Web form accessible from the “IT-CE Home page” (`http://cern.ch/it/ce`, link “Feedback” on the left).

COMMUNICATIONS AND NETWORKS

A complete and up-to-date list of contacts is at the “Communications Systems Group” Home page at the URL: `http://network.cern.ch`

INTERNET SERVICES

Service Definition	E-mail Address
Web support	`www.support@cern.ch`
Mail support	`mail.support@cern.ch`

In addition to this generic address, the normal support lines are listed
at the URL: `http://consult.cern.ch/service/mail/problems/contacts.html`

COMPUTER SECURITY

Computer Security Information	`computer.security@cern.ch`
Security Alert Reports	`cert@cern.ch`

Other Newsletters

Computing for Engineers (http://cern.ch/it/ce/newsletter/newsletter.html)

Editor/Distribution: L.Pregernig ludwig.pregernig@cern.ch
Administrative Information Services (http://ais.cern.ch/newsletter)

Editor/Distribution: AIS.newsletter@cern.ch

Some Useful Web Pages

TvScreen (Systems Status)	http://tvscreen.cern.ch
Network Connection Request Form	http://network.cern.ch/register
Remote/Mobile Computing	http://network.cern.ch/mobile
User-Oriented Catalogue	http://cern.ch/consult
User-Oriented Service Directory (*)	http://cern.ch/services
FrameMaker Support	http://cern.ch/framemaker
Text Processing related documentation	http://cern.ch/it/asdoc/textproc.html
IT Home Page	http://cern.ch/it
User Support	http://cern.ch/it/support
NICE Home Page	http://nicewww.cern.ch/
PC Support Service Page	http://pcwww.cern.ch/
PC Farms Homepage	http://cern.ch/it/pdp/pc/
UNIX Workstation and X Terminals Support	http://cern.ch/it/pdp/ose/support.html
CASTOR Query Report	http://wwwinfo.cern.ch/htbin/pdp/castor_report.pl
Software Development Tools	http://cern.ch/sdt
Computing Support for Engineering	http://cern.ch/it/ce
Joint Controls Project (LHC Experiments)	http://itcowww.cern.ch/ITCO/jcop/
Web Services	http://cern.ch/WebServices

(*) *Useful service entry points at this page include the Computing Helpdesk, User Registration, Computing Bookshop, Mail, Netscape, Listbox, etc.*

Seminars and Presentations

Designation	Usual Location	Organizer(s)	Tel.	E-mail Address
Computing Colloquia	Auditorium / bldg 500	S. Jarp, IT	74944	sverre.jarp@cern.ch
Computing Seminars	IT Auditorium 31/3-005	L. Pregernig, IT	72666	computing.seminar@cern.ch
Technical Presentations	IT Auditorium 31/3-005	S. Cannon, IT	75036	stanley.cannon@cern.ch
Informatics Tutorials	IT Auditorium 31/3-005	M. Marquina, IT		it-training.service@cern.ch

Computer Resources Allocation and Divisional Contacts

Computer Resources Allocation Committee (COCOTIME):

Chairman:	H. Hoffmann/DG	60/5-006	75458	Hans.Falk.Hoffmann@cern.ch
Secretary:	H. Meinhard/EP	1/1-028	76031	Helge.Meinhard@cern.ch

Divisional Representatives for Computing

For questions about new registration and resource allocation please ask the contact person for your division.

Div.	Repres.	Div.	Repres.	Div.	Repres.	Div.	Repres.	Div.	Repres.
AS		AT	P. Heymans	IT	A. Koppanyi	ETT	M. Draper	FI	C. Saitta
EST	Ch. Delamare	HR	M. Buttner	EP	A. Bjorkebo	PS	J. Riche	SL	M. Bornand
ST	E. Sanchez-Corral	TH	E. Gianolio	TIS	G.R. Stevenson				

NICE Divisional Representatives

See the URL http://nicewww.cern.ch/homepage/nicerepr.htm

Divisional Representatives for Networking

For questions about new installations, testing, local coordination, etc., please ask the contact person for your division (or, if you are an experimental physicist, the networking contact person within your experimental group).

The complete and up-to-date list of contacts is given in the “Communications Systems Group” Home page, at the URL: http://network.cern.ch.

Editorial

Happy Birthday to Us

*We have crossed the milestone of a new millennium. At least some of us think so (but it is not unanimous, see the related article in this issue, "*When did the Millennium Start?*").*

With it, a special birth-date: February 15th was the 35th anniversary of Mrs. Computing NewsLetter. *She was born in 1966 with the vocation to serve as a communication channel between the users and us. Many changes have happened since, and our CNL has witnessed most of them.*

For this reason we propose this special issue to address with some historical perspective the events which have marked DD-CN-IT over the past years, both from the technological as well as the human points of view.

The event came to our attention quite incidentally. Miguel was organising last summer the scan of the indices to all issues and there it was, a modestly typed first number. Since then we have been working on charming colleagues to get contributions appropriate to the event.

This effort would not have been fruitful without the contributions of the protagonists, in particular the senior staff members who have witnessed a fair number of events over their professional careers. Allow us to express to the many people who have made all this history possible:

- *respect to those who have already left,*
- *"envy" to those going nowadays (you don't look forward to full immersion in the LHC-computing times, do you? :-))*
- *continued courage and vitality for those here and coming over the next 30 years (We'll not witness ourselves the "retirement" of Mrs CNL).*

We will not emphasize any particular contribution, because they are all, with their own style, a source of interest and, for some, even fun. We hope you will enjoy the reading as much as we do. We already invite you to a "Rendez-vous" in fifteen years, to mark the 50th birthday of the CNL if it still exists. Although a "human editor" may no longer be producing it at that time!

Nicole Crémel and Miguel Marquina,
IT/User Support.

1. Announcements

This chapter contains announcements concrete in time and/or of informative nature from the division to the User Community.

1.1 The UCO is dead, Long Live the Computing Helpdesk

Nicole Crémel and *Roger Woolnough, IT/User Support*

Abstract

Computer accounts associated with the "UCO" were deleted at the end of February 2001. Since then, mail sent to `uco@cern.ch` or to `uco@mail.cern.ch` has been rejected.

The e-mail address `uco@mail.cern.ch` was phased out in 1998 in favour of the more explicit e-mail addresses `Computing.Helpdesk@cern.ch`, or simply `Helpdesk@cern.ch`.

The following news (in English and French) was posted on March 2, 2001 in the `cern.computing` and `cern.nice` newsgroups, following the deletion of the obsolete "`uco`" address:

While the UCO (User Consultancy Office) used to exist there were several e-mail addresses one could use, e.g., `uco@cern.ch`, `user.support@cern.ch`, etc. With the implementation (in 1999) of the new outsourced Computing Helpdesk within IT, one of the goals was to centralise problem submission via e-mail to `Helpdesk@cern.ch`, or via telephone to 78888.

We have now deleted all "`uco`" logins, and mails sent to `uco@cern.ch` or `uco@mail.cern.ch` will be rejected. You should send any request for Help to the Computing Helpdesk at `Helpdesk@cern.ch`.

Please note: The <`User.Support@cern.ch`> e-mail address is no longer linked to them. It can still be used to contact the User Support group for comments or suggestions on the service you receive from the Helpdesk.

If `uco@cern.ch` or `uco@mail.cern.ch` is still an entry in your personal address-book, please modify it accordingly.

A l'époque de "l'UCO" (*bureau d'aide aux utilisateurs*), on pouvait contacter le bureau à différentes adresses électroniques, comme `uco@cern.ch`, `user.support@cern.ch`, etc. Avec la mise en place (en 1999) d'un nouveau "Computing Helpdesk" sous-traité par la division IT, un objectif a été de centraliser la soumission des problèmes informatiques à l'adresse électronique `Helpdesk@cern.ch`, ou par téléphone au 78888.

Nous avons désormais supprimé les comptes informatiques "uco", et les messages adressés à `uco@cern.ch` ou `uco@mail.cern.ch` seront rejetés. Vous devez envoyer toute requête pour un problème informatique au "Computing Helpdesk" à l'adresse électronique `Helpdesk@cern.ch`.

Remarque: L'adresse électronique <`User.Support@cern.ch`> n'est plus liée au Helpdesk. Elle peut encore être utilisée pour contacter le groupe du Support aux Utilisateurs pour transmettre des commentaires ou des suggestions concernant le service reçu du Helpdesk.

Si `uco@cern.ch` ou `uco@mail.cern.ch` provient d'une entrée dans votre liste d'adresses personelles nous vous invitons à la modifier.

About the author(s): *Nicole Crémel and Roger Woolnough are members of the User Support group in IT, and are the CERN responsible for the supervision of operations of the Computing Helpdesk.*

1.2 Why "Helpdesk" and "User Support" are two Entities?

Miguel Marquina, IT/User Support

Abstract

We have recently decoupled both user entry points, emphasising the role of `Helpdesk@cern.ch` as a point of first-level assistance. Offering additional justification for our motivation in doing this does not harm.

In CNL 2000-003, p. 11, "*Change in E-mail Destination for `User.Support@cern.ch`*") we announced that the e-mail address `User.Support@cern.ch` is no longer linked to the Helpdesk. Following some demands for clarification from colleagues, we think it is useful to provide some additional background to this decision.

We are aware that it will take a while to align all the documents, pointer, links, etc., that are still referring to `User.Support@cern.ch` as a synonym to the point of first-level assistance to users, our present `Helpdesk@cern.ch`. Hopefully obsolete stuff will just die and new stuff will contain the "improved" pointers.

As what regards added value to users, we are convinced that we gain in promoting one-desk, one-address, one-number (`Helpdesk`, 78888). And that we have to offer users a different "window" (`User.Support`) where to complain when this single entry point does not perform as expected.

In what regards the future, the consolidation of this model of work in IT offers confidence that the term (and concept) for "Helpdesk" will last, regardless of how internal restructuring might affect units like "User Support".

About the author(s): *Miguel Marquina is the Head of User Support in IT.*

2. Physics Computing

Functional description of all the Central Public Services, specifically oriented to use by the Research sector, either for interactive (central and CPU intensive) or batch processing. The main physics-computing services consist of providing computer farms, central data recording facilities and central data storage facilities.

2.1 Nostalgia from the Computer Centre

Roger Woolnough, IT/User Support

Abstract

History and anecdotes as remembered by one of the Computer Centre Operators.

30 years ago when the Computer Centre was in 510 there was really close touch with Users and Operators alike. I wonder what happened to the rolling racks where users sent their jobs in punched cards contained in small cardboard boxes? Priority jobs had a code "`zip`" at the end, one of our own with a name ending in `zzi` and the initial of `p` was always served quickly (smart move at the time Paolo!)

We had characters that we remembered, the Wim Kleins, those who only came in at night, those that never went home and of course the famous pipe smoker of 513. All these people had cause to pass by whereas now they remain in their offices to work.

We had characters too, one who could tear a phone directory in half, one that worked at a beach and couldn't swim, one who chewed garlic constantly, where are they now? We had CDC engineers with endless patience; I'd seen them clean huge disks over 12hours only for the heads to crash when they land them again, a further 12-hour cleaning job. We had a CDC analyst who spent 3 months in his office to find a bug from dumps and even one that had a station masters hat! In fact we had so many characters then when one wanted to take holiday if the team was not forty strong we were refused. I can see the operations manager counting names in the afternoon and refusing two half days as there were only 38 operators available… But where were they when you needed them?

The machinery has changed slightly. No more one inch magnetic tapes that weighed eight pounds each and took four minutes to rewind having been written at 220bpi. No more paper tape that wrecked and was full of static electricity. The punched cards, the box of binary cards so lovingly stacked and then dropped on the floor accidentally. Remember the printers with chains and hammers, pre-printed paper that had carbon in between, anything over 4ply and you couldn't read it…

Even the infamous RIOS (Remote Input Output Stations) that were scattered around the site had tales to tell. The operator who returned without the Renault 4L van he had driven there but had the L-shaped gear stick in his hand having walked back from West Hall. Not to mention a 4L on its roof in the car park on Restaurant no 2 at 2am… Printers that regularly wrecked paper, people found asleep waiting for output.

One of the engineers left the company and became a pinball engineer. There were also the traditional parties that produced their fair share of notoriety. The yearly "fondue" party at the Café de Renfile in Vernier was always well attended, even by the assistant division leader at the time. "Fondue" was not always confined to the pots and one member arrived dressed in a polythene dustbin bag for protection. Naturally with the consumption of alcohol being fairly copious it was always the aftermath that produced the most laughs. Various modes of transport (for those still capable of standing) were then used to continue the festivities in Town, fortunately no members of the Group then had to come to work until the following morning.

Once in 513 the Centre lost some of its character although the football team still did well. The rolling racks for input jobs were still present although now they came through a wall and you couldn't see who sent them. The famous television set, which projected an image of a white board with magnetic letters, these had to be updated during breakdowns, this happened often. The paternoster device, like an ancient dinosaur that spat out boxes of tapes that came from a dark and mysterious basement area where people came and went independently of other teams, one just hoped they were there when needed. We had a fireman on permanent duty, he turned out to be very useful. The day we had a fire he had to call the fire brigade while the author who was present had to fight the fire! Plastic wrapping around tubing that caught alight in the vault causing extensive damage to the top of the range Newbury terminals we had at the time; never mind, there was always the card-reader.

With innovations came a micro-fiche device, this meant dozens of readers had to be bought, some can still be seen around the site in dark and dusty corners. Still operators came and went. We recruited one person from CERN, later to run away back to England with a female friend. One returned to be a policeman in England, some went to France, Sweden and South Africa, one even became a salesman for ladies underwear. All spreading the word of the great team spirit we had then. Like many other activities it is now outsourced and instead of the 40 or so persons per team we are down to 1, that's progress but the characters are missing!

2.2 Once Upon a Time… The Mainframe Era

Eric Mcintosh, IT/API

Abstract

A light hearted look back to the mainframe era.

I came to CERN in 1964 as an employee of CDC, more or less packed in a box with the CDC 6600. This was the Serial Number 3 machine, literally the third system to be manufactured. Serial Number 1 was installed at the Livermore Laboratory and Serial Number 2 was used for software development until it caught fire.

The 6600 was intended to replace the IBM 7090 and its 1401 front end system. There was no mass storage at that time, no disks or drums, just magnetic tapes, cards and paper tape. Jobs and programs were on punched cards which were copied to magnetic tape on the 1401. The tape was read by the 7090 and the output also written to tape for subsequent transfer to the 1401 for printing. With the long queues and this lengthy procedure, desk checking was worthwhile. I personally found the punched card system a big improvement over the paper tape system I had been using in the UK; punched cards were considered too expensive there, as were flexowriters, so I was a real expert at decoding the punched holes on the paper tapes.

In order to facilitate the migration from IBM to CDC Drs Erskine and Lipps implemented a 7090 simulator for the CDC 6600. Since there was no CDC 6600 available, we tried to test it on a CDC 1604 system in Zurich which had a 6600 simulator. This project was abandoned after finding that it took about an hour to simulate one IBM instruction in this simulated simulated environment!

The 6600 had no software but CDC were implementing a new super operating system in California… It was never finished. In fact the Chippewa (after Chippewa Falls, the home town of Seymour Cray) Operating System or COS was finally used by almost everyone. First it was translated from octal into assembler and was the basis of further OS development leading to the NOS and NOS/BE systems. Every Sunday evening we would test a new version of whatever OS we were using; it consisted of a couple of trays of blue binary cards with a few extra holes punched or covered up, as last minute patches. Every Monday morning we would introduce the new version at 08:00 and withdraw it at 08:30 because of user complaints.

A well known physicist, later to become a Nobel prize winner and DG of CERN, seemed to always be the first to discover these problems; perhaps because his jobs were somehow always at the front of the queue. The Computer Coordinator had a collection of red-rimmed priority cards, and when users were particularly upset or had some rush work, he would kindly dish out a few of these to the person concerned.

For the next 20 years or so, the mainframe era, machines were expensive. A single processor system was of the order of 20 million currency units. The power of a single processor increasing from about 0.5 of a CERN Unit for the CDC 6600, to 1 Unit for the IBM 370/168, and to about 8 Units for the Cray X-MP, and to 20 for the last IBM mainframe at CERN. Today the dual processor 800MHz PC we purchase for a few thousand currency units, has some 180 CERN Units per processor.

On the other hand, people have now become relatively expensive. At one time DD had over thirty people working on basic operating systems. I spent major amounts of time, as did many of my colleagues in CDC, IBM and CERN, analysing the dead-start dumps (or IBM equivalents) described elsewhere by Julian Blake. Tensions were high with a 6600 mean time to failure of about two hours in 1965. I remember having to separate a CERN staff and an outsourced software manager as they were coming to blows over responsibility for the latest crash. My career at CERN almost ended before it started as I wrote a CDC 6600 peripheral processor program for a physicist, who also remarkably was awarded a Nobel Prize many years later, and this was counter to the DD policy of the time.

Still much progress was made in terms of functionality and reliability over the years; the advent of disk storage, and (semi-)permanent files made everyone's life easier. The introduction of a general interactive service (INTERCOM or Wylbur) made everyone much more productive, although painfully exposing the users to every system hiccup. Reliability remains a major issue today; even if each PC has a mean time to failure several orders of magnitude better than the original mainframes. When clusters of thousands of processors will be used by each LHC experiment, applications and systems will need to be designed to have fault tolerance in one way or another.

Throughout this era, Fortran was THE programming language of choice; from Fortran IV, through Fortran 66, VM Fortran, to Fortran 77. Language standards were a great help, even if CERN's attempt to define a standard CERN Fortran failed, but of course every manufacturer had, and still has, proprietary extensions. It will be interesting to see if Fortran continues to play a major role, at least for engineering applications, as all CERN developed physics applications move to C++, Java, or C#?

If I had to summarise a few things I have learned in these very exciting and happy years I would say the following:

- There is always an explanation for a problem; it is just becoming too expensive in human resources to identify it and correct it.
- Every application should be ported to at least one computing platform other than that on which it was developed. All code should be checked against the relevant standard as well, and checked by all available tools for finding memory leaks, mismatched procedure references, and array limits being exceeded. Given the long lifetime of an application, and the rapid evolution of computing technology, both hardware and software,

this effort will be well paid in the long run. (Witness the current difficulties in porting LEP applications to PCs.)

Finally, monopolies are bad. Healthy competition is an absolute must in order to meet the LHC computing requirements at minimum cost. Who knows if a significant amount of LHC computing might not be done on Z-boxes or Playstation 5's, or whatever other commodity system is installed in every home.

2.3 35 Years Ago — the Control Data 6600

Julian Blake, IT/PDP

Abstract

When CDC ruled the computing playground. . .

I arrived at CERN in October 1966, and started work as a system programmer on the Control Data (CDC) 6600. It is instructive to compare some of the characteristics of this computer with the computers we have today.

The CDC 6600 had one central processor (CP) and ten peripheral processors (PPs). There was one central memory (CM), and a private peripheral memory (PM) for each PP. There were twelve input/output channels and some pseudo channels. The CP could access its own registers and CM, but nothing else. A PP could access its accumulator register, its PM, the CM, and the I/O channels. A PP could also "exchange jump" the CP; this single instruction stopped the running CP program and started another, by exchanging the contents of the CP registers with a CP-register-image in CM. In practice one PP acted as system monitor (MTR, the heart of the operating system), another ran the operator's console (DSD, dynamic system display), and the remaining PPs constituted a pool of I/O processors. A CP program would place an I/O request in a mailbox, MTR would find the request, and allocate one of the pool PPs to perform it.

The operating system did not depend on interrupts. PP programs, notably MTR and the peripheral memory resident (PMR) in the pool processors, polled mail boxes for requests for action. The operating system ran wholely in PPs, with one exception: the memory move program, which relocated programs in CM to make space for an incoming program. The memory move program was effectively an extension of MTR running on the CP.

The CP was recognizably a RISC processor. It had multiple registers (18-bit address (A) and index (B) registers, 60-bit data (X) registers), a three address order code with 64 distinct instructions, and multiple functional units with pipelined execution. There were instructions for integer and floating point arithmetic, boolean operations and shifts, but no character manipulation instructions. Ones-complement arithmetic was used for integers and for floating point numbers. The floating point number format was proprietary, and included representations for infinity and undefined (as in today's IEEE floating point formats). CM consisted of 60-bit words, and was word addressed. By convention, a character occupied six bits and was represented in a proprietary "display code" containing digits, upper case letters, arithmetic and punctuation symbols, space and newline.

The 6600 CP had an 8-word instruction stack which functioned rather like an instruction cache, but without the flexibility of a modern cache. Program optimization consisted of allocating heavily used variables to registers, loading operands from memory a few instructions before they were needed, writing operands to memory a few instructions before their registers were needed to hold new operands, keeping several functional units busy simultaneously, and trying to get inner loops to fit into the instruction stack. The divide instruction was notorious, because it took about thirty clock cycles and its functional unit was not pipelined. The peak CP instruction rate was one instruction every 100 nanosecond clock cycle.

The ten peripheral processors worked with 12-bit words, but had an 18-bit accumulator which could hold a CM address. Instructions occupied one or two words, and could use any of the first 64 words of PM as an index register. There were 63 distinct instructions in the order code. Instructions included integer addition and subtraction, boolean operations and shifts, block and single word I/O operations on the 12-bit channels, block and single word read from and write to CM, and the exchange jump. Ones-complement arithmetic was used.

The ten PPs shared the same processor, in a cyclic manner. Each PP was served for 100 nanoseconds every microsecond. This sharing was invisible to the programmer.

Each PM contained 4096 12-bit words, which had to hold the PMR code, device driver code and a buffer for device data. Block sizes for disk and magnetic tape were limited by the size of PM. Complex code, such as that used to recover from magnetic tape I/O errors, was divided into a number of overlays, with only one overlay in memory at a time. Device drivers had to perform many of the operations that would nowadays be performed by firmware in a device's controller.

So how did one program the CDC 6600? The users wrote programs in FORTRAN IV. The CERN program library was written mainly in FORTRAN, but included some optimized routines written in assembly language. For detailed optimization, there was a CERN-written program to simulate the operation and timing of the 6600 CP. Large programs had to be divided into overlays, since there was no paging of memory. All memory was real, and each program was responsible for loading its overlays.

The operating system was written in PP assembly language, with small bits in CP assembly language. The initial assemblers (ASCENT for CP, ASPER for PP) were soon replaced by Control Data's COMPASS macro assembler, which provided a comprehensive listing of the machine code which it generated. The CP and PP order codes were simple and regular enough that it was not difficult to understand their assembly languages. CP code was the more difficult of the two, because the CP had 24 registers (8 each of A, B and X). The operating system was divided into overlays small enough to fit into PM, so each overlay was of an understandable size.

What did one do when the operating system went wrong? Provided DSD was still running in its PP, one could get a lot of information from the operator's console, which consisted of two alphanumeric displays and a keyboard. The system dayfile (log file) was a good source of diagnostic messages for predictable errors. After a crash or hang-up, one could take a dead start dump: the computer was rebooted, and printed the contents of some or all of the ten PMs and the CM in octal on a line printer. One learnt to look for the zero word in peripheral memory, which identified the PP's program counter value at the time of reboot. One learnt also to look for a particular bit dropped in every 32nd word of CM, which indicated a bad memory line driver module for that bit in that memory bank.

There was no read-only memory in the 6600. The dead start bootstrap program was set on 12 rows of 12 toggle switches. The contents of the switches was loaded into the start of PM0 to begin the bootstrap process. Changing the dead start panel switches without warning the next operator of the computer was considered an unfriendly act.

The input/output devices attached to CERN's 6600 in 1966 were a hard disk, several half inch and one inch magnetic tape units, a card reader, a card punch and a line printer.

Readers who are interested in digging further, can find a fair amount of information about the CDC 6600 on the Web (ask *Google* to search for "`CDC6600`"). There are photographs at the URLs `http://histoire.info.online.fr/cdc6600.html` and at `http://www.newmedianews.com/tech_hist/cdc6600.html`. There is technical information at `http://ed-thelen.org/comp-hist/cdc6600.html`.

About the author(s): *Julian Blake wrote a program loader and a memory scheduler for CERN SCOPE, which was CERN's derivative of Control Data's Chippewa and 6000 SCOPE 2.0 operating systems.*

2.4 A *SHIFT* in Outlook

Ben Segal, IT/PDP

Abstract

How the all-powerful IBM and Cray mainframes were replaced by nasty distributed Unix boxes.

I don't remember exactly who first proposed running physics batch jobs on a Unix work station, rather than on the big IBM or Cray mainframes which were doing that sort of production work in 1989 at CERN. The work station in question was to be an Apollo DN10000, the hottest thing in town with RISC CPU's of a formidable 5 CERN Units each and costing around 150,000 Sfr. for a 4-CPU box. It was the combined idea of Les Robertson, Eric McIntosh, Frederic Hemmer, Jean-Philippe Baud, myself and perhaps some others working at that time around the biggest Unix machine that had ever crossed the threshold of the Computer Centre (a Cray XMP-48, running UNICOS).

At any rate, when we spoke to the Apollo sales people of our idea, they liked it so much that they lent us the biggest box they had, a DN10040 with four CPU's plus a staggering 64 MBytes of memory and 4 Gigabytes of disk. Then, to round it off, they offered to hire a person of our choice for three years to work on the project at CERN.

In January 1990, the machine was installed and our new hire, Erik Jagel, an Apollo expert after his time managing L3's Apollo farm, had coined the name "HOPE" for the new project. (HP had bought Apollo, OPAL had expressed interest, so it was to be the "Hp Opal Physics Environment"). We asked Jean-Claude Juvet where we could find the space to install HOPE in the Computer Centre. We just needed a table with the DN10040 underneath and an Ethernet connection to the Cray — for getting access to the tape data. He replied: "Oh, there's room in the middle" — where the recently obsolete round tape units had been — so that's where HOPE went, looking quite lost in the huge computer room with the IBM complex on one side and the Cray on the other.

Soon the HOPE cycles were starting to flow. The machine was surprisingly reliable and porting the big physics FORTRAN programs was easier than expected. After around six months the system was generating 25% of all the CPU cycles in the Centre, which became known to the management when we included HOPE's accounting files into the weekly report that plotted these things into nice histograms for easy reading.

We were really encouraged by this success and went to work on a proposal to extend HOPE in a systematic way. The idea was to build a scalable version from interchangeable components: CPU servers, disk servers and tape servers, all connected by a fast network and software to create a distributed mainframe. "Commodity" became the keyword - we would use the cheapest building blocks from those manufacturers who had the best price-performance for each function at any given time.

How big could such a system be built and what would it cost? We asked around and received help from many colleagues for a design study. A simulation was done of the work flow through such a system, bandwidth requirements were estimated for the fast network "backplane" needed to connect everything, the prices were calculated, the required software was sketched out and the manpower for development and operation was predicted.

The software development would be quite a challenge. Fortunately, some of us had been working with Cray at CERN, adding to Unix some missing facilities vital for mainframe computing: a proper batch scheduler and a tape drive reservation system, for example. These facilities could be reused quite easily. Other functions required would be a distributed "stager" and a "disk pool manager". These would allow preassembly of each job's tape data, read from drives on tape servers, into efficiently managed disk pools located on disk servers, ready to be accessed by the jobs in the CPU servers. Also new would be "RFIO", a remote file I/O package to offer a unified and optimized data transfer service between all the servers via the backplane, looking like NFS but with much higher efficiency.

Finally, a suitable name was coined for the project, by Erik Jagel again: "SHIFT". It stood for "Scalable Heterogeneous Integrated FaciliTy", a bit forced for the final "T" but inescapably suggesting the paradigm shift occurring in large scale computing: away from mainframes and towards a distributed low-cost approach. The "SHIFT" proposal report was finished in July 1990. It had ten names on it, of the colleagues from several groups that had helped with their ideas and worked on the document. It was sent to the CN management and then we waited.

Our surprises began. It seemed that not everyone thought this was a good idea after all. Enquiries were received from Director level: had there really been ten people working on such a scheme? How many precious Cray resources were being used to feed this HOPE gadget — and how many of HOPE's claimed 25% CPU cycles were in fact Cray cycles? In fact the Cray had been used only as a convenient tape server, because it was the only Unix machine in the Computer Centre with access to the standard tape drives, all physically connected to the IBM mainframe at that time.

The CN Divisional management was more open. We were told the following: if we could persuade at least one of the four LEP experiments to invest in our idea we would have matching support from the Division. The search began. We spoke to ALEPH, but they replied "No thank you, we're quite happy with our all-Vax VMS approach". L3 replied "No thanks, we have all the computing power we need". DELPHI replied "Sorry, we've no time to look at this as we're trying to get our basic system running".

Only OPAL took a serious look. They had already been the partner in HOPE and also had a new collaborator from the University of Indiana with some cash to invest and some SCSI disks for a planned storage enhancement of their existing VMS based system. They would give us these contributions until March 1991, the next LEP startup, on condition that everything was working by then or else we'd have to return their money and their disks. It was September 1990, and there was a lot of work to do.

I believe that our modular approach and use of the Unix, C language, TCP/IP and SCSI standards were the keys to the very short development timescale we achieved. The design studies had included technical evaluations of various work station and networking products. By September, code development could begin and the first orders for hardware went out. The first tests on site with SGI Power Series servers connected via UltraNet took place at the end of December 1990. A full production environment was in place by March 1991, the "go-nogo" date set by OPAL.

And then we hit a problem. The disk server system began crashing repeatedly with unexplained disk errors. Our design evaluations had led us to choose a "high-tech" approach: the use of symmetric multiprocessor machines from SGI, for both CPU and disk servers, connected by the sophisticated "UltraNet" Gigabit network backplane. One supporting argument had been that if the UltraNet failed or could not be made to work in time, then we could put all the SGI CPUs and disks together into one cabinet and ride out the OPAL storm. We hadn't expected any problems in a much more conventional area, the SCSI disk system.

Our disks were mounted in trays inside the disk server, connected via high performance SCSI channels. It looked "standard", but we had the latest models of everything. Like a Ferrari it was a marvel of precision but a pig to keep in tune. We tried everything but still it went on crashing and we finally had to ask SGI to send an engineer over from Mountain View. He found the problem: inside our disk trays was an extra metre of flat cable which had not been taken into account in our system configuration. We had exceeded the strict limit of 6 metres for single ended SCSI, and in fact it was our own fault. Instead of SGI charging us penalties and putting the blame where it belonged, they lent us two extra CPUs to help us make up the lost computing time for OPAL and ensure success of the test period.

At the end of November 1991, a satisfied OPAL doubled their investments in CPU and disk capacity for SHIFT. At the same time, sixteen of the latest HP 9000/720 machines, each worth 10 CERN Units of CPU, arrived to form the first Central Simulation Facility or "Snake Farm" on account of the machine's code name within HP. The stage was set for the end of the big tidy mainframes at CERN and the much less elegant but evolving scene we see today on the floor of the CERN Computer Centre. SHIFT became the basis of LEP-era computing and its successor systems are set to perform even more demanding tasks for LHC, scaled this time to the size of a world-wide Grid.

About the author(s): *I graduated in Physics and Mathematics in 1958 from Imperial College London, then worked for 7 years on fast breeder reactor development, first for the UK Atomic Energy Authority and later in the USA for the Detroit Edison Company. I've been at CERN since 1971, after finishing my Ph.D. at Stanford University in Mechanical and Nuclear Engineering.*

Except for a sabbatical in 1977, when I worked at Bell Northern Research in Palo Alto on a PABX development project (and encountered Unix for the first time), CERN has kept me pretty busy on various projects, including the coordinated introduction of the Internet Protocols at CERN beginning in 1985 (see: "A Short History of Internet Protocols at CERN"). For more details on my work, see my CV and some associated Notes, together with a partial List of Publications.

3. Desktop Computing

Description and news concerning the CERN desktop environment centrally provided on UNIX and PCs (NICE95 and NICE-NT), and providing the "general-purpose computing" environment. It consists of the hardware, software strategy, and services for the basic applications, which one expects to be generally available on desktops. This environment presents the basic computing infrastructure of the laboratory.

3.1 Tribulations of a SupportMan in ComputingLand

Miguel Marquina, IT/User Support

Abstract

I have also my modest anniversary to celebrate. In July this year it will be 20 years that I got hooked on computing. This is my personal perspective of how have I lived this period.

I am physicist by education. Actually until the summer of 1981 I had not had access to a computer, ever. Not even a Commodore or Spectrum.

All started with FORTRAN

In spring 1981 I was in my last year of studies. Juan Antonio Rubio, head at the time of High Energy Physics department at "Junta de Energía Nuclear", was head-hunting students to work in his group as part of a yearly recruitment exercise. He came to our university to tell us about particle physics and CERN, and invited in exchange those interested to submit CVs to his consideration.

I wonder what it was. Maybe the magnetic attention I was paying to his talks (they were very good), maybe the prospect to pursue a "research career" rather than recycling myself into another field... Whatever it was, I applied and I became one of the lucky ones.

By the summer the course was over, and I joined Rubio's group. My first assignment was simple enough: "Here are the notes to a FORTRAN tutorial attended by another colleague; here is a punched card machine that all of us use regularly. However being a newcomer we will save you of that hassle; we have installed one terminal (to a UNIVAC mainframe). Learn the language, learn how to send batch jobs via the terminal, come back when ready."

After two weeks of feverish activity I reported being ready for the next step. Interesting enough, I had wondered during that time why other staff was poking shyly into the terminal room and departed after a humble "Good morning" or "Good afternoon". But since there was no other interaction I was just turning back and spent my training hours in front of the green screen (minus lunch time).

I shared this curious behaviour with my supervisor, who then replied to me with an ample smile: "Ah Miguel, I am afraid I forgot to tell you. This brand new terminal is one of the two in the whole building, and staff needing it are expected to book through this notebook their slots. You must have been bypassing the reservations of everybody over the past 10 days!"

Although I had not realised it then, I had become a computoholic.

Discovering the pleasure to help

So I had a frantic startup at the UNIVAC of my institute, but I had not really been charmed into the intricacies of Assembler, 36 bits or the guts of the Operating System.

I slowly found out that my proficiency at the FORTRAN language and the literal devoring of the computer documentation (just delivered) turned me into one of the possible references to "difficult questions". I don't say I was not flattered by that feeling. Being able to offer advice to my own colleagues!

Maybe due to these regular demands for help, but also because I had to learn about the "CERN Programming environment" in order to participate in the software developments of the group, I started to acquire the basic notions of the CERNlib of the time: HYDRA, HBOOK, HPLOT, etc. And the corresponding manuals.

I had been assigned to share an office with another colleague (and good friend of mine) Carmen Albajar. Maybe because we both liked a systematic approach to problems and had a higher level of "self-organization", we started compiling relevant documentation. To such extent that we built a de-facto "documentation reference library" for the group. Colleagues offered us some of their (unique) copies such that everybody would know where to find them when needed.

I guess that one of my leading focus in supporting users seeded there. Good Quality Documentation is the Valet of Good Quality Computing.

Meca of Computing

My second computing "encounter" happened in the summer of 1982. My supervisor, Pedro Ladrón de Guevara, took me to CERN. He was member of several NA collaborations and there was some field work which had to be prepared at the experimental area.

Pedro was a good friend of a certain Julius Zoll. So I was introduced to a very charming, sympathetic fellow; sometimes perhaps dogmatic in his computing opinions but who was I to do anything else than to learn from a lifetime experience? Only a while later I learnt I had met one of the ancillary references to CERN computing.

A visit to the Computer Centre at building 513 was an obliged visit after the large halls holding the fixed-target experiments ("Here is the heart to our research"). I found out that the impressive computing facilities portrayed in James Bond' films were not invented (actually I wonder if some film director of the time would have considered the CERN Computer Centre as an scenario for a futuristic spy movie; the means were there). I discovered a hall filled with two-meter high IBM "cupboards" holding several hundred megabytes each, a quite respectful collection of powerful CDC boxes, and some impressive printing/plotting equipment.

When we came out of the gallery, he brought me to a modest location (for the anecdote, my own office now) to get hold of some manuals and pose a couple of questions. I read there "Program Enquiry Office". When we came out I asked my supervisor, and he explained: "While you develop your programs at these mainframes, sometimes they will blow up at execution time. You will think you did not change anything which could explain the error and that the machine just failed. So you would phone or contact this place and they will invite you to bring along a dump of the aborted program. After some time of tracing the logs by hand, they will find out which is the instruction which caused the failure…"

I found that kind of work fascinating. Sherlock Holmes and the Lost Bit.

Designing GUIs for GUYs is a serious business

Shortly after I was proposed to join in the Spanish team from my institute, participating in the Mark-J experiment at DESY. During the next three years I learnt the basics of High Energy Physics life: running programs and making shifts. For a doctoral student that is.

Computing-wise I got acquainted with MVS/NEWLIB and the GEP graphical package (both very fine pieces of work by the DESY/Rechnenzentrum staff); and also the JES3 and JCL horrible technobabble of devine meanings that put off mortals at once. In the collaboration it was customary not only to do software analysis in the corresponding area of research or thesis topic, but also contribute to some task of common usefulness for the Collaboration.

I opted to team up with a Dutch colleague in order to develop a tape staging pooling system based on our IBM maxidisk allocations; nothing to do with today's complex environment, but still with enough challenge to complement night thesis fun with some programming bursts.

This little project taught me that user satisfaction builds more on robust and intuitive user interfaces and meaningful error/fault messages, than on an infallible back-end. Faults are (better) tolerated as long as (scientist) users are given valid (scientific) explanations.

My second lesson is that it is physically impossible to meet conflicting user requirements of more than three users at a time.

A User Community on its own rights

Until May 1986 I was at DESY in an experiment of some 40 members working on location, who were actually the colleagues interacting on a day-to-day basis. We knew each other by first name.

Then I was reassigned to the Spanish team working at CERN, this time joining the UA1 Collaboration. I kept busy for a while automating our (institute) software chain under MVS/Wylbur, scripting JCL stuff and so on. Nothing significant really in the context of a collaboration five times as large. Although reluctantly I learnt VM (I always thought it was a loss compared to MVS/Wylbur and specially NEWLIB) and REXX. I explored the possibilities of the UA1/Merlin graphics system, and came to terms with HTV (oh!, how badly I was missing GEP).

I lived like some others my first user migration: CERN was leaving behind MVS and was asking users to embrace CERNVM. Critical me (but I was not the only one; it was a habbit in UA1), I found that DD was not really getting the act right by proposing users three different interfaces to migrate information from MVS to VM: one to move flat ASCII files, one for binary, one for MVS libraries; of course the three written by different authors. I undertook to glue them into one for the good of UA1 mankind, and "MVSGET" was born at our user "U" disk. With a friendly panel and understandable choices, so I thought.

It occurred to me that such interface might be good enough to save other CERN users some migration hassles. I approached Harry Renshall (as I had seen so much from him in the news and CNLs) to offer my goodie for the sacred VM "P" disk. "MVSGET" did not make it, but I somehow impressed the gentleman because few months later I was invited to apply at DD for a user-support position (created by the departure of a colleague).

I did not think twice about it. The board turned down the application of a very young and gentle Belgian lady and opted for me. Harry confessed to me later that he had helped the choice by stating that either he would carry over his support duties with an older-but-experienced Marquina or he would resign at once and let the support work be done on its own.

I remember Harry in my night swears ever since.

King Salomon's job

Moving to DD in July 1988 was not as simple as it could have been. Some months before I had had the initiative to script the UA1 simulation chain at the newborn Cray/UNICOS system. A good move for UA1 because there was a major source of CPU power unexploited at our fingertips. The pleads of my colleagues turned my move into a liaison between UA1 and DD, representing UA1 in their never-enough requests for CPU.

I learnt what was the cost of representing user interests "against" one's own colleagues; that it is not straight-forward to match demands to available resources and that it was tough to be sandwitched between both and still come up with certain satisfaction of an accomplished job.

But this was not my only newcomer problem. I suddenly found myself with the power to modify things that only months before I had felt so critical about. I had a password to *modify the "P" disk*. The read/write password! What a misfortune, because I started "improving" (that was my justification) feverishly everything I could think of. And worse, I was informing users about it.

My closer friends worried for my health. The more humorous ones just nicknamed me the "DD Lighthouse", because my office window was at the side of the entrance to building 513 and was welcoming visitors at almost any time at night.

A Library is not only a collection of books

Federico Carminati definitely understood how to take advantage of the situation. He was the CERN Program Library manager in the User Support group in 1989, and needed some silly youngster to undertake the UNIX builds. He was responsible for the Vax/VMS, Cray and Apollo versions (besides the maintenance of MVS and CDC for external sites); but needed really a hand for the CERNVM and forthcoming AIX, DEC, HP and SunOS4. So he wicked me at some bus trip returning from I don't remember where.

Given my CERNVM fluency he naturally delegated the VM port to me and entrusted me with the development of automated and portable procedures to produce all the UNIX builds. And there I was, turned into a "CN Lighthouse" and facing the impossible job to manage multiple ports from a single set of shell scripts.

In 1990 René Brun took over a reshaped User Support group under the CN seal; it was called rather "Application Software" (which changed in fact the entity and scope of the unit for the following seven years). Federico found a golden opportunity to escape into better duties, took over GEANT and left me stuck with CERNlib and my own masochistic pride.

Of course, the more ports we incorporated, the more companies expressed interest in getting into the bandwagon: Alliant, Convex (32 and 64 bits), NEC Supercomputers, NeXT, you name it. Everybody wanted to sell to High Energy Physicists. And more HEP institutes expressed desire to have their local variants of computer brands and compilers to be supported by the capital of the empire.

I became the perfect example of "give him more since he still digests", and my mailbox started becoming a black hole (a lot coming in, nothing coming out). I also particularly enjoyed the library builds the weekend before a release. Delightful hundreds of thousands of lines of Fortran code not passing the validation tests on a Sunday evening…

Four years later luckily somebody thought it was a good time to offer me a change…

From VM to UNIX by the hand

In the fall of 1993 CN division started considerations of how to terminate an otherwise healthy mainframe era, since the user demands for CPU capacity were about to exceed by several orders of magnitude what any possible extension of the IBM system (3081, 3090, 9000) would allow. The Programme of Work (POW) of that year addressed an internal reorganisation, and identified among others the need to start a migration campaign to get users into the unavoidable distributed UNIX environment. Chris Jones was mandated to lead the "Desktop Computing Infrastructure" group and I was invited to join in. My job would be to sell VM users UNIX well enough that they would happily and unconditionally abandon VM three years later.

Since the word "users" was in my job description, Harry pleaded the "User Consultancy Office" unit to be transferred under my supervision rather than have it shut down as an undesirable result of the reorganisation.

Well not quite that. The truth is that I had entered in DD with a User Support hat and that I had remained in such a role ever since. Harry my DD mentor trusted me enough to be able to carry over his "vision" on how to assist users, and I did appreciate and enjoy that challenge.

Chris kept me busy for the following three years. The beginning of the migration campaign was not easy. I had been mandated to run something promisingly called the "User Migration Task Force". One of my initial reports at some internal meeting started with the following joyful remark: "Dear colleagues, I am afraid I have not made much progress in the run of the User Migration Task Force. Unfortunately I have been given the Task, but not the Force…"

Chris jokes with me about that lapidary statement ever since. He actually made something better than that, because he filed solid resourcing requests for UMTF at the following POW of 1994. It came up well and UMTF could fly. The whole of DCI group responded at each area of responsibility and, together with the rest of the division and PDP group in particular we managed to set the user countdown to the end of June 1996.

Although the replacement services showed initial glitches, CN had managed to have several thousand users switching from CERNVM to the AIX-based cluster RSPLUS in quite an orderly fashion. I think the day we officially shut down the system I recorded the first sign of computing nostalgy. In spite of the champaign at the Computer Room.

User Support: part II

Jürgen May became our next Division Leader in 1997, and we became IT. Coming from DESY, he had solid convictions for the existence of a unit looking specifically after user issues as a "raison d'etre", and not as part of its service provision mandates. There were internal considerations of all sorts speaking in favour and against such initiative, and under what kind of umbrella the "User Consultancy Office" should stand.

In the end Jürgen pleaded a chance for the idea among the divisional management and offered me the undisguised honour of becoming its advocate. Since then I have tried to work hard to stand up to expectations, by both users and colleagues.

Manuel Delfino followed Jürgen two years later, revalidating with the next turn of divisional management the place created for such a unit. I can only be indebted on behalf of myself and my team for giving us a regular chance to prove or show our possible added values, and to let us work on our strong convictions of helping users to our best possible.

Wrap-up to 20 years of supporting users

I have tried to give a dynamic style to the story, which is partly mine, partly my living of CERN computing. To those people I have mentioned I ask for their acceptance of my rather colloquial descriptions. To those people I have not I ask for their forgiving, since I have all in my memories. To all I ask for your tolerance in the freedom (and imprecisions) I have given to myself in describing events.

About the author(s): *Miguel Marquina is head of the User Support group in IT since 1997. He has worked in different support assignments since joining DD in 1988. Worth quoting are his responsibilities as the CERN Program Library manager in the period 1990–1993 and the leading of the User Migration Task Force during 1994–1996.*

4. Internet Services and Network

Everything related to Internet in a large sense: i.e., network issues and performances, all technical Web issues, electronic mail, Internet news, home access to CERN facilities, video conferencing and multi-media, network security, etc.

4.1 A Short History of Internet Protocols at CERN

Ben Segal, IT/PDP

Abstract

The beginnings of Internet at CERN. A reprint from Ben's original published in April 1995.

Now that the Internet has exploded in popularity on a world wide scale, with a major component of its success (the *World Wide Web*, see the URL `http://www.cern.ch/CERN/WorldWideWeb/WWWandCERN.html`) being developed at CERN, it seems a good time to look back and trace the history of the Internet at CERN. Even before the Web allowed Internet penetration in the most unexpected places, the presence of the Internet protocols at CERN had already encouraged their adoption not only in many other parts of Europe but also in such influential organizations as the ITU and ISO in Geneva.

Another reason for writing this history today is that it is almost exactly ten years ago that CERN named me as its first "*TCP/IP Coordinator*". The TCP/IP protocols (as Internet protocols were then called) had actually entered CERN a few years earlier, inside a Berkeley Unix system, but not too many people were aware of that event. In the computer networking arena, a period of 10–15 years represents several generations of technology evolution. Readers of this history will perhaps be surprised that in a period of only three years there can be developments that radically change the whole way that people think about computer communications. This has just happened with the Web (prototyped in 1990–1, fully accepted over 1993–4), but several related steps needed to take place beforehand to enable the Web's emergence. First of all, standards had to emerge in computer systems themselves, and in programming techniques. Next, standards were needed in network and computer hardware, with the accompanying price reductions. Finally there had to be a change in mentality, among both manufacturers and computer users, for them first to allow and then to insist that their systems should be able to communicate freely.

Another interesting element, apart from the rapidity of change, is the factor of accident or coincidence, often traceable to a personal event or a meeting of one or two people in critical circumstances. Bringing the Internet to CERN was not a simple business, although similar events probably occurred at other pioneer sites. Being very well acquainted with the people involved, the present author was ideally placed to observe the interplay of technical, personal and political elements at CERN that helped bring about a major part of today's Information Revolution.

In the Beginning — the 1970's

In the beginning was — chaos. In the same way that the theory of high energy physics interactions was itself in a chaotic state up until the early 1970's, so was the so-called area of "Data Communications" at CERN. The variety of different techniques, media and protocols used was staggering; open warfare existed between many manufacturers' proprietary systems, various home-made systems (including CERN's own "FOCUS" and "CERNET"), and the then rudimentary efforts at defining open or international standards. There were no general purpose Local Area Networks (LANs): each application used its own approach. The only really widespread CERN network at that time was "INDEX": a serial twisted pair system with a central Gandalf circuit switch, connecting some hundreds of "dumb" terminals via RS232 to a selection of accessible computer ports for interactive login.

CERNET, beginning in 1976, offered a fast file transfer service between a number of mainframes and minicomputers via 2Mbit/s serial lines using packet switching in a network of gateway nodes. Remote login (known as "virtual terminal service") was only supported to a single system, the central IBM mainframe. At the end of its ten year life CERNET supported 100 systems, including its own version of a LAN bridge, connecting some of CERN's first Ethernets. However, even though architecturally CERNET resembled ARPAnet, all its protocols had been developed independently. It was therefore doomed, though this was of course unknown at the beginning. Even if its designers had been in contact with Vint Cerf and company, there was no efficient way to run a transatlantic collaboration. Imagine a period without electronic mail. . . but no, this was only introduced at CERN to any extent at the beginning of the 1980's.

This was also a period without standards for computer systems themselves, not just communication systems. There are therefore few computer files from this period and we must rely on the written record. We must look back into our paper files, typed by our typists, back into a time when there were no PC's, no Macintoshes, no Unix and no C programming at CERN. . .

The Stage is Set — early 1980's

To my knowledge, the first time any "Internet Protocol" was used at CERN was during the second phase of the STELLA Satellite Communication Project, from 1981–83, when a satellite channel was used to link remote segments of two early local area networks (namely "CERNET", running between CERN and Pisa, and a Cambridge Ring network running between CERN and Rutherford Laboratory). This was certainly inspired by the ARPA IP model, known to the Italian members of the STELLA collaboration (CNUCE, Pisa) who had ARPA connections; nevertheless the STELLA Internet protocol was independently implemented and a STELLA-specific higher-level protocol was deployed on top of it, not TCP. As the senior technical member of the CERN STELLA team, this development opened my eyes to the meaning and potential of an Internet network protocol.

Ethernet made its appearance at CERN at about that time (1983), when an initial stretch of the soon-to-be-famous yellow cable arrived to support a demonstration of the very advanced Symbolics machine from MIT, which actually ran ChaosNet, XNS and TCP/IP protocols if I remember correctly. Before that, starting in 1982, we had installed some Apollo Domain coaxial cables for a 12 Mbit/s token ring network running Apollo's proprietary protocol, the first to offer a real distributed file system as well as network virtual memory paged over the ring.

The Apollo workstations brought CERN firmly into the distributed computing business. They had no equal for large scientific and graphics applications. The fact that the system was proprietary seemed unimportant at that time: Unix based competitors like Sun and HP were outclassed, and Unix itself had little appeal for CERN physicists. A file-exchange gateway was made (by this author) between the Apollos and the home-grown CERNET network, sufficient to exchange data and program files with the central CERN mainframes and other CERNET hosts. Remote login was made, if desired, by terminal emulation via RS232 cables; other connections were improvised by other techniques, in the disjoint spirit of those times.

In 1983, for the first time, a Data Communications (DC) Group was set up in the CERN Computing Division (then "Data-handling Division" or "DD") under David Lord. Before that time, work on computer networking in DD had been carried out in several groups: I myself belonged to the Software (SW) Group, which had assigned me and several others to participate in DD's networking projects since 1970. All my work on STELLA had been sponsored in this way, for example. The new DC Group seemed to have a mandate to unify networking practices across the whole of CERN, but after a short time it became clear that this was not going to be done comprehensively. DC Group decided to leave major parts of the field to others while it concentrated on building a CERN-wide backbone infrastructure. Furthermore, following the political currents of the time, they laid a very formal stress on ISO standard networking, the only major exception being their support for DECnet. PC networking was ignored almost entirely; IBM mainframe networking (except for BITNET/EARN) as well as the developing fields of Unix and workstation-based networking all remained in SW Group. So did the pioneering work on electronic mail and news under Dietrich Wiegandt, which made CERN a European leader in this field. (From the early 1980's until about 1990 CERN acted as the Swiss backbone for USENET news and gatewayed all Swiss e-mail between the EUnet uucp network, BITNET, DECnet and the Internet). As these were precisely the areas in which the Internet protocols were to emerge, this choice led to a situation in which CERN's support for them would be marginal or ambiguous for several years to come.

It was from around 1984 that the wind began to change.

TCP/IP Introduced at CERN

In August, 1984 I wrote a proposal to the SW Group Leader, Les Robertson, for the establishment of a pilot project to install and evaluate TCP/IP protocols on some key non-Unix machines at CERN including the central IBM-VM mainframe and a VAX VMS system. This was to decide if TCP/IP could indeed solve the problem of heterogeneous connectivity between the newer open systems and the established proprietary ones. It also proposed to evaluate Xerox's XNS protocols as a possible alternative. The proposal was approved and the work led to acceptance of TCP/IP as the most promising solution, together with the use of "sockets" (pioneered by the BSD 4.x Unix system) as the recommended API.

In early 1985 I was appointed the "TCP/IP Coordinator" for CERN, as part of a formal agreement between SW Group (under Les Robertson) and DC Group (under its new leader, Brian Carpenter). Incorporating the latter's policy line, this document specifically restricted the scope of Internet protocols for use only within the CERN site. Under no circumstances were any external connections to be made using TCP/IP: here the ISO/DECnet monopoly still ruled supreme, and would do so until 1989.

Between 1985 and 1988, the coordinated introduction of TCP/IP within CERN made excellent progress, in spite of the small number of individuals involved. This was because the technologies concerned were basically simple and became steadily easier to buy and install.

A major step was taken in November 1985 when the credibility of the Internet protocols as implemented within CERN was sufficient to convince the management of the LEP/SPS controls group that the LEP control system, crucial for the operation of CERN's 27 km accelerator LEP then under construction, should use TCP/IP. This decision, made by P-G. Innocenti, Jacques Altaber and Pal Anderssen, combined with a later decision to use Unix-based systems, turned out to be essential for the success of LEP. The TCP/IP activity in LEP/SPS included a close collaboration with IBM's Yorktown Laboratory to support IP protocols on the IBM token ring network that had been chosen for the LEP control system.

Other main areas of progress were: a steady improvement of the TCP/IP installations on IBM-VM/CMS, from the first University of Wisconsin version (WISCNET) to the fully-supported IBM version that still runs today; the rapid spread of TCP/IP on VMS systems, using third-party software in the absence of any DEC product; the first support of IBM PC networking, starting with MIT's free TCP/IP software and migrating to its commercial descendant from FTP Software. This latter work, by Mike Gerard and Brian Henningsen (who would later move to ISO in Geneva), led directly to the very comprehensive support of PC and Macintosh networking that exists today at CERN using TCP/IP, Novell and Apple protocols. All this was accompanied by a rapid change from RS232 based terminal connections to the use of terminal servers and virtual Ethernet ports using TCP/IP or DEC-based protocols. This permitted either dumb terminals or workstation windows to be used for remote login sessions, and hence to today's X-Windows. In particular, starting from 3270 emulator software received from the University of Wisconsin and developed by myself for Apollo and Unix systems, a full-screen remote login facility was provided to the VM/CMS service; this software was then further developed by Mike Gerard and used as the basis for the Terminal Access Gateway service (TAG) which became a standard way for CERN users to access VM/CMS systems world-wide.

As late as September 1987, DD's Division Leader P. Zanella would still write officially to a perplexed user, with a copy to the then Director of Research, J. Thresher: "The TCP-IP networking is not a supported service." This illustrates the ambiguity of the situation (already referred to above) as these words were written at essentially the same time as another major step forward was made in the use of Unix and TCP/IP at CERN: the choice to use them for the new Cray XMP machine instead of Cray's well-established proprietary operating system COS and its associated Cray networking protocols. Suddenly, instead of asking "What use is Unix on a mainframe?" some users began to ask "Why not use Unix on everything?".

The Cray represented CERN's first "supercomputer" according to US military and commercial standards and a serious security system was erected around it. As part of this system, in 1987 I purchased the first two Cisco IP routers in Switzerland (perhaps in Europe?), to act as IP filters between CERN's public Ethernet and a new secure IP segment for the Cray. I had met the founder of "Cisco Systems", Len Bosack, at a Usenix exhibition in the USA in June 1987 and been very impressed with his router and this filtering feature. Cisco was a tiny company with about 20 employees at that time, and doing business with them was very informal. It was hard to foresee the extent to which they would come to dominate the router market, and the growth that the market would undergo. Unfortunately I did not purchase any Cisco shares when a little later they went public…

Birth of the European Internet

In November 1987 I received a visit from Daniel Karrenberg, the system manager of "mcvax", a celebrated machine at the Amsterdam Mathematics Centre that acted as the gateway for all transatlantic traffic between the US and European sides of the world-wide "USENET", the Unix users' network that carried most of the email and news of that time using a primitive protocol called "uucp". Daniel had hit on the idea of converting the European side ("EUnet") into an IP network, just as major parts of the US side of USENET were doing at that time. The news and mail would be redirected to run over TCP/IP (using the SMTP protocol), unnoticed by the users, but all the other Internet utilities "telnet", "ftp", etc. would become available as well, once Internet connectivity was established. Even better, Daniel had personal contacts with the right people at the NIC who would grant him Internet connect status when he needed it. All he was missing was a device to allow him to run IP over some of the EUnet lines that were using X.25 - did this exist? I reached for my Cisco catalogue and showed him the model number he needed. Within a few months the key EUnet sites in Europe were equipped with Cisco routers, with the PTT's, regulators and other potential inhibitors none the wiser. The European IP network was born without ceremony.

CERN Joins the Internet

In 1988, the DC Group in DD Division (later renamed CS Group in CN Division) finally agreed to take on the support of TCP/IP, and what had been a shoestring operation, run out of SW Group with a few friendly contacts here and there, became a properly staffed and organized activity. John Gamble became the new TCP/IP Coordinator, performing this task until quite recently; he had just returned from extended leave at the University of Geneva where he had helped to set up one of the very first campus-wide TCP/IP networks in Europe. A year later, CERN opened its first external connections to the Internet after a "big bang" in January 1989 to change all IP addresses to official ones. (Until then, CERN had used an illegal Class A address, Network 100, chosen by myself).

CERN's external Internet bandwidth flourished, with a growing system of links and routers managed by Olivier Martin and Jean-Michel Jouanigot. Concurrently with the growth of the new European IP network (later to be incorporated as "RIPE" within the previously ISO-dominated organization "RARE"), many other players in Europe and elsewhere were changing their attitudes. Prominent among these was IBM, who not only began to offer a good quality mainframe TCP/IP LAN connection product of their own but also began to encourage migration of their proprietary BITNET/EARN network towards IP instead of the much more restricted RSCS-based service. They even began a subsidy programme called EASINET to pay line charges for Internet connection of their European Supercomputer sites of which CERN was one. In this way, the principal link (1.5 Mbit/sec) between Europe and the USA was located at CERN and funded by IBM for several years during the important formative period of the Internet.

By 1990 CERN had become the largest Internet site in Europe and this fact, as mentioned above, positively influenced the acceptance and spread of Internet techniques both in Europe and elsewhere. Brian Carpenter, still the leader of CS Group, is today a member of the Internet Architecture Board and a well known speaker at Internet gatherings. Experts like Olivier Martin and Jean-Michel Jouanigot are world authorities on Internet traffic and routing questions. CERN also facilitates the efforts of some staff members, including myself, who take time to teach Internet technology in developing countries.

The Web Materializes

A key result of all these happenings was that by 1989 CERN's Internet facility was ready to become the medium within which Tim Berners-Lee would create the *World Wide Web* with a truly visionary idea. In fact an entire culture had developed at CERN around "distributed computing", and Tim had himself contributed in the area of Remote Procedure Call (RPC), thereby mastering several of the tools that he needed to synthesize the Web such as software portability techniques and network and socket programming. But there were many other details too, like how simple it had become to configure a state of the art workstation for Internet use (in this case Tim's NeXT machine which he showed me while he was setting it up in his office), and how once on the Internet it was possible to attract collaborators to contribute effort where that was lacking at CERN.

Footnote

The above is a short and personal record and I may have missed out some people or events that deserve mention. I would be happy to hear from colleagues whose memories can add to this history, which does not have to be considered as my private property. Of course in the end I take responsibility for what appears under my name.

The key words that came to my mind while writing this history were: synergy, serendipity and coincidence. Many of life's most propitious happenings are very deeply "a question of timing". It is my personal belief that the Web could have emerged considerably earlier if CERN had been connected earlier to the Internet: the first Web proposal was written immediately after the opening of CERN's first external connection and it is known that Tim had been working with hypertext ideas since 1980, influenced by Ted Nelson's work on Xanadu among other things. But this remains in the realm of speculation. What is certain is that the Internet has provided a unique opportunity for some of us at CERN to take part in a series of events which are helping to change the world for countless people, hopefully for the better.

About the author(s): *See biographic note in the previous article, "*A SHIFT in Outlook*", in this CNL.*

4.2 A Short History of the World Wide Web From 1960s to 1995

Edited by Nicole Crémel, IT/User Support

2001 is the 10th birthday for the release of WWW at CERN!

*The following timeline shows the historic development of the beginning of the Web. It is based on a Web page available on the W3C (*World Wide Web Consortium*) Web site. The original article is at the URL `http://www.w3.org/History.html`.*

1960s

Doug Engelbart prototypes an "oNLine System" (NLS) which does hypertext browsing editing, email, and so on. He invents the mouse for this purpose [1].

Ted Nelson coins the word Hypertext in "*A File Structure for the Complex, the Changing, and the Indeterminate*", 20th National Conference, New York, Association for Computing Machinery [2–4].

Andy van Dam and others build the Hypertext Editing System and FRESS in 1967.

1980

While consulting for CERN June–December of 1980, Tim Berners-Lee writes a notebook program, "*Enquire-Within-Upon-Everything*", which allows links to be made between arbitrary nodes. Each node had a title, a type, and a list of bidirectional typed links. "Enquire" ran on Norsk Data machines under SINTRAN-III.

1989

In March 1989 (later redistributed unchanged apart from the date added in May 1990), the document "*Information Management: A Proposal*" [5], written by Tim Berners-Lee was circulated for comments at CERN.

1990 — or the birth of *World Wide Web*

May
: The proposal [5] is recirculated (see above).

September
: Mike Sendall (Tim's manager at CERN) accepts the purchase of a NeXT cube, and allows Tim to go ahead and write a global hypertext system.

October
: Tim starts work on a hypertext GUI browser+editor using the NeXTStep development environment [6].
: First names for the project were "Mesh", "Mine of Information", and "Information Mine", but Tim decided on "World Wide Web" when writing the code in 1990.
: The original Project proposal was reformulated with encouragement from CN and ECP divisional management.

November
: Robert Cailliau (ECP) is a co-author of the new version of the proposal (`http://www.w3.org/Proposal.html`, dated 12 November 1990).
: Initial *WorldWideWeb* program development continues on the NeXT. This was a WYSIWYG browser/editor

with direct in-line creation of links. Technical Student Nicola Pellow (CN) joins and starts work on the line-mode browser. Bernd Pollermann (CN) helps to get an interface to the CERNVM "FIND" index running. Tim Berners-Lee gives a colloquium on hypertext in general.

December (Christmas)

Line mode browser and *WorldWideWeb* [7] browser/editor are demonstrated: access is possible to hypertext files, CERNVM "FIND", and Internet news articles.

1991 — the first CERN release for WWW

February

Workplan is built for the purposes of ECP division. On 26 February 1991 a presentation [8] of the project is made to the ECP/PT group.

March

The line mode browser (WWW) is released to a limited audience on "PRIAM" Vax, RS6000, Sun4.

May

A new work-plan is produced for the CERN CN/AS group. On 17 May 1991 a Presentation [9] is made to the "C5" Committee. This lead to a *general release of WWW on the central CERN machines*.

June

On 12 June 1991 a CERN Computing Seminar [10] is given on "Hypertext" and the "WorldWideWeb" program.

August

Files are available on the net by FTP, posted on the Internet newsgroup `alt.hypertext` (6, 16, 19th Aug), `comp.sys.next` (20th), `comp.text.sgml` and `comp.mail.multi-media` (22nd). Jean-Francois Groff joins the project.

October

VMS/HELP and WAIS gateways are installed. The CERN mailing lists `www-interest` (now `www-announce`) and `www-talk@info.cern.ch` have been created [11].

December

A poster and demonstration are presented at the conference Hypertext'91 [12] in San Antonio, Texas (US). A W3 browser is installed on VM/CMS.

CERN Computer Newsletter (CNL) 204 announces W3 to the HEP World!

On December 12, Paul Kunz installs the first Web server outside of Europe, at SLAC.

1992 — the first GUI browsers for X-Windows

January

On January 15 the line mode browser release 1.1 is available by anonymous FTP, and any FTP site becomes a W3 information source [13]. A presentation is made to AIHEP'92 at La Londe (FR).

February

Line mode browser release 1.2 is announced on various newsgroups and mailing lists.

April

On April 29 the Finnish "*Erwise*" GUI client for X is released with comments [14] made by Tim Berners-Lee.

May

Pei Wei's "*Viola*" GUI browser for X test version is available on May 15 ([15] contains review by Tim). Technical Student Carl Barker (ECP) joins the project.

June

Presentation and demo at HEPVM (Lyon, FR). People at FNAL (Fermi National Accelerator Laboratory, USA), NIKHEF (Nationaal Instituut voor Kern- en Hoge Energie Fysika, NL), DESY (Deutsches Elektronen Synchrotron, Hamburg, DE) join the project with WWW servers.

July

Distribution of WWW (including *Viola*) is made through the "CERN Program Library" (CernLib) standard procedure. The WWW library code is ported to DECnet.

September

Plenary session demonstration made to the HEP community at CHEP'92 in Annecy (FR).

November

Look at a snapshot of the "*World Wide Web*" Project Home page [16] at that time, including the list of all 26 reasonably reliable Web servers [17] (NCSA's having just been added, but no sign of *Mosaic* yet).

1993 — the birth of NCSA *Mosaic*

January

By now, *Midas* (Tony Johnson, SLAC), *Erwise* (HUT), and *Viola* (Pei Wei, O'Reilly Associates) browsers are available for X; a CERN Mac browser (ECP) is released as *alpha*. There are around 50 known `HTTP` servers.

February

NCSA (National Center for Supercomputing Applications, University of Illinois (US)) releases the first alpha version of Marc Andreessen's "*Mosaic for X*". Another "computing seminar" is given at CERN.

March

WWW (Port 80 `HTTP`) traffic measures 0.1% of NSF backbone traffic. WWW is presented at "*Online Publishing 93*", in Pittsburgh.

April

April 30: declaration by CERN's directors that WWW technology would be freely usable by anyone, with no fees being payable to CERN.

July

Ari Luotonen (ECP) joins the project at CERN. He implements access authorisation, proceeds to re-write the CERN `HTTPD` server.

August

O'Reilly hosts the first *WWW Wizards Workshop* in Cambridge Mass. (US).

September

WWW (Port 80 `HTTP`) traffic measures 1% of NSF backbone traffic. NCSA releases working versions of *Mosaic* browser for all common platforms: X, PC/Windows and Macintosh.

October

Over 200 known `HTTP` servers. The European Commission, the *Fraunhofer Gesellschaft* and CERN start the first Web-based project of the European Union (DG XIII): *WISE*, using the Web for dissemination of technological information to Europe's less favoured regions.

December

WWW receives IMA award. John Markov writes a page and a half on WWW and *Mosaic* in "The New York Times" (US) business section. "The Guardian" (UK) publishes a page on WWW, "The Economist" (UK) analyses the Internet and WWW.

Robert Cailliau gets go-ahead from CERN management to organise the *First International WWW* Conference at CERN.

1994 — birth of *Netscape*, CERN ends WWW development

January

O'Reilly, Spry, etc., announce "*Internet in a box*" product to bring the Web into homes.

March

Marc Andreessen and colleagues leave NCSA to form *Mosaic Communications Corp.* (later *Netscape*).

May

On May 25–27 the *First International WWW Conference* [18] takes place at CERN, Geneva, and is heavily oversubscribed (800 apply, 400 allowed in). This first meeting has been called the "*Woodstock of the Web*". VRML was conceived at the same conference.

June

M. Bangemann produces a report for the *European Commission Information Superhighway* plan. Over 1500 servers are now registered and the load on the first Web server (`info.cern.ch`) is 1000 times higher than is was 3 years earlier.

July

The MIT/CERN agreement to start the *W3 Organisation* is announced by M. Bangemann in Boston (press release in Wall Street Journal, Boston Globe, etc.)

August

Founding of the IW3C2: the *International WWW Conference Committee*, in Boston, by NCSA and CERN.

September

The European Commission and CERN propose the *WebCore* project for development of the Web core technology in Europe.

October

The second International WWW Conference, "*Mosaic and the Web*", takes place in Chicago and it is again heavily oversubscribed (2000 apply, 1300 allowed in).

December

On December 14 the first W3 Consortium Meeting [19] takes place at M.I.T. in Cambridge (USA).

On December 15 there is the first meeting with European Industry and the European Consortium branch, at the *European Commission*, Brussels.

On December 16 CERN Council approves unanimously the construction of the LHC (Large Hadron Collider) accelerator, CERN's next machine and competitor to the US' already defunct SSC (Superconducting Supercollider). Stringent budget conditions are however imposed. CERN thus decides not to continue WWW development, and in concertation with the European Commission and INRIA (*Institut National pour la Recherche en Informatique et Automatique*, FR) transfers the *WebCore* project to INRIA.

1995

February

The Web is the main reason for the theme of the G7 meeting hosted by the *European Commission* in the European Parliament buildings in Brussels (BE).

March

CERN holds a two-day seminar for the European Media (press, radio, TV) [20], attended by 250 reporters, to show WWW. It is demonstrated on 60 machines, with 30 pupils from the local International High School helping the reporters "surf the Web".

April

The third International WWW Conference, "*Tools and Applications*" [21], is hosted by the Fraunhofer Gesellschaft, in Darmstadt (DE).

June

Founding of the *Web Society* in Graz (AT), by the Technical University of Graz (home of Hyper-G), CERN, the University of Minnesota (home of Gopher) and INRIA.

If you want to read further

- *"Weaving the Web"* (`http://www.w3.org//People/Berners-Lee/Weaving/Overview.html`), T. Berners-Lee, Harper Collins 1999
- *History of Internet and WWW: The Roads and Crossroads of Internet History* (`http://www.internetvalley.com/intval1.html`), 1995-1998 by Gregory R. Gromov
- *How the Web Was Born* (`http://www.oup.co.uk/isbn/0-19-286207-3`), Oxford Univ. Press 2000, or *Die Wiege des Web* (`http://www.dpunkt.de/buecher/3-89864-108-2.html`), dpunkt.verlag 2001, James Gillies and Robert Cailliau.
- List of Internet Histories (from ISOC) [22].

References and Links

[1] *Bootstrap Institute library (1960s).* (`http://www.bootstrap.org/library.htm`).

[2] *Association for Computing Machinery (1965).* (`http://www.acm.org/`).

[3] *Literary Machines (1960s).* (`http://www.sfc.keio.ac.jp/~ted/TN/PUBS/LM/LMpage.html`).

[4] *Hypertext and hypermedia: a selected bibliography (1991).* (http://www.lcc.gatech.edu/~harpold/papers/ht_bibliography/history.html).
[5] *Information Management: A Proposal (1989).* (http://www.w3.org/History/1989/proposal.html).
[6] *"The WorldWideWeb browser" and "screenshot" in 1990.* (http://www.w3.org/People/Berners-Lee/WorldWideWeb.html and http://www.w3.org/History/1994/WWW/Journals/CACM/screensnap2_24c.gif).
[7] *What were the first WWW browsers? (1990).* (http://www.w3.org/People/Berners-Lee/FAQ.html#browser).
[8] *CERN Presentation of the project (26 February 1991).* (http://www.w3.org/Talks/Title.html).
[9] *CERN Presentation to "C5" (17 May 1991).* (http://www.w3.org/Talks/C5_17_May_91.html).
[10] *CERN Computing Seminar on WWW (12 June 1991).* (http://www.w3.org/Talks/ComputerSeminar.html).
[11] *CERN www-talk 1991 mailing list archives.* (http://ksi.cpsc.ucalgary.ca/archives/WWW-TALK/www-talk-1991.index.html).
[12] *Hypertext'91 Conference.* (http://www.w3.org/Conferences/HT91/Overview.html).
[13] *News for FTP access (1992).* (http://www.w3.org/News/9201.html).
[14] *Tim's report on* "Erwise" *(first GUI client for X, 1992).* (http://www.w3.org/History/19921103-hypertext/hypertext/Erwise/Review.html).
[15] *Tim's report on* "Viola" *(GUI browser for X, 1992).* (http://www.w3.org/History/19921103-hypertext/hypertext/Viola/Review.html).
[16] *"World Wide Web" Project Home page (November 1992).* (http://www.w3.org/History/19921103-hypertext/hypertext/WWW/TheProject.html)
[17] *List of the 26 Web servers in November 1992.* (http://www.w3.org/History/19921103-hypertext/hypertext/DataSources/WWW/Servers.html).
[18] *First International WWW Conference (1994).* (http://www.cern.ch/WWW94/).
[19] *The World Wide Web Consortium: Prospectus.* (http://www.w3.org/Consortium/Prospectus/Overview.html).
[20] *CERN World-Wide Web Days for the European Media.* (http://www.cern.ch/CERN/WorldWideWeb/mediaday/).
[21] *Third International WWW Conference "Tools and Applications".* (http://www.igd.fhg.de/www/www95/www95.html)
[22] *List of Internet Histories (from ISOC).* (http://www.isoc.org/internet/history).

5. Scientific Applications and Software Engineering

This includes descriptions of: scientific applications and libraries from CERNLIB and LHC++, mainly used by the Physicists;engineering applications likes the ones on PaRC or those associated to online controls;software development environment.

5.1 Programming Languages over the Years

Julian Blake, IT/PDP

Abstract

This is a personal tour of programming languages which I have encountered since I first learnt to program a computer in 1959.

Assembly codes

Assembly languages have changed over the years, starting very simple (binary or decimal instruction codes and operands), growing to accept mnemonic operation codes and operands in place of numeric ones, and adding macro and conditional assembly features. Nowadays programming in assembly language is rarer than it used to be, and rather simple mnemonic assemblers, sufficient to process the output of a compiler, are used.

The English Electric DEUCE was programmed in binary (one 32-bit word to each row of a punched card). Each instruction had to specify the location of the next instruction to be obeyed, and the way to get a fast program was to place instructions in the mercury delay lines such that there was no unnecessary waiting between instructions. An interesting technical challenge was to write a bootstrap program of twelve instructions on a single punched card. During my time as a pre-university student at the English Electric Company, a staff member was in the process of writing an assembler which would, *inter alia*, look after instruction placement.

The EDSAC 2 and Titan (ATLAS 2) computers at Cambridge University had decimal assemblers. Both operation codes and operands were specified by decimal numbers. Both computers had built-in firmware to perform common tasks, for example conversion between binary and decimal. The order code of the Titan was designed so that it would be relatively easy to remember the decimal operation codes (the regularities in the order code showed most clearly in decimal).

The most powerful assembler with which I have worked was Control Data's COMPASS macro assembler for the CDC 6000 and 7000 Series central (CP) and peripheral (PP) processors. Operations and register operands were specified by mnemonics. CP instructions were written in a particularly user-friendly form: "SA1 A0+B1" denoted "set address register A1 to the sum of address register A0 and index register B1" (this initiated a read from central memory address A1, as reading was the function of registers A1 to A5). COMPASS was a classical two-pass assembler with macro and conditional assembly features, and generated a full listing showing both the source assembly code and the generated machine code (in octal). CDC's operating systems were written almost entirely in COMPASS assembly language. So too, at CERN, was the software for the Remote Input/Output Stations (RIOS, running on Modular 1 computers) and for SUPERMUX (terminal concentrators running on HP 2100 computers); the technique was to define a set of COMPASS macros for the target computer's order code, and to post-process the output of COMPASS to shorten 60-bit words to the word length of the target.

Several cross assemblers were written at CERN:

- in BCPL for the ModComp computers of the CERNET packet switching network, and for the I8080, M6800 and TMS9900 microprocessors;
- in Pascal for the M6800/6801/6809 and M68000/68020/68030+68881 microprocessors.

This list is not exhaustive. There were certainly others, for example the cross assemblers for the ESOP and XOP bit slice processors.

Structured assembly languages

Structured assembly languages provide a mixture of access to machine specific features such as registers and addressing modes, and high level language constructs such as begin-end blocks, procedures, if and case statements, and for and while loop constructs. The first such language was Niklaus Wirth's PL/360. At CERN, Robert Russell created PL-11 (for PDP-11) and later PL-VAX.

I had brief contact with PL-VAX, and with the PL/M language for Intel microprocessors.

Autocodes

My first programming language was DEUCE Alphacode. The language provided a set of floating point variables (X1, X2, ..., X2200) and a smaller number of counting (integer) variables (N1, N2, ..., N63). One line of code could perform a single operation, for example "`X1 = X2 + X3`" or "`X4 = ROOT X5`". The statements of Alphacode were usually interpreted, not compiled. Writing and using an Alphacode program was an improvement on performing pre-specified calculations on an electro-mechanical calculating machine, my previous activity at English Electric.

EDSAC 2 and Titan Autocode at Cambridge University supported operations on a fixed set of integer variables (I, J, ...) and of floating point arrays (A[5], B[7], ...). Autocode was a compiled language. Since I was interested in algorithms for syntactic analysis of programming languages at that time, the lack of arrays of integers ruled out the use of Autocode during my doctoral studies.

Algol 60

I was introduced to Algol 60 while working as a summer student at Elliott Automation, using the Elliott 803 computer. I remember being amazed when shown a recursive algorithm for the travelling salesman problem, and discovering that Algol did indeed allow a function to call itself.

FORTRAN

I first encountered FORTRAN IV when I arrived at CERN. How could a computer programmer live to the age of 26 years before using FORTRAN? The answer lies in the traditions of Cambridge University's Mathematical Laboratory, now renamed the Computer Laboratory. Shortly after I left Cambridge, the University bought an IBM computer for its computing service, thereby introducing FORTRAN.

FORTRAN was the most important computer language at CERN, so much so that people used to say, only half jokingly, that whatever computer language physicists were using in five, ten or fifteen years time, it would be called FORTRAN. Only when CERN decided to move to C++ instead of making the transition from FORTRAN 77 to Fortran 90, did the use of FORTRAN decline.

Although Pascal and C were the most frequently used high level languages on microprocessors at CERN, FORTRAN was used too. An extreme example was Hans von der Schmitt's RTF/68K (Real Time Fortran 77 for 68K Processors), whose compiler was written in a compiler description language which was translated into FORTRAN by a compiler generator, also written in FORTRAN.

SYMPL

SYMPL was an Algol-like systems implementation language created by Control Data for its 6000 and 7000 Series computers. I used it to write the CP (central processor) part of the 6000 Series software for CERNET; the PP (peripheral processor) part was written in COMPASS assembly language. Dietrich Wiegandt, writing CERNET software for the IBM System 370, had to write it all in assembly language. I was the more fortunate person.

BCPL

Martin Richards' BCPL is known as an ancestor of the C programming language. BCPL was used at CERN to write CERNET node software for ModComp II computers, and CERNET host software for PDP-11 and Nord 10. The BCPL compiler was written in BCPL. It ran at CERN as a native compiler for CDC 6000, IBM 370, VAX, PDP-11, Nord 10, HP 2100; and as a cross compiler for ModComp II and TMS9900. Cross assemblers, a linker, a librarian and preloaders ("pushers") were written in BCPL for ModComp, I8080, M6800 and TMS9900.

Ada

Ada 83 was tried at CERN by an *ad hoc* evaluation group, but never went much further. It was however used to write the error message module of the MODEL data acquisition system. One Ada critic felt that the inter-task rendez-vous mechanism was unsuitable for real-time data acquisition systems. My own experience was that type checking was so strict that even carefully constructed code could be rejected by the compiler for no understandable reason.

Pascal

The second generation of CERN cross software for microprocessors was written in Pascal. This included cross assemblers for M6800/6801/6809 and M68K (M68000/68020/68030+68881), and M68K cross compilers for Pascal, Modula-2, FORTRAN 77 and C (unfinished). The FORTRAN cross compiler was little used as there were well established UNIX-derived f77 and C cross compilers. MoniCa, a debugging monitor for M68K, was written in Pascal and assembly language. MoniCa was used in the ALEPH event builder, in VALET-Plus test systems, and elsewhere. The Pascal cross compiler, which originated in Siemens research laboratories and was extended at CERN to become a multi-language compiler, was sufficiently good to see off competition from some commercial Pascal compilers. A few years later, however, DD divisional management decided that the time had come to stop developing microprocessor cross software.

At the height of its use at CERN, Pascal was considered a safe language in which to write programs, whereas C and BCPL were considered dangerous.

Modula-2

Modula-2, Niklaus Wirth's successor to Pascal, was used on M68K hardware in the LEP control system.

At about the same time, Robert Cailliau in PS division was promoting the use of his enhanced Pascal, P+.

In both cases, necessary Pascal extensions included facilities for separate compilation and for bit operations.

Modula-2 was capable of running on a microprocessor without operating system, since its SYSTEM module provided the minimum necessary for accepting interrupts and switching context.

C

C came to CERN with UNIX (on PDP-11 and then on VAX), and with the OS-9/68K operating system (on M68K) which was used in LEP experiments' data acquisition systems. It is now the dominant language for control and data acquisition systems. Much code which used to be written in assembly language is today written in C.

The introduction into C of function prototypes, and the existence of instrumentation and debugging tools such as Purify and Insure++, has made programming in C less hazardous than it used to be.

The GNU C compiler has set a high standard, below which no commercial C compiler is viable.

C is the programming language with which I work most often today.

C++

C++ has taken over the role of FORTRAN as CERN's scientific data processing language. I have followed the development of C++ with interest, have helped to solve some problems with the use of C++, but have not become a practising C++ programmer. So take my views with a pinch of salt.

The definition of C++ and its run time library has taken a long time to stabilize. Early users will have discovered how much the language and library have changed over the years. Some of the newer features are still not correctly implemented by current compilers. C++ is so powerful that a programmer can write programs which his colleagues, and perhaps he himself, cannot understand. So select the C++ features which you use with care, look at some of the generated code (it may be more than you expect), and use a good debugging tool (such as Insure++).

Java

I admire Java and its class libraries, have been to a Java programming course and read the nutshell book, but do not actually write Java programs today. So my views on Java may be worth even less than my views on C++. Let's simply say that I like Java better than C++, and I think its use will spread well beyond its current application area of Web-based human interfaces.

Acknowledgments

In this article I have touched on the work of too many CERN colleagues to mention every one of them by name. Ian Willers was the driving force behind the use of BCPL at CERN. Horst von Eicken drove the second generation of microprocessor support software. Jean Montuelle, David Foster and Jonathan Caves made major contributions to the microprocessor cross software.

About the author(s): *Julian Blake was the leader of the DD/SW/LM (languages and microprocessor support) section before it was dissolved.*

5.2 A Brief History of Cernlib (1966–????)

Ian Mclaren, IT/API

Abstract

Past, evolution and future for CERNLIB.

Sixties (the beginning)

Although there is no mention of the Cern Program Library in CNL 1, the first recorded submissions of kernlib and mathlib routines were in the same year, 1966. At that time, the Computer Centre had CDC 60-bit computers and the first library routines were of a mathematical nature. The Physics specific programs were developed by their users in the Track Chamber (TC) Division, including the first version of PATCHY, also in 1966. The following year saw the birth of MINUIT. During the last years of the sixties, many additions were made to both libraries, until in 1969 the administration of the TC library was merged with the DD Cern Program Library, and PATCHY3 and its manual were released.

Seventies (CDC era)

The CDC era continued well into the seventies and more famous projects were born, GEANT3 and HBOOK in 1974 with HPLOT following in 1976, based on one of the first device independent graphics packages, GD3, which was also developed in DD Division. The average rate of new submissions continued at about 8/year, mostly written in Fortran but increasingly also in Compass, the CDC assembly language, to improve performance. With little interactive computing and no WWW, the link between the Program Library and the Computing Newsletter was very strong. All official library notifications were made in the CNL and users were deemed to have read and taken note of them.

The library source was maintained using a CDC proprietary code management utility called Update, which was considered at that time to be easier to use than PATCHY. Despite this, the libraries were increasingly exported to external institutes and the first tools to assist in converting to double precision began to appear. However, the importance of the IBM 360 and its 32-bit architecture for scientific computing increased and with it the problems of portability and the need for double precision floating point arithmetic which was unnecessary on the 60-bit CDC machines.

In 1977, IBM returned to the Cern Computing Centre with the opening of the 370/168 service, the PATCHY4 suite was released and the Program Library sources were converted from Update to PATCHY. The name "`cernlib`" appeared for the first time in the name of the IBM library "`CR.PUB.CERNLIB`" and the IBM version of the new

HBOOK3 was announced.

In the following year the definitive standard Fortran 77 was published and since then the Cernlib Fortran source has adhered to it as closely as possible. To supplement the deficiencies in Fortran, the dynamic memory management package ZBOOK was added to the Program Library.

Eighties (Increasing complexity)

The decade started with libraries for two flavours of CDC and one IBM, but by 1983 there were also versions for Cray, Vax, Apollo and Nord, laying the foundations for the Unix versions. There were many additions in the mathematical parts of the libraries, to add new functionality and new systems, and also conversions to standard Fortran 77. Consequently, there were many deletions and most of the CNL's had many pages with Program Library modifications and news. However, in 1984 the Fortran 66 libraries were finally frozen, the 7600 was turned off and the CERN directorate made Cernlib freely available to any laboratory involved in the CERN program.

1986 was a noteworthy year. Although ZBOOK had been offered as the dynamic memory management package, a different system, HYDRA, developed in TC division was also widely used. The ZEBRA project was an effort to combine the best points of these two packages and culminated in the addition of ZEBRA to Packlib. The first ISO graphics standard GKS, was starting to replace the ageing GD3 and a new version of HPLOT was using it. The Christmas Newsletter had the first big article on PAW which would be released the following year. In addition to new versions of HBOOK and HPLOT it required two new packages, KUIP (a command interpreter) and HIGZ, a graphical interface to GKS using ZEBRA for data management.

As HBOOK4 and HPLOT5 were incompatible with their predecessors, the last years of the decade were difficult for everyone; users had to convert their applications and parallel libraries had to be maintained. To facilitate change management the "old, pro and new" structure was introduced and the cernlib command extended to all CERN supported platforms. The PAW suite was completed with SIGMA in 1988 and COMIS in 1989. After almost 20 years since its introduction, GD3 was deleted. The last CNL of the decade described the new FATMEN File and Tape Management System, putting in place all the major computing components for LEP which was just starting operation.

Nineties (Unix and C++ arrive)

Although there were few Unix workstations at CERN, the library source was accumulating machine specific PATCHY sections, much of it written in C to interface to the Unix input/output streams and the X-window system. There was growing interest in new Monte Carlo event generators and the Program Library was becoming a redistributor for their authors.

Following the Cernlib users meeting in 1991, the decision was made to break the tradition of synchronizing Cernlib releases with the CNL. With 10 platforms to support, it had become impossible to make reliable library updates with every CNL and interactive newsgroups were well established as a communications medium. The new naming scheme contained the year instead of the CNL number so that v92a was the first release in 1992. Over the years, the mathematical parts of the library had become somewhat irrationally spread over kernlib and genlib so a major reorganisation was made with the creation of mathlib and the suppression of genlib. 1992 also saw the introduction of the database management package for HEP (HEPDB) and the freezing of GKS after a useful but rather short life. The Internet had now become the preferred method of distribution of Cernlib material. Probably the last major new functionality to be added was the PAW interface to Motif in 1993, when PAW++ for Unix was released.

The mood amongst the experiments was now turning to the object oriented paradigm and away from Fortran and the first CNL article on C++ and CLHEP was published in 1994, the last year to have more than one official Cernlib release. Cernlib was now a truly global service and there was pressure from our external users to move away from PATCHY, which was perceived as difficult to use and install. Despite the introduction of the more modern PATCHY5 the Program Library sources were converted to use the C preprocessor together with the public domain code versioning system CVS. The last PATCHY based release was made on 12 platforms at CERN in 1995.

Although declared "frozen" in 1998, Cernlib continued to enjoy popularity amongst a more diverse community, probably due to a very successful port to Linux on PC and also to relaxation of the distribution rules. Despite the great age of most of the code, only a modest effort was required to make it Year 2000 compliant.

The New Millennium

Our current plans are to have a final release in 2002 on a declining set of platforms with enough support to finish the analysis of the LEP data in 2003.

About the author(s): *Ian is the CERN Program Librarian since 1997 (see the article in CNL 225 at the URL `http://consult.cern.ch/cnls/225/art_pgl.html`).*

5.3 The Design and Implementation of Large Packages: Hbook, GEANT3, PAW, ROOT

René Brun, IT/DI

Abstract

The software development evolution as seen by a CERN Physicist involved in the development of Physics software since the very beginning of his professional life.

When the CNL editor kindly suggested me to write an article for this special edition of the Newsletter, I had a long hesitation for two reasons:

- I still feel myself pretty young, but have to admit that I have been floating around for three decades by now. At a time where many Internet startup/down managers are teenagers, I thought that may be the CNL readers will not be interested by stories of the kind "How good was the time when my grand mother was making tasteful strawberries jams"
- the second and main reason was that when you have been involved in the development of large packages used at a large scale in our field of research and others, you know very well that the true story involves not only technical matters but also and mainly sociological events with people moving between projects, sections, groups and divisions.

At the end, I accepted to write this note because I always had a fantastic opportunity to work in a very challenging area in contact with thousands of people in the world. During this time, I have been the witness of a tremendous evolution of the computer industry and also of the way large software systems are developed.

The seventies

In 1971, I was lucky to be accepted as a summer student. At the time, I was preparing my thesis in a small experiment at the SC evaluating the differential cross-sections in interactions between protons and thin targets from Beryllium to Gold. The events consisted of only two quantities, the energy of a particle absorbed by a NAI block and the dedx information collected by a germanium diode. The events of two bytes were recorded on rolls of paper tape, about 1 cm per event and 10,000 events on a roll. To analyze the events, histograms and scatter plots of dedx versus energy had to be produced. That is how I started a long career with histograming packages. The paper tape rolls were "manually" converted to temporary disk files with a life time of a few days and analyzed on the CDC6600 in the room where the printing shop is today. The analysis program consisted of about one "box" of 2000 punched cards with about 50 Fortran subroutines. The name of the routine was written with a pen on the top side of the cards. The size of the subroutine was carefully chosen such that there were enough cards to make the name visible and the order of routines in the box corresponded in general to the logical sequence of execution. The size of the executable module was typically a few tens of Kilobytes.

Running a job was a long process consisting of the following steps:

- put your box in one of the input trays X, S, M, L, T (express, short, medium, long jobs, jobs with tapes). The jobs were queued in each tray;
- an operator reads the box in the card reader and controls the execution;
- the result of the job is written to the line printer output;
- Once an hour, the line printer outputs are sorted in special racks.

Starting in 1972, a new and faster facility (FOCUS) was made available. You could enter yourself your program in the card reader and could get the output on a line printer in the user area. While your job was running, you could find the status by typing a command of the style `Job inquiry brun 1287`. Waiting for the job completion, you had plenty of opportunities to discuss with your colleagues and the cafeteria was just near by.

Immediately after having completed my thesis, I was hired in the DD division to work on Special Hardware processors to be used in a p-p elastic experiment at the ISR. Due to problems with the hardware, I migrated gradually to software activities. At the fall of 1973, the first version of *HBOOK* was available and this was the beginning of a long saga.

With my colleagues Michel Hansroul and Jean-Claude Lasalle, we were working on the so-called *Electronic Experiments* (as opposed to *Bubble Chamber Experiments*). We started together the development of several packages. I will only name a few (FFread, Lintra, Mudifi, Vertex). In 1974, we launched a framework for simulation, initially called *Geneve* (*Generation of Events*). When we discovered that this name was already used by at least another program, we changed the name to *GEANT* (*Generation of Events and Tracking*). After a first rudimentary version 1, we came with a more reasonable package *GEANT2* in 1977.

During this period, most users were still using punched cards. I had already switched to a more interactive system using a teletype. The editor was very primitive, but I could do everything from the teletype. In 1975, I was happy to have exclusive access to a huge but noisy Tektronix 4002 device. This display was connected to the FOCUS CDC3100 system (a front-end to the CDC6600). Running a session was a complex operation:

- To login, I had to type a long message with my name and a tape number. Following this message, a lamp was flashing in the computer center. The lamp was to attract the attention of the operators to mount a tape and copy the contents of my previous session to a local disk of a total capacity of about 10 Mbytes.

- I could use an advanced editor and submit my jobs interactively to the CDC6600. I could also inspect the graphics metafile produced by the job, a huge progress.
- To logout, the same lamp was again flashing to instruct the operator to copy my files to tape, the disk files were deleted and the space used by another client. Only a few people (less than 5) could use the system concurrently.

A major step came with the Intercom system. Intercom was a replacement of FOCUS using a CDC6400 and a CDC6500 as a front-end to the CDC7600. At the same time, I switched from my good old Tektronix4002 to a more lightweight display a Tektronix 4006. I developed the HBOOK and GEANT2 systems in this environment. We were always struggling with the memory limitations on the CDC7600 system and were experts in building large programs using overlays and the LCM (Large Core Memory) of the machine.

Our products were well documented. The first Hbook manual was produced in 1974 with a *text processor* called *BARB*. The input and output characters to *BARB* were exclusively uppercase characters. A better version *BARBascii* available in 1975 allowed upper/lowercase text in the output.

The eighties

The memory size limitations and the growing complexity of our programs pushed us to implement *Memory Managers*. I implemented a very simple system called *ZBOOK* (8000 lines) that we used both as a memory manager and a data structure manager. At the same time a more powerful but also complex system *HYDRA* had been designed for the bubble chambers software.

The *GEANT2* system had been in use quite successfully in several experiments, but was lacking a very important component, a geometry package. Pushed by my experience in the simulation and reconstruction programs of the deep inelastic muon scattering experiment NA4, I developed several prototypes of geometry packages before finding an acceptable solution. I was lucky to meet *Andy McPherson* from OPAL to help me implementing the package. *GEANT3* was born in 1981. At the time, I was far from imagining that the system would survive for more than two decades.

The move from the painful Intercom/Scope/CDC7600 to the user friendly VAX/VMS environment made possible the rapid development of *GEANT3*. The OPAL Vax/780 from RAL with 1 MByte of memory and 66 Mbytes disk plus the nice compiler and debugger was an essential step to develop a complex software. By the end of 1982, we had made a lot of progress in developing the simulation program for OPAL. L3 joined the effort in participating to the development of the tracking system with Francis Bruyant and later of the hadronic physics processes with Federico Carminati.

GEANT3 was based on the *ZBOOK* system. An official project called *GEM* had been launched in 1981 with the goal to deliver an enhanced data structure management system in view of LEP. GEANT3 was waiting for *GEM*. However, because of multiple problems that I have seen repeating multiple times, this ambitious project was stopped. This generated hot discussions between the experiments. In 1983, following the *GEM* problems, and strongly pushed by L3, an agreement was reached with *Julius Zoll* who was the main developer of *HYDRA* to create a new system that we named *ZEBRA* (ZBOOK + HYDRA). We quickly replaced the HBOOK and GEANT3 data structure managers by *ZEBRA*.

In June 1982, I was pleased to accept an invitation to visit the Apollo company in Boston. I immediately realized the potential and the advantages of multiple windows on the same screen and of a user friendly operating system. In September 1982, I had the first Apollo workstation in Europe in my office (half a Megabyte of memory and 33 Mbytes of disk). This again strongly contributed to speed up the development of *GEANT3*.

At the same time, I had been involved in an interactive version of *HBOOK*, called *HTV*. *HTV* used a small command interpreter called *ZCEDEX* and could be used to produce plots by calling interactively the *HBOOK* and *HPLOT* systems. A special version working only on Apollos (HTVGUI) was developed. It offered more interactivity by exploiting the bitmap functionality of the workstation. *HTV* had a growing number of users and in 1985 a group of people started a series of meetings to prepare the ground to design a successor that Rudy Böck named *PAW*, the "*Physics Analysis Workstation*". The birth of *PAW* had been a difficult exercise. There was a conflict between those pushing to adopt standards (*GKS* in particular) and associated commercial systems and those pushing for a more integrated and home-grown system. If we had followed the recommendations of the design committee, *PAW* would have had a primitive command line interface, a menu system using the SMG (Screen Management Graphics) designed for a VT100, a GKS-only graphics interface and no Ntuples. An attempt was made to setup a collaboration with Eric Bashler from DESY who had designed a popular package called *GEP*. Unfortunately, *GEP* was written in PL1, a language popular at the time but with a restricted portability. It was clear that we had to develop a more powerful package portable on as many platforms as possible.

I have enjoyed coordinating the design and the implementation of *PAW* until 1993. The development of such systems takes time and requires an intensive cooperation between developers and users. One of the characteristics of interactive systems is that you have nearly an infinity of possible combinations that can be used. The testing phase cannot be done without having a large users base. Vice-versa, having many users imposes a lot of constraints such as backward compatibility. It is also imperative to provide an efficient user support.

Some exotism in the middle of the 80s

In 1985, we saw a first serious campaign to move away from Fortran. The *ADA* language had been adopted by the DoD in the USA and many people thought that HEP could not stay "behind". It was an interesting experience to participate to the ADA club discussions. A technical student Jean-Luc Dekeyser implemented an important subset of *GEANT3* in

ADA, the materials, cross-sections and tracking. This was the subject of his diploma. The language had indeed very interesting features, but was totally unusable in reality, in particular the compilation/Link cycle was far too long.

Some effort was spent in porting our software on IBM emulator machines, 168E, then 370E, that proved to be useful in several online applications. Because the mainframe world was dominated by IBM at that time, we had to adapt our mini interactive systems to the semi-interactive system Wylbur. We also had some attempts to use the native IBM/TSO interactive system for applications such as *HTV* or *GEANT3*. In the middle 1980s, supercomputing was synonym of vector computing. We had a CDC7600 and IBM mainframes with some vector capabilities. We spent a tremendous amount of time in understanding how one could vectorize large programs such as *GEANT3*. Most of these attempts failed. In 1988, RISC machines appeared and were generally considered as the alternative to vector computers. The very positive experience with my Apollo DN1000 pushed OPAL to start the *HOPE project*, the predecessor of *SHIFT*.

Discussions on software methodologies were a hot topic. The *SASD* methodology, in particular, was fashionable. You were considered as a hacker and dumb programmer if you were not able to show some bubble diagrams for your code. Many managers liked these systems considered as a solid guarantee of a good design. I remember a nice talk at a CHEP conference at Oxford in 1990 with the title “Bubbles and no code”. Was this talk fatal to this methodology in HEP? I do not know, but its main advocates stop preaching its virtue soon after.

The nineties

At the end of the 1980s and early 1990s, we spent a considerable effort in the continuous development of the *PAW* and *GEANT3* systems. With *PAW* we had moved from the old *GKS* times to a continuous development of the *HIGZ* system with more and more interfaces (obviously X11) and more operating systems (explosion of the Unix flavours). The success of the row-wise Ntuples called for new developments, the column-wise Ntuples. A Motif-based version called *PAW++* was available in 1992.

GEANT3 also had been in continuous development. The electro-magnetic processes were continuously improved by Michel Maire and Lazlo Urban. We had several interfaces with hadronic packages. The small package Tatina, developed by Tony Baroncelli, was gradually replaced by Geisha, Fluka and Calor. At the same time, the facilities developed for *PAW* were also gradually introduced in GEANT3.

In the early 1990s, we had an interesting parenthesis with the so-called MPP (Massively Parallel Processors). It was a painful exercise to show that parallelism other than event-level parallelism was a non-trivial task and this generated many lovely discussions. A parallel version of *PAW* called *PIAF* was developed by Fons Rademakers, in collaboration with *Hewlett-Packard* . The successful experience with *PIAF* for the analysis of large Ntuples has been essential to design and implement the *PROOF* (Parallel ROOT Facility) a few years later.

On the software side, we had a pleasant workshop in Erice at the end of 1989. The first solid attacks against Fortran were made and the first advocates of Object Oriented Programming appeared in our field. In 1990, Toby Burnett from SLAC had an interesting prototype called *GISMO* of a simulation program written in Objective C and developed on a very advanced workstation (Next). This was also the time when Tim Berners Lee had started the development of the Web. Tim was in my group at that time and I remember all the discussions between him and the majority of us who did not understand what he was doing (described in another article in this Newsletter).

Since the Erice meeting, we were under strong pressure to investigate Object Oriented Programming. Following CHEP in Annecy in September 1992, I became convinced that we had to do something concrete. However, Fortran90 had strong advocates and in many places, it was usual to hear “*I do not know what will be the next programming language in HEP, but I know that it will be called Fortran*”. I had been working a bit with Mike Metcalf, who had written several nice introductory examples to Fortran90. Thanks to his examples, I realized how difficult it will be to implement an I/O subsystem without having runtime type information support in the language. Discussions on languages degenerated quickly between the strong believers in Fortran90 and the strong believers in OO languages. Which OO language was the best was not clear. A project called *MOOSE* had concluded that Eiffel was the best OO language. However, more and more people were looking to C++ as a possible serious candidate.

The years 1993 and 1994 were a turning point in my professional career. After long and hot debates, with my colleagues Fons Rademakers and Masaharu Goto from Japan, I started a new project *ROOT*, that generated even more new pleasant and hot debates. These two lines just to mention that the transition from the procedural Fortran-based programming environment to an Object-Oriented environment has taken far more time that was initially anticipated by everybody, including the true believers.

The new Century

There are now many developers with several years of experience in the OO technology. Different ideas are being tried in different collaborations. Competition in several software areas has been a good thing, at least we can compare different solutions. This is true for the data Base technology, but also, for graphics, User Interfaces, and the methodologies. Frameworks like the *ROOT* system consist of more than 500 classes and more than 500,000 lines of C++ code. Thanks to the fantastic evolution of computers, memories, disk capacity, and tools, a small team of two or three people can still manage these large systems that will probably reach 1 or 2 millions lines of code when the LHC will start in a few years. I do all my development on my laptop with 256 Mbytes of memory and 20 GBytes of disk, yes more than a factor 1000 times more powerful than the machine where I developed Hbook in 1973. Compared to a few years ago when a major release of similar size systems was taken at least one week, we can generate in just a few hours the tar balls for 3 times more systems distributed on many machines on 3 continents.

In all the projects where I have been involved, users have obviously played a key role. Involving users early in a project and getting new users at the right rate corresponding to the state of maturity of a project is the key for success. The naive views of the early eighties with the waterfall development model have always failed. More appropriate models with macro and micro cycles have been more successful. Large systems take years before reaching a sufficient stability (at least 5 years for *PAW* and more than 6 years for *GEANT3* or *ROOT* for example). Stability is a major requirement for a system to be used on a large scale. This explains why the transition to the OO world has taken so much time. *PAW* has been a success once it reached a stability point and also because it was the *lingua franca* for data analysis. However, stability does not mean that a system is frozen. It means that the main concepts are there and already proven with a large users base.

Getting a large users base is a chicken and egg problem. For sure, with today's Internet technology, Open Source concepts, things are easier. However, going from a few users to a few ten, then a few hundred, then a few thousand is a non-trivial task. Past experience is a great help. User support with instantaneous responses to a problem is a must. This is the most difficult thing to do, but also the most rewarding in the long term. Just to give some numbers to illustrate the scale of the problem: the *ROOT* tar files have been downloaded more than 100,000 times via the Web. The Web site has about 500,000 clicks per month and the Users Guide has been downloaded 15000 times in two months. To these numbers, one must add about 4000 messages per year to the public list and nearly 10000 other messages related to the project. Yes, in some sense, this can be seen as a nightmare, but it is also a fantastic opportunity to meet people in all countries of the world. When I start answering mails in the morning, I take advantage of the time difference between countries, starting with mails coming from Japan and Asia, then Europe. At noon, it is time to process the mails from the Us East coast, etc. Our world is really a small village.

I want to thank all my colleagues in the small teams of the *PAW*, *GEANT3* and *ROOT* projects for their essential contribution. They all know the efforts required to support thousands of users, but also the immense satisfaction to work in this very challenging area.

5.4 The PAW History Seen by the CNLs

Olivier Couet, IT/API

Abstract

This article shows how the "Physics Analysis Workstation" (PAW) program evolution has been seen across CNL editions.

From the initial announcement of PAW in 1986, until the Y2K modifications done in its code thirteen years later, PAW has appeared in almost 30 Computer Newsletters. We thought it would be interesting to quickly scan all of the articles and obtain an overview of the "PAW History as Seen by the CNL".

PAW was first mentioned in CNL 186 at the end of 1986. "*A schematic comparison of HTV and its successor PAW*" was presented at this time. PAW was in the testing phase and had not yet really been put to use. Certain of the PAW basic components, HBOOK and HPLOT, were already present in HTV. In this schematic view, present were almost all of the fundamental PAW elements still used today: HBOOK (histogramming), HPLOT (plotting package), HIGZ (basic graphics), ZEBRA (memory management and I/O), KU (user interface). Although, for reasons unexplained in CNL 190 (1987), KU would then be renamed KUIP. A few months after the initial announcement, the first test release of PAW was publicized in CNL 188 in May 1987 and was only for "selected test users". The test period lasted just a few months, and in mid 1987, CNL 189 announced the first (non-test) release of HBOOK 4.00, HPLOT 5.00 and of a new package called HIGZ 1.00 (now at release 1.27/02). Based on these recently released packages, the first official PAW release on Apollo, VAX and IBM VM/CMS was announced in CNL 190 at the end of 1987. This was also the first release of KUIP. Compared to the schematic presented in CNL 186, two packages were added: the Fortran interpreter, COMIS, and the arrays manipulation package, SIGMA. The minimization package MINUIT was not mentioned at this time.

In CNL 190 also, a very simple "`Hello World`" style PAW example was presented. It was suggested to users to create a `PAWLOGON.KUMAC` file with:

```
MESSAGE 'Hello, this is my logon macro executing'.
```

And the article said: "Once your logon macro is executed you get the prompt:

```
PAW >
```

Amazing isn't it ? . . .

An article of 1987 began with the sentences: "*PAW is conceived as an instrument to assist physicists in the analysis and presentation of their data. It provides interactive graphical presentation and statistical or mathematical analysis [. . .]*". This description of PAW has since been used many times in various articles and PAW presentations, and still today is the first piece of text one can find on the PAW home page. The article also explained how to install PAW and its related packages from PATCHY PAM files.

The next summer (CNL 192, 1988), a PAW meeting was announced to “obtain first experience from users and suggestions”. The PAW version was 1.3 (still on the 3 previous platforms). It had an improved version (the first) of the “Ntuple Selection Mechanism and cuts”. In the next CNL (194 January - March 1989), an HBOOK example was given showing how to create an Ntuple in a batch program and then perform an analysis on it utilizing PAW. This Ntuple is now part of the standard PAW examples.

Before mid 1989 (CNL 195), PAW was distributed only as an executable. Since then, the PAWLIB library is available containing PAW callable routines, COMIS and SIGMA. This allows access of these functionalities for any user programs or to build special PAW versions (more memory, experiment specific commands, etc.). The discussion list LPAW is created only at the end of 1990 (CNL 200). Later, the HEPLIB news group would replace this list.

With the user community of PAW growing, it then went into a production phase. At the beginning of 1991, the version number was 1.11. The X11 graphics driver (even on VM) had more and more importance even though a “New Falco driver” was announced in CNL 201. At that time, graphical user interfaces became more and more common. Motif was the main system in this area, and the first version of PAW using Motif appeared in CNL 203 (late 1991). It was not yet Paw++ but that first attempt was a good prototype to understand what could be done and what needed to be accomplished. It was in the same CNL that an improved version of KUIP allowing it to perform `DO` loops, `IF-THEN-ELSE` etc. in the macro scripting language was announced. At the end of 1991, the “World Wide Web” appeared for the first time in CNL 204. The PAW Web site will appear a couple of years later. Several improvements in HIGZ had been accomplished in summer 1992 (CNL 207). The number of graphics drivers was growing including Mac and MSDOS.

CNL 209 in late 1992 announced big modifications in PAW that would later be explained in CNL 210 in the longest article ever published on PAW in the CNL. These news items included: Paw++, the Column Wise Ntuple and Kuip/Motif. The new functionalities were in such an amount as to justify an increment of the main digit of the PAW version number, hence the PAW version number is now 2.00! In that CNL, PIAF’s concepts were also explained. The first PIAF cluster would be in production almost a year later on five HP 755 (CNL 213) together with a modernized version of the Ntuple query processor (based on LEX and YACC).

Between these major changes some more minor events occurred in the “PAW world”. From CNL 211 to 217 (March 1993–July 1994), a test kit was set up, PAW was ported to Linux, the current HEPLIB discussion list started, KUIP rewritten fully in C, and also during that period, CERN reached its 40th anniversary.

Appearance of PAW on the WWW was communicated in CNL 218 at the end of 1994. That first version of the PAW Web site contained items still existing on the site today. In particular, a list of seven FAQs was already available (we have 185 today). These same questions are still “frequently asked” because they are still there in modified and more complete forms. A PAW questionnaire was also launched in that CNL, attempting to better understand how PAW was being used, and the first attempt to manage PAW code with CVS was mentioned in this article.

Even if the second version of the Ntuple Query processor was much improved over the first, it was not powerful enough to handle the power of the Column Wise Ntuples. A new version was requested by users and appeared at the beginning of 1995. With the new Ntuple query processor, the last weak element of the system disappeared and PAW entered a consolidation phase. This was also visible on the PAW Web site where the new sections “PAW usage”, “Tutorial” and “Reference manual” appeared during mid-1995 and the users section “Contributions” at the end of the same year (CNL 220). The “end of PAW” was announced in CNL 223 (June 1996). It was the “last major PAW release” and the people involved in the PAW development and support were directed to concentrate on “*the gathering of user requirements for a future data analysis tool, including requirements for the components, e.g., histogramming, minimization, data input*”, a sad occasion for the PAW developers.

Since CNL 225 (December 1996), PAW announcements can be found only on the WWW. The last reference of PAW in the CNL was in CNL 237 (December 1999) telling users that they should not worry about a Y2K bug affecting PAW, it should go over the New Year (millennium) border safely. . . and it did. So, since 1999, PAW activities are no longer reflected in the CNL. However, all information can be found on the PAW Web site. Upon visiting there, you may find that it was not the “end of PAW” as CNL 223 predicted in 1996.

5.5 EUCLID: the Beginnings of Computer Aided Design

Mik Ferran, N.Høimyr and *I. Seis, CERN/IT-CE*

Abstract

For almost 20 years, the EUCLID 3D CAD-system has been used for designs of CERN’s accelerators and detectors. CERN was a pioneer using a 3D CAD-system when the few CAD-tools in the marketplace were mainly 2D drafting tools. The choice made in 1982 was a good one, the vendor survived until recently and EUCLID has been the workhorse for design of LEP and LHC.

Background

While CERN continued to develop software specific to our needs, the rest of the world was beginning to produce sophisticated general-purpose computer tools. At around the same time (1982) both Oracle and Euclid came to CERN. Each was then fairly new and each was just one of a set of competing products in a new domain.

Selection of a CAD-system for LEP

In the case of Euclid the study group to select a Computer Aided Design (CAD) system for the design of LEP worked in close collaboration with the future users. Claude Hauviller, Hans Horisberger, Detmar Wiskott,. . . had to learn all about the problems of CAD — then in its infancy and with not much more than principles and promises to recommend it. They had to shepherd the thought processes of the study group to the best choice and defend that choice right through to the final decision.

At the time the CAD market consisted of a few packages for Computer Assisted Drafting and Drawing, that were little more than computerised drawing boards, and more advanced systems that allowed for design of 3D surfaces but were very complex and targeted the aerospace and car industries. The EUCLID system was one of a few tools that worked with true 3D solid geometry, knowing the precise volume, shape and position of each element. It thus was able to fit pieces together, detecting interference if an element is moved among others, as for example in a tunnel. Of course an assembly of elements can be projected on to any plane to make up a conventional drawing. Other, graphics-oriented, systems which simply remembered and reproduced the formal collection of lines which constitute a drawing could merely be ancillary and were no match for Euclid.

Another quality of Euclid was that it had a true database that allowed users to share components in a design at a time where other CAD systems worked on individual files. This allowed all designers and engineers to access the same versions of parts in an assembly and keep it coherent, a major advantage for dealing with the very large assemblies made at CERN.

Computing Power and Resources

The computing power needed was underestimated. The power available for 100 kCHF was miserable by today's standards. The VAX had a cycle time of 1 microsecond and was equipped with only 12 Mbytes of memory. The first displays were on Tektronics 14" storage tubes and each terminal cost about 100 kCHF. To make it acceptable to the directorate, Euclid ought to have served electronics applications as well! To get best value out of the large investment, the design office officially extended normal CERN working hours to allow people to work from 7am till 8pm. New money came hard too; memories linger of a day spent running around user divisions to get 4*10 kCHF to buy a 400 Mbytes disk.

Early users were not merely early risers. Indeed, without the dedication and commitment of a small number of designers who put up with extremely difficult conditions, CAD introduction at CERN would have been delayed for years. It is important to recall their working conditions in 1982–1983. Whole days were spent in darkness behind windows covered with black curtains. There was no mouse — the cursor had to be moved with two knobs. There was no documentation and little or no training. Paper copies of drawings produced in building 513 were delivered by van to a local RIOS (Remote Input Output Station). Of course these pioneers had to make reservation in advance for one of the few terminals available. . .

The directorate was extremely sceptical and wanted to put all available money into LEP construction instead of "computer games". However, in 1988 when LEP was well under way, Professor Schopper acknowledged that it had become clear that LEP had very much benefited from CAD, indeed some designs were only possible because of the 3D representations.

When Detmar Wiskott left, Roland Messerli took charge of the computing, interfacing. . . aspects for the maturing years of Euclid, to oversee its use expand at CERN and transfer CERN's knowhow elsewhere in Europe through his involvement with the EUCLID user group.

Individual Contributions

Many key contributions were made in that innovative phase. For instance, Marcel Mottier made the new software his tool and the first convincing results were delivered largely due to his commitment. He wrote the programs ASMAC and LEGO that produced, overnight, complex layout drawings of complete LEP accelerator sections in the tunnel.

Serge Oliger used Euclid and Oracle for the installation of electrical equipment and cable routing in the LEP tunnel. This led to improvements in installation, maintenance, ordering and contract follow-up.

Another outspoken and convincing protagonist was Claude Hauviller, founder and longtime chairman of CAEC, a committee firmly in control of mechanical CAD at CERN. But throughout the acquisition, introduction and deployment it was teamwork and setting up and running the institutions of teamwork, such as CAEC's "Euclid Correspondants" subcommittee, under Roland Messerli, voicing grass-roots opinion, which made it all possible.

We cannot mention everyone who contributed to the success of EUCLID at CERN in a short article, but a major factor behind this success was effective coordination and central support, be it in providing adequate computing resources or disciplined working methods in the drawing offices.

Conclusion

It is a tribute to the care and foresight of those involved that EUCLID, of all the systems proposed at the time, was the only one destined to carry through a product cycle of unbroken development. At CERN the number of experts has grown from about ten in the early days to more than a hundred today. Each user has at his disposal something approaching the power of a computer centre of twenty years ago. One of the few CAD systems still supported after 20 years, Euclid itself is still highly capable and is the central workhorse in the design of LHC and its experiments. The more than 100 000 3D models and drawings in its database attest to the scale of Euclid's contribution to CERN.

But there is no miracle, Euclid is fundamentally a twenty-plus-year-old Fortran program and has fallen behind today's CAD capabilities. Euclid's parent company, *Matra Datavision* was bought by its competitor *Dassault Systèmes* who agreed to keep Euclid alive — for a while. So Euclid and CERN-Euclid knowhow will live on till 2005 and finish the bulk of LHC work. Today's CAD market is being studied by the CAD 2000 project to find Euclid's successor which will gradually supplant the ancestor before the fatal date, end of 2005.

About the author(s): *Based on input from people who were there at the time, the article has been edited by members of the Mechanical CAE section of the Computing for Engineering support group.*

5.6 Old Things and News from SDT

Eric Poinsignon, IT/API

Abstract

History and news from the SDT service.

History and stories...

Once upon a time, in 1992 in ECP Division, PT group (*Programming Techniques*) led by *Paolo Palazzi* in building 1, an "embryo" of the SDT service was created, but nobody knew at that time it was a service.

Arash Khodabandeh started this "service" with two tools: *StP* and *Artefix*. RD13 used *Artefix*, a tool based on *Petri nets* that generates software code. It was the first time generated code was running for a test beam, and it worked! The first purpose of this service was to give access and promote some commercial software engineering tools. They could be tested for their own benefits as the tools were paid centrally by Gottfried Kellner, who was working for ECP division. 50 different tools have been put in production or evaluated over a period of nine years.

As SDT was not defined as a service, there were no servers, no disks and no resources allocated to it. So the tools were installed on the Sun group server: `ptsun00`, on individual disks, which were too small time to time. At each upgrade the tools were moved to new disks but the "service" increased its success, and the first organization became insufficient. Thus the Sun Sparc 20 Server, `pttools`, was installed with the possibility to add disks easily.

An annecdotic story at that time, is a problem we had with the power supply, because the power switch was near the light button and either some house-cleaners or others didn't pay attention, or just wanted to test those funny buttons. This was not so "annecdotic" for our users, and, after some catastrophic recoveries, we finally get our UPS (Unit of Power Supply) to secure the machine.

One other day, a system warning fell in the mailbox saying some files were missing. As the operator was going through the file system, files were disappearing at the same time. The decision was taken to save all the 30 Giga bytes. It was certainly the best solution because after the backup, all the files disappeared. We reconfigured the file system with a mirror and now when a disk crashes it is transparent to the users.

Today...

The service and the server are now located in building 513, in IT/API group. `Pttools` is still distributing the tools and licenses, and the up-time of the server reaches regularly more than 300 days.

You can find the complete information on what we have and how to use it at URL: `http://sdt.cern.ch/` and you can contact us at e-mail address: `Sdt.Support@cern.ch`

Latest news

- *Docsys*: Johan Karlsson replaces Mario Ruggier to take care of `docsys`. The Web site stays at `http://cern.ch/docsys/`, and you can contact him at the usual mail address `docsys.support@cern.ch`. His office is 513,1-005 and phone number 79342.
- *FrameMaker 5.5.6 on Linux* (`http://sdt.cern.ch/FrameMaker/ProcLinux.html`). The script to start FM on your Linux machine but running on a Solaris server is now available. Read the FrameMaker Linux page.
- *FrameMaker 6.0 for Nice 2000* is available (`http://sdt.cern.ch/FrameMaker/ProcWin2K.html`). Each user needs to contact us to be added in the accessible group.
- *Rose 2001 on Solaris and MSWindows* is available (`http://sdt.cern.ch/Rose/`). It will not be installed anymore for IBM/AIX, HP/HPUX and Compaq/Tru64. Rose is NOT available for PC/Linux.
- *Look!*: is no longer used and has been taken out of production.

Catalogue of present tools

Analysis and Design

- *Rose* (`http://sdt.cern.ch/Rose/`)
  OMT/Booch/UML analysis and design tool with C++/Java/Ada code generation.
- *StP* (`http://sdt.cern.ch/Stp/`)
  OMT/Booch/UML analysis and design with C++ code generation.

Programming and Debugging

- *SNiFF+* (`http://sdt.cern.ch/Sniff/`).
  Integrated development environment for C, C++, Java, Fortran and CORBA IDL.

Graphical User Interface Builder

- *X-Designer* (`http://sdt.cern.ch/XDesigner/`).
  Interactive X-platform GUI builder with OSF/Motif toolkit widgets, C, JAVA and Windows code generation.

Quality, Metrics and Testing

- *Logiscope Tool Kit* (`http://sdt.cern.ch/Logiscope/`).
  Source code analyser and test coverage for C/C++/Fortran.
- *Insure++* (`http://sdt.cern.ch/Insure/`).
  Comprehensive tool for error detection in C/C++ programs.
- *CodeWizard* (`http://sdt.cern.ch/CodeWizard/`).
  Coding standards enforcement tool that prevents errors and standardize C/C++ code.

Documentation

- *FrameMaker* (`http://sdt.cern.ch/FrameMaker/`).
  Complete and integrated publishing tool.
- *IXgen* (`http://sdt.cern.ch/IXgen/`).
  Indexing and marker management tools for FrameMaker.
- *SoDA* (`http://sdt.cern.ch/Soda/`).
  Software Documentation Automation.
- *VisualThought* (`http://sdt.cern.ch/VisualThought/`)
  Multipurpose drawing, diagramming and flowcharting tool.
- *SDLT* (`http://www.cern.ch/docsys/framemaker/templates/sdlt/`) (API product).
  Software Documentation Layout Templates, for FrameMaker.
- *TDR* (`http://www.cern.ch/docsys/framemaker/templates/tdr/`) (API product).
  Technical Design Report, for FrameMaker.
- *UGT* (`http://www.cern.ch/docsys/framemaker/templates/GuideTemplates/`) (API product).
  User Guide Templates, for FrameMaker.

About the author(s): *Eric Poinsignon is looking after the SDT Service run by IT/API.*

6. Desktop Publishing

Everything related to "Text Processing": HTML issues, LaTeX, Word, Framemaker, etc.

6.1 A History of Scientific Text Processing at CERN

Michel Goossens, IT/API

Abstract

Text processing has always had some kind of special place in the computing environment at CERN and elsewhere. Processing documentation is an essential part of communication to explain how to use (software and other) tools, describe procedures, and publish results. In fact CERN has more often than not been using "state of the art" standard tools. In this article I review the main scientific text processing systems that have been in use at CERN since the early 1970s. I will show how they evolved logically over time so we end up with the present situation where the main processors can be optimally integrated via XML technologies.

First a word of caution. Text processing systems have evolved considerably over the last decades. First, electronic typewriters acquired memory and developed into a large set of incompatible word-processing systems on dedicated machines (e.g. Norsk Data, Wang, AES, Philips, IBM, Olivetti, and Nixdorf), all of which were used in various CERN services in the 1970s and 1980s. This overview does not discuss these specific systems, but limits itself to (scientific) text processors that were available on CERN's central computing facilities.

The early days: From the typewriter to Waterloo Script

The first issue of the Computer Newsletter (dated 15 February 1966) was produced on a classic typewriter. Later issues were put together by more complex means (no indication of which system was used remains), with "cut and paste" techniques for including figures, tables, etc. playing an important role.

It has to be remembered that commonly-available printers had very limited capabilities in the 1970s, with uppercase-only being the norm. It is thus no surprise that text processing systems available on general-use computers only began to appear when printers got more flexible.

The first text formatting program I found mentioned in the CNL (88, December 1973) was BARB (CERN Program Library entry Q500), which was a program "TO EASE THE CHORE OF OPERATING AND UPDATING PROGRAM WRITE-UPS". It had text strings interspersed with format control and allowed uppercase-only texts with titles, subtitles, appendices, headings, justification (left, right, centered) and **boldface**. Column mode (an elementary way to represent tabular information) was also available.

BARB was superceded a year later by BARBA (BARB ASCII, CERN Program Library entry Q501), which allowed upper and lowercase letters, as well as underlining. The input to these programs was generally in "card form", with column 1 used for control, columns 2–72 for text and control parameters, and columns 73–80 for sequence record information, not read by the processor, but which came in handy when one dropped a box of punched cards (in those days the preferred input method for computer information).

Also in the middle 1970s, Horst von Eicken developed AUTHOR (CERN Program Library Q510), an interactive text processing system for the Control Data 6000/Cyber computer systems. It contained a superset of ASCII, plus the Greek alphabet and some mathematical symbols, and allowed for tabular input and other layout commands. Version 2 was released in 1978 (CNL 136, August 1978).

At about the same time, Tony Shave, wrote SIMTEX (SIMple TEXt processor). SIMTEX was written in BCPL, and hence ran on a fairly wide variety of computers (CDC 6000, IBM 370, Nord 10, PDP-11, VAX, HP 2100). CERN-written documentation of BCPL and of the first generation of CERN microprocessor cross-software, was prepared with SIMTEX.

In the meantime an IBM Mainframe had been installed at CERN, and the PEO (Program Enquiry Office) decided to transfer all its documentation work to IBM (CNL 132, April 1978). To do this they used a program to convert BARBA files to SCRIPT files, that could then be run with the Waterloo SCRIPT processor (e.g., using SYSPUB, a simplified set of SCRIPT macros). This was the beginning of the SCRIPT era at CERN, and that formatter would remain the basis of most text processing work at CERN until the advent of personal workstations in the late 1980s.

The first laser printers

As already mentioned, the quality and functionality of general text processing systems is closely linked to the availability of output devices. It should therefore come as no surprise that the arrival at CERN in April 1979 (CNL 143) of the first laser printer, an IBM 3800, opened up a new realm of possibilities for higher quality typesetting thanks to these laser devices. Anders Berglund, who was to shape text processing at CERN in the 1980s, showed in a further article in September 1979 (CNL 147) how, with the SYSPUB macros, one could obtain accents using a "EURO" character set, as well as miscellaneous other characters for composing block diagrams (the first time that such complex output was possible without having to use a plotter).

More character sets became available at the beginning of 1980 (CNL 149), and they offered for the first time a choice between various type sizes (10, 12 , and 15 characters per inch). Moreover, in the same issue of the CNL Berglund contributed an article "SCRIPT as an Aid in Preparing Papers for Physics Results", where he gave some hints on how to prepare publications for submission to physics journals. He describes PHYSPAP, a macro set developed at CERN and based on Waterloo SCRIPT's SYSPAPER, that allowed high-quality output to be sent to the Photon photo-typesetter— subsequently replaced by a Compugraphics photo-typesetter— connected to a NORD 100 computer, and was customised for styles similar to those of the *Nuclear Physics* journal. A lot of scientific symbols were available, and one-line equations could be typeset, e.g.,

```
E[(X-@m)@2] = &s'-@B.@I&S'@B.(x-@m)@2 f(x) dx = @s@2
```

would produce something like:

$$E[(X-\mu)^2] = \int_{-\infty}^{\infty} (x-\mu)^2 f(x) dx = \sigma^2.$$

One notes the frequent use of the `@` as shorthand, and the `&` as functional operator. Typing high-energy physics processes was quite straightforward, e.g.,

```
@p@+p@AK@+@S@+
```

which gives:

$$\pi^+ \mathrm{p} \to \mathrm{K}^+ \Sigma^+.$$

For multi-level formulae one had to define tabular positions and construct the alignments by hand, e.g.,

```
.stmath
.tb 5 9 12
.tb set $
$@B$1$1
$@S$-$-
$n=1$n@2$m
.emath
```

for the following output:

$$\sum_{n=1}^{\infty} \frac{1}{n^2} \frac{1}{m}.$$

The availability of this system was the basis of a mini-revolution, since for the first time scientists could consider preparing their scientific papers themselves in a reliable way. Because this new application was so much better than BARBA, support for the latter was dropped in August 1980.

The next big event was the arrival of the first loose-sheet laser printer, the IBM 6670. It came with an extended set of Greek and mathematical symbols, including accented characters. This printer proved a huge improvement over what was available with the IBM 3800. The 6670 laser printer also offered for the first time proportionally-spaced fonts (previously only available on photo-typesetters). The latest SCRIPT installation introduced easy-to-remember shorthand for common entities. For instance, when using the proportionally-spaced accented font set 302, one could write:

```
..im SYSPAPER FONT=302
..ch /'e/&eacute.e/
Le monde est carr'e.
```

yielding: "Le monde est carré." The first line in the SCRIPT source loads the SYSPAPER macro package with the font 302, while the second line instructs SCRIPT to change all occurrences of `'e` to `&eacute.e`, where one recognises the precursor notation for entity references in GML, and later SGML and XML.

CERNPAPER

Over the years, users of SCRIPT at CERN had been using various macro packages, such as SYSPAPER, SYSPUB, PHYSPAP, or had developed their own SCRIPT-based macro definitions library (e.g., Gino De Bilio for the manuals of DD-EE Group, Julius Zoll for his Hydra & Patchy documentation, Horst von Eicken for writing yellow reports documenting microprocessor cross-software). Therefore, it seemed appropriate to propose and develop a generic SCRIPT document processing macro-system optimised for CERN use, and Eric van Herwijnen, working with Anders Berglund, released CERNPAPER in January 1984 (CNL 172). It presented a set of high-level macros for various layouts, such as letters, memoranda, technical notes, reports, minutes and agenda for meetings, papers for physics journals, writeups, and manuals. A user-friendly interface to Wylbur, an interactive line-oriented interface to IBM's MVS, was available to easily produce a skeleton job for the various supported document types, thus eliminating the need to be a SCRIPT expert. In fact, most basic SCRIPT commands were made "invisible" and were replaced with generic commands, such as

```
.chapter
.point begin
.point xxxxx
.point yyyyy
.point end
.para
```

The above lines start a chapter, define a numbered list, and start a paragraph.

New versions of Waterloo SCRIPT were made available as soon as they were released. They included support for the newest photo-composition and laser printing devices, that allowed for better font handling, negative skips and overlaps. They also introduced spelling checking possibilities via the inclusion of dictionaries, improved hyphenation, more flexible super and subscript handling and better error-reporting.

At the beginning of 1985 an APA6670 (All Points Addressable) high-volume single-sheet printer was installed at the Computing Center. This printer was announced in CNL 178, and henceforth the Computer Newsletter was typeset on that printer, a quantum leap forward in quality from the monospaced fonts used for the earlier issues.

From GML to SGML

At the beginning of 1985 SCRIPT version 84.1 was installed. It introduced support for GML (Generalized Markup Language). But rather than use the native GML syntax, Berglund took the wise step of introducing directly the reference concrete syntax of SGML (Standard Generalized Markup Language) that was at that time in the last stages of becoming an ISO standard 8879[10]).

SGML considers documents as tree-structures, where a grammar can be defined for any given class of documents. SGML is therefore not a "markup language" in the same sense as SCRIPT or TeX. SGML only defines the syntax for creating an infinite variety of markup languages and is hence completely *independent* of the text formatter. Documents marked up with SGML can be interchanged between different systems and rendered on different installations.

Berglund published the first edition of the CERN SGML User's Guide in October 1986. It offered a rich tag set for preparing the same kind of documents as those proposed by CERNPAPER, as well as for foils. Here is an example of the markup:

```
<!DOCTYPE sgmlmins>
<GDOC SEC="Secondary title">
<TITLEP>
<TITLE>Minutes of todays meeting
<DATE>Whenever
<DISTRIB>
<DIST>Present were
Me
Some Others
</DISTRIB>
</TITLEP>
<BODY>
<P>Nous avons discut&eacute; :
<OL>
<LI>bla bla
<LI>More bla bla
</OL>
<CLOSE>
<TYPIST>MG/xyz
<RECORDER>MG
</CLOSE>
</GDOC>
```

We see the use of SGML's concrete syntax, i.e., < and > for starting and ending element tags (GML uses `:` and `.` respectively). We have an attribute (`SEC="..."`) on line 2, while special characters (`é` yielding the accented "é" on line 13) are entered using *entity references*. Note that I typed all tag names in uppercase, but this is not necessary in SGML, since in most systems element and attribute names are case-insensitive. However, entity names are case-sensitive. On the other hand, in XML (see later) element and attribute names, as well as entities, are *always* case-sensitive.

The introduction of LaTeX

In the late 1980s various Unixes and VAX/VMS became popular at CERN (and elsewhere) and the need for a text processing system that ran on all systems became ever more important.

Physicists and engineers who visited the United States of America, especially SLAC, told us with great enthusiasm about TeX, a publicly available text processing system that D.E. Knuth of Stanford University had been working on with his students since 1977. It consists of two main components, TeX[6] and METAFONT[8]. About the aim of his project, Knuth wrote in the foreword of the TeX book[6]: "TeX [is] a new typesetting system intended for the creation of beautiful books — and especially for books that contain a lot of mathematics. By preparing a manuscript in TeX format, you will be telling a computer exactly how the manuscript is to be transformed into pages whose typographic quality is comparable to that of the world's finest printers".

TeX popularity with thousands of scientists is mainly due to the ease with which any kind of writing can be transformed into various document classes, such as articles, reports, proposals, books, in a way that is completely under the control of the writer through a rich set of formatting commands.

By its very conception, TeX is particularly useful when the document contains mathematical formulae that have to be rendered with high typographic precision. Moreover the program, originally written in an enriched dialect of Pascal, but now distributed in C, can be compiled on almost any operating system in the world, so that it runs on a wide range of computer platforms, from micros to mainframes. It behaves 100% identically on all machines, a fact extremely important in the scientific and technical communities. Related to this portability is TeX's printing device independence, so that a document can be printed on anything from a CRT screen, a medium-resolution dot or laser printer, to a professional high-resolution photo-typesetter.

Because of these qualities and since it is available in the public domain TeX has become the *de facto* standard text processing system in many academic departments and research laboratories, It has also been adopted by members of the professional publishing world as a printing engine. In his foreword to "TeX and METAFONT, New Directions in Typesetting"[9], Gordon Bell wrote that "Don Knuth's Tau Epsilon Chi (TeX) is potentially the most significant invention in typesetting in this century. It introduces a standard language in computer typography and in terms of importance could rank near the introduction of the Gutenberg press".

At the beginning of the 1980s, Leslie Lamport started work on a document preparation system LaTeX[7], based on the TeX formatter. The system adds a level of abstraction to the plain TeX commands and lets the user concentrate on the structure of the document rather than on formatting details. A few high-level commands allow the user to easily compose most of his documents. Users do not have to bother about typographic details, which are left to the document designer, who provides style files for every application.

TeX was first officially introduced at CERN in September 1987 (CNL 189) running on the central VAX Service. System support was provided by Alexander Samarin who developed a set of integration tools. However, on the VM/CMS system, the "recommended" system remained SGML. A TeX service on VM/CMS was announced in May 1988 (CNL 191).

Support was by Jurgen de Jonghe of the TP Section of the newly created OCS Group in MIS Division. In September 1989 SGMLTEX, an application for supporting SGML on VAX/VMS, was announced (CNL 196).

Defining a text processing policy

The need to define a global text processing policy for CERN became ever clearer and in 1989 a proposal (CERN/DD/89-25) was presented to the Meddle Committee and approved. The proposed policy was presented during a CERN-wide seminar about "Text processing at CERN" in December 1989.

A summary, detailing which text processing systems were supported at that time, was published in CNL 198:

- LaTeX on all platforms (IBM, VAX, PC, Mac, Unix workstations);
- SGML/BookMaster on IBM (based on IBM's DCF SCRIPT interpreter);
- Microsoft Word on PC and Mac;
- Interleaf (later FrameMaker `http://www.adobe.com/products/framemaker`) on workstations (later also on PC and Mac);
- Waterloo based SGML, SCRIPT, and CERNPAPER were to be phased out.

The move from SGML based on Waterloo SCRIPT to IBM's BookMaster suite was because the latter presented a more professional and standard-compliant product, including a very flexible high-level style language for defining document layout. CERN customisation of the IBM styles was available via the BOOKIE exec. The math formatter was SMFF, a variant of EQN, the formula processor originally developed for `troff` on Unix systems.

As of issue 198, the CNL was produced with SGML/BookMaster and started featuring a regular section on text processing. For instance CNL 199 (June 1990) contained a long article about SGML by Eric van Herwijnen (who, after Anders Berglund had left for ISO, had become responsible for text processing developments at CERN), and an introduction to the PostScript language (by M.G.). CNL 202 (June 1991) contained an overview article describing the various text processing systems in use at CERN, recommendations for typesetting rules for writing scientific documents (still very relevant today, see `http://home.cern.ch/goossens/typorules/typorules.html`), and an explanation of the importance of SGML in the publishing world.

CNL 203 contained further information on how to optimally prepare one's documents for typesetting, and proposed a set of entity names for elementary particle. At the back it included a questionnaire about the text processing needs of the user community (use of systems and macro packages, needs for multiple input languages, training, required included graphics material).

During this period the documentation for the software packages supported by the User Support (later Application Software) Group (e.g., Hbook, PAW, CERNLIB, GEANT) was translated from Waterloo SCRIPT/SGML, into LaTeX, and made available as printable PostScript files (CNL 205).

The creation of the Web and HTML

With mainframes being abandoned and most development, administrative, and production work moving to Unix and Microsoft Windows workstations, there was for a few years a three-prong approach for producing documents at CERN, namely TeX for physics documents (all systems, with first-line support by Michèle Jouhet's Team in Michael Draper's ETT/DH Group), FrameMaker for large technical manuals and reports (all systems, with support until the end of 2000 by Mario Ruggier, and presently by Johan Karlsson in IT/API), and Microsoft Word (or, more generally, Microsoft's Office Suite) for administrative work (and some technical work in the engineering sector) on PC and Mac (with support also in ETT/DH).

With the help of a consultant, Sebastian Rahtz, in 1992/93 we installed at CERN a reference system containing all the latest LaTeX developments. This work became the basis of the TeXLive CDROM, which is now a world reference for TeX distributions, as well as of three books [3, 4, 5] on the use of LaTeX (articles about LaTeX developments appeared in almost every CNL from 203 to 225) and from CNL 206 (until today!) the printed version of the CNL has been produced with LaTeX (with one exception, see below).

In the early 1990s a major event happened at CERN, namely Tim Berners-Lee and collaborators developed the basics of what was to become the Web. In those days (1992–1993) Tim B.-L. was sitting just a few offices down the corridor from where we (M.G. and S.R.) were working, and already at the beginning of 1993 we had translated,with the active help of Tim, some LaTeX documents into HTML (we started with the HBOOK manual), first using a home-made *ad hoc* set of LaTeX macros, later with LaTeX2HTML. A first article on HTML appeared in September 1993 (with a nice picture of an HTML page displayed with Xmosaic, the first generally-available X-Window browser and the precursor of Netscape), clearly showing that at CERN HTML was being used well before the rest of the world, that became aware of the Web mostly after the "Woodstock of the Web", the First World Wide Web Conference organised at CERN on May 25-27 1994 (a more detailed history of the Web can be found in [1, 2]).

The Web is essentially based on the successful triad: the HTTP protocol, the URL uniform addressing scheme, and the HTML language. The syntax of the HTML language was inspired by that of the SGML language system that we had been running at CERN since the middle 1980s. However, the first versions of HTML were presentation-directed, and it was not until version 3 of HTML that a formally correct DTD of the language was published.

By a decision of the CERN Council in December 1994 CERN officially left Web development to the World Wide Web Consortium (W3C, `http://www.w3c.org`), that had been set up a few months earlier. INRIA (France) and MIT (USA), later joint by Keio in Japan, were to co-host the W3C, and coordinate further Web-related activities.

As far as the CNL is concerned there was a move from LaTeX-source to HTML-source in the middle of 1996 (CNL 223), and one issue (CNL 224) was produced with the "Print" menu option of the browser. However, several readers protested about the "significant loss of markup, structure and readability" (a fact that the CNL editors agreed with), and hence it was decided to transform the HTML back to LaTeX to prepare the printed version with a higher typographic quality.

Coming full circle

The Web became so popular that several browser vendors started competing to offer specific extensions to the HTML language to attract users wanting to publicise their products. Therefore, various mutually-incompatible dialects of HTML appeared. Moreover, to really benefit from the Web and the various applications that were being developed, the XML initiative was initiated in late 1997. Jon Bosak, who published his seminal article "XML, Java, and the future of the Web" (`http://www.xml.com/pub/a/w3j/s3.bosak.html`) around that time, was one its main promoters. All this culminated in the publication of the XML W3C Recommendation (First edition `http://www.w3.org/TR/1998/REC-xml-19980210`, Second edition `http://www.w3.org/TR/2000/WD-xml-2e-20000814`), which defines the XML language in a formal way.

XML is truly and explicitly international in that it espouses Unicode as its basic character set. Unicode (`http://www.unicode.org`) provides an unambiguous, fixed 16-bit (2-octet) code set (in fact 17 planes, each 16 bits wide, are defined). The 65,536 characters of plane 0 contain ASCII, plus most characters needed for writing in the major living languages. The other sixteen planes with 65,536 characters each, provide for the inclusion of ancient languages, more specialised math and other characters, plus over 100,000 code points for private use areas.

Recently, many XML based applications have been developed, and XML seems to have taken over the Internet World. XML is a light-weight version of SGML, that can be parsed by a relatively simple program (various publicly-available XML parsers exist in almost every computer or scripting language). XML-based standards exist for document navigation and manipulation (XPath, `http://www.w3.org/TR/xpath`, XSLT, `http://www.w3.org/Style/XSL/`, XML Query, `http://www.w3.org/XML/Query`), Schema definition (XML Schema `http://www.w3.org/XML/Schema`), Formatting and presentation (XSL-FO, `http://www.w3.org/TR/xsl/` and CSS, Cascading Style Sheets, `http://www.w3.org/Style/CSS/`), Scalable Vector Graphics (SVG, `http://www.w3.org/Graphics/SVG`), mathematics (MathML, `http://www.w3.org/Math/`, Generalized links (XLink/XPointer, `http://www.w3.org/XML/Linking`), and an XML version of HTML (`http://www.w3.org/MarkUp/`) is being worked on.

Looking back over a period of almost thirty years one observes that CERN started with specific markup (BARB and SCRIPT-based languages), and went via a more generic approach (CERNPAPER and SGML) to a portable mixture of presentation and structure (LaTeX). HTML started off as essentially presentational (for making nice-looking and attractive Web pages), but soon it was realized that presentation (style) and structure should be independent. Hence there was a return to the SGML approach, pioneered in the early 1980s, in the form of XML, the SGML-lite of the Web. Table 6.1 shows a comparison of the syntax of various text processors used at CERN since the 1980s until now.

XML and LaTeX, perfect twins for the 21st century?

Most Web-related activities are currently centered on the use of XML for data exchange and document production (Elliotte Rusty Harold's "Cafe con Leche XML News and Resources" at the URL `http://www.ibiblio.org/xml/` is a good source of up-to-date information). Microsoft, Oracle, IBM, Sun, as well as open source applications, such as perl, Python, the GNU Project, GNOME, have all included XML into their basic strategic plans. Hundreds of dedicated data vocabularies (DTDs), each dealing with a particular subject area, have been proposed (see Robin Cover's "XML Pages" at the URL `http://www.oasis-open.org/cover/xml.html`).

For scientific and computer documentation DocBook (`http://www.docbook.org`) markup has been in use for many years. The DocBook DTD or schema contains hundreds of elements to mark up clearly and explicitly the different components of an electronic document (book, article, reference guide, etc.), not only displaying its hierarchical structure but also indicating the semantic meaning of the various elements. Moreover, the structure of the DTD is optimised to allow for customisation, thus making it relatively straightforward to add or eliminate certain elements or attributes, to change the content model for certain structural groups, or to restrict the value that given attributes can take.

A short example of a document marked up in DocBook and including some math follows:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE article SYSTEM "mybook.dtd"[
<!ENTITY DB "<application>DocBook</application>">
]>
<article>
<articleinfo>
<title>Docbook with a few formulae</title>
<author><forename>Michel</forename>
        <surname>Goossens</surname>
</author>
<pubdate>Wednesday, 18 March 2001</pubdate>
<abstract>
<para>
This XML document is marked up according to
the &DB; schema. It shows a few elements of
the &DB; vocabulary, as well as a couple of
examples of mathematical expressions where
we used MathML markup.
</para>
```

```
</abstract>
</articleinfo>
<section>
<title>A MathML example</title>
<para>
A MathML formula can be typeset inline, as here
<inlineequation>
<math>
 <mrow>
  <msup><mi>&#x03C0;</mi><mo>+</mo></msup>
  <mi>p</mi>
  <mo>&#x2192;</mo>
  <msup><mi>K</mi><mo>+</mo></msup>
  <msup ><mi>&#x03A3;</mi><mo>+</mo></msup>
 </mrow>
 </math>
</inlineequation>, a simple particle reaction.
</para>

<para>
A mathematical equation can also be typeset in
display mode using &DB;'s
<sgmltag class="element">informalequation</sgmltag>
element, as is shown in the following example
containing a summation expression:
</para>

<informalequation>
<math>
 <mrow>
  <msubsup><mo>&#x2211;</mo>
           <mrow><mi>n</mi><mo>=</mo><mn>1</mn></mrow>
           <mrow><mi>&#x221E;</mi></mrow>
  </msubsup>
  <mfrac><mn>1</mn><msup><mi>n</mi><mn>2</mn></msup></mfrac>
  <mfrac><mn>1</mn><mi>m</mi></mfrac>
 </mrow>
 <mtext>.</mtext>
</math>
</informalequation>
</section>
</article>
```

The DTD `mybook.dtd` (see below) is loaded on line 2, and in the internal subset (line 3) an entity `DB` is defined as a shorthand for "DocBook" (marked up as an `application` element). Then the article informationis set up, specifying title, author, abstract, etc. The body of the text shows two examples of markup of formulae in MathML. The first corresponds to the physics process and the second to the summation equation shown as examples earlier in this article. Note the Unicode character entity references for denoting non-ASCII characters, in particular `π` (π), `Σ` (Σ), and `→` ($\rightarrow$) in the first formula, and `∑` ($\sum$), and `∞` (∞) in the second formula.

The DTD `mybook.dtd`, which follows, declares how to combine MathML's `<math>` "master" element with the DocBook DTD (lines 1 and 2), while the following lines declare parameter entities for the MathML and DocBook DTD, and load them (last two lines), so that their elements are available for the XML parsers. The attribute definition for the `math` element on line 5 defines the namespace for that element and its daughters.

```
<!ENTITY % equation.content "(math+)">
<!ENTITY % inlineequation.content "(math+)">
<!ENTITY % mathml SYSTEM
   "/opt/XML/cdrom/dtd/mathml/mathml2.dtd">
<!ATTLIST math xmlns CDATA #FIXED
   "http://www.w3.org/1998/Math/MathML">
<!ENTITY % docbook SYSTEM
   "/opt/XML/cdrom/www.nwalsh.com/docbook/xml/docbookx.dtd">
<!-- load MathML and docbook -->
%mathml;
%docbook;
```

The XML document described above was first transformed with an XSLT processor into XSL-FO formatting elements with the help of Normal Walsh's XSLT stylesheets (`http://nwalsh.com/docbook/xsl/`). These formatting objects were then interpreted and typeset with Sebastian Rahtz' `PassiveTeX` (`http://users.ox.ac.uk/~rahtz/passivetex/`), and the result is shown in Figure 6.1.

Various other ways exist to transform XML documents into viewable information with the help of XSLT stylesheets (via HTML or directly with CSS in XML-aware browsers). XSLT to XSL-FO transformations and specific applications (such as the Apache Project's FOP, `http://xml.apache.org/fop/`, or PassiveTeX, see above) can be used to obtain PDF or PostScript.

More generally, as already explained in CNL 2000-002 (`http://ref.cern.ch/CERN/CNL/2000/002/xml-strategy`), XML can be considered as the central element in an integrated Internet strategy, where documents (and data) are stored in an "online" electronic repository in XML, from which they can be transformed into various formats. We reproduce here the figure of that article (Figure 6.2), where we emphasise the complementary roles of XML (at the top right), that can be manipulated by a whole set of mostly publicly-available standard tools, and LaTeX (at the top left), that is available for high-quality typesetting purposes, where precision and clarity is desired.

As browsers are becoming more powerful in the way they can display mathematics (with MathML) and simple two-dimensional graphics (SVG), by using plug-ins or native code, it is expected that in the not too distant future (end 2001?) one will be able to cut and paste a formula or data in a graph from one application (e.g., a browser) into another (e.g., a Java or C++ program) using XML technology.

CERN is presently collaborating with other academic and commercial partners in an IST project funded by European Union, TIPS (Tools for Innovative Publishing in Science). The aim of TIPS (`http://tips.sissa.it`) is to develop a set of user-friendly and advanced tools and services that are organised in an open system to support research information production, management, access, and use in a coherent manner. These ideas are being tested using articles of the electronic *Journal of High Energy Physics* (`http://jhep.sissa.it`). In particular, the articles are transformed from LaTeX into XML to see how this generic format can be used to optimally support the activities of document writing, reviewing, publishing, searching, disseminating and reading, as well as communication among members of the research community.

Table 6.1: A comparison of the syntax of various text processors

Description	CERNPAPER	CERN SGML (HTML)	LaTeX	nroff/troff (mm macros)	XML/DocBook (MathML)
Document sectioning commands					
Level 0	.part	<H0>	\part	not available	<part>
Level 1	.chapter	<H1>	\chapter	.NH1/.H1	<chapter>
Level 2	.section	<H2>	\section	.NH2/.H2	<sect1>
Level 3	.subsection	<H3>	\subsection	.NH3/.H3	<sect2>
Level 4	.subsub	<H4>	\subsubsection	.NH4/.H4	<sect3>
New paragraph	.para	<P>	\par	.PP/.P n	<para>
Highlighted and other special text					
Normal	normal text	normal text	normal text or \textrm{*text*}	normal text or .R	normal text
Emphasis	.highl1 ... .ehighl1	<HP1> ... </HP1> (<EM> ... </EM>)	\emph{...}	.I ...	<emphasis>...</emphasis>
Quotation	.quote begin ... .quote end	<Q> ... </Q>	\begin{quote} ... \end{quote}	.QS ... .QE	<quote> ... </quote>
Footnote	.footnote ... .footend	<FN> ... </FN>	\footnote{...}	.FS ... .FE	<footnote> ... </footnote>
Lists					
Ordered	.point begin .point .point end	<OL> <LI> xxx <EOL>	\begin{enumerate} \item ... \end{enumerate}	.AL .LI xxx .LE	<orderedlist> <listitem> ... </orderedlist>
Unordered	.bullet .bullet ... .bullet end	<UL> <LI> xxx <EUL>	\begin{itemize} \item ... \end{itemize}	.BL .LI xxx .LE	<itemizedlist> <listitem> ... </itemizedlist>
Description (glossary)	.glossary begin .glosssary xxx ... .glossary end	<DL> <DT> xxx <DD> ... <EDL>	\begin{description} \item[xxx] ... \end{description}	.VL .LI xxx ... .LE	<variablelist> <varlistentry> <term>...</term> <listitem>...</listitem> </varlistentry> </variablelist>
Principle of how to type mathematics and other special characters					
Greek (e.g. α)	@a	α	α	\(*a	α (α)
Math symbol (e.g. $\approx$)	&approx.	≈	$\approx$	\(ap	≈ (≈)
Inline math	*special math chars*	*special math chars*	$*math text*$ or \(*math text*\)	# *math text* #	<inlineequation> $*MathML code*$ </inlineequation>
Display math	.stmath ... .emath	*special math chars*	\[*math text*\]	.EQ *math text* .EN	<informalequation> $*MathML code*$ </informalequation>
Superscript	&S'*text*.	^{*text*}	$\sp{*math text*}$	# sup {*math text*}#	<subscript>*text*</subscript>
Subscript	&s'*text*.	_{*text*}	$\sb{*math text*}$	# sub {*math text*}#	<supscript>*text*</supscript>
Accents and umlauts	&acute.e or &trema.u	é or ü	\'e or \"u	e*' or u*:	é (é) or ü (ü)
Non breaking space	\ (set by command ".blank \")		~	\ or \(space)	
Specific formatting page markup					
Line break	.break	Fall through to formatter	\newline	.br	Processing instruction for formatter
Page break	.page	Fall through to formatter	\newpage	.bp	Processing instruction for formatter

> **Docbook with a few formulae**
>
> Michel Goossens
> Wednesday, 18 March 2001
>
> This XML document is marked up according to the DocBook schema. It shows a few elements of the DocBook vocabulary, as well as a couple of examples of mathematical expressions where we used MathML markup.
>
> **A MathML example**
>
> A MathML formula can be typeset inline, as here $\pi^+ p \longrightarrow K^+ \Sigma^+$, a simple particle reaction.
>
> A mathematical equation can also be typeset in display mode using DocBook's `informalequation` element, as is shown in the following example containing a summation expression:
> $\sum_{n=1}^{\infty} \frac{1}{n^2} \frac{1}{m}$.

Figure 6.1: A simple DocBook document typeset with LaTeX

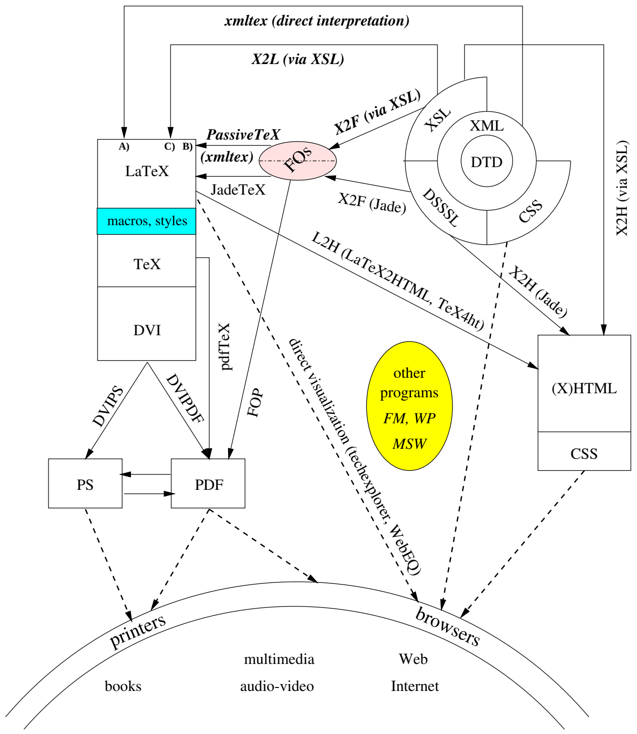

Figure 6.2: An XML strategy for the Web

Bibliography

[1] Tim Berners-Lee. *Weaving the Web*. Orion Business Books, London, 1999.

[2] James Gillies and Robert Cailliau. *How the Web was Born*. Oxford University Press, Oxford, UK, 2000.

[3] Michel Goossens, Frank Mittelbach, and Alexander Samarin. *The LaTeX Companion*. Addison-Wesley, Reading, 1994.

[4] Michel Goossens and Sebastian Rahtz, with Frank Mittelbach. *The LaTeX Graphics Companion*. Addison-Wesley, Reading, 1997.

[5] Michel Goossens and Sebastian Rahtz. *The LaTeX Web Companion*. Addison-Wesley, Reading, 1999.

[6] D.E. Knuth. *The TeX book*. Addison-Wesley, Reading, 1990.

[7] Leslie Lamport. *LaTeX, A Document Preparation System*. Addison-Wesley, Reading, 1986.

[8] D.E. Knuth. *The Metafontbook*. Addison-Wesley, Reading, 1990.

[9] D.E. Knuth. *TeX and Metafont, New Directions in Typesetting*. The American Mathematical Society and Digital Press, Stanford, 1979.

[10] International Organization for Standardization. *SGML, The Standard Generalised Markup Language*. ISO 8879, ISO Geneva, 1986.

Acknowledgements

I would like to thank Anders Berglund, Julian Blake, Ian McLaren, Miguel Marquina, Jutta Megies, Harry Renshall, David Stungo, David Williams, and Roger Woolnough for sharing information or documents with me while I was preparing this article.

About the author(s): *Michel Goossens is a CERN authority on LaTeX, XML and Electronic Document Publishing techniques in general. He has written several articles and books on the subject.*

7. The Learning Zone

All kinds of "educational" computing issues, not necessarily directly CERN-related, such as computing news from the Press and tutorials on various subjects.

7.1 Computing at CERN, Once Upon a Time...

Miguel Marquina (Editor), IT/User Support

Abstract

A retrospective to events in the CERN Computing arena.

Five years ago...

End of DXCERN Ultrix Service.
(CNL 222, "*Migration Special - Mail Issues*", January–March 1996)

After many years of faithful service the `dxcern` Ultrix service will be stopped towards the end of 1996... DIGITAL is encouraging users to migrate to Digital Unix...

This year was of particular relevance to CN services. It meant the bridging to new technologies and the stoppage of old hardware and operating systems. It was also two years away from the DEC-Compaq merger, which symbolically expressed the merger trend of the late nineties (Ed.)

Roadmap to MAIL Services.
(CNL 222, "*Migration Special - Mail Issues*", January–March 1996)

The main criteria you should use when choosing your preferred MAIL service are: the combination Desktop/OS with which you feel most comfortable; and the need to process your mail from more than one location...

This is not the only relevant article on mail, it was rather part of a dedicated CNL issue. I would tag 1996 as the year of our "User Desktop Revolution". Hundreds of users who had come to terms over the years with the console screen and rather unexciting interfaces, started encountering the richness of a "multiple-window environment. Not always as a more pleasant experience! (Ed.)

Ten years ago...

World-Wide Web: On-line information for everyone.
(CNL 204, October-December 1991)

A world of information is now available online from any computer platform. Information sources at CERN and across the world span subjects from poetry to biochemistry and supercomputing. We summarise the information currently sourced at CERN, and we introduce the `WorldWideWeb` (W3) program which allows you to browse and search all the data in a simple and consistent manner...

Ten years back few of us were colleagues of Tim Berners-Lee. I was even in the next office and, for a while, his (nominal) supervisor. He had a very clear vision of the worth of `http`, and history has taken care of putting it at the right place.

This CNL article describes in some detail his line-mode interface, as such the proof of concept (enhanced by a Next-special variant). Few of us thought that an X/Motif interface would make a real difference in user experience for the UNIX-based physicists.

Unfortunately not enough effort could be invested in this aspect. Then NCSA/Mosaic came out; the rest is history... (Ed.)

Can we and should we save the CNL?
(CNL 205, January-February 1992)

The objective of the CNL has always been to inform users about the *computing infrastructure* at CERN. The content and form... have been static. There was almost no feedback from the readers. We need feedback.

It was true ten years ago, it is true still today. No initiative meant to be useful to customers can survive without customer feedback. Luckily (for us) Bernd Pollermann did get enough feedback to continue, such that we can now celebrate the CNL anniversary (Ed.)

Twenty years ago...

6250 BPI tapes on CDC.
(CNL 156, January-February 1981)

Now that NOS/BE is running without problems... 6250 bpi tapes can be used freely on the CDC machines. Since not all units are capable of processing tapes at 6250 (or 800) bpi, the parameter... *Use PARTIAL STAGING whenever possible.*

6250 bpi tapes were a novelty in the early eighties, storing three times as much as a "normal" tape! About 150 MB worth of information. Part of the spectacle at any Computer Centre at this time was seeing the set of tape units all chewing data in jerky movements.

Nowadays that much can be stored in one "SuperDisk" diskette or a fraction of DVD, and it can be read at our home computer. We will have to find other excuses to be impressed (Ed.)

Adding Black stripes to a car does not make it go faster.
(CNL 156, January-February 1981)

A letter from one of our readers giving his opinion of FORTRAN 8X.

This is a criticism of the draft standard of FORTRAN 8X, summarised in the last CNL. Besides some technical questions, one particular point arises, namely, is it really worthwhile to "pump up" a well established language well beyond its limits?

FORTRAN 77 survived in the eighties and the nineties at least two attempts at "revamping". Meaning, of course, as a de-facto standard in High Energy Physics software. The last question remains: that is probably one good reason why Java was born rather than inventing C+++ (there are many others; however, I don't intend to launch a polemic now) (Ed.)

Thirty years ago and beyond...

A Large European Computer installed at CERN.
(CNL 60, 2nd April 1971.)

The main computer for the Omega and Split Field Magnet (SFM) Projects, a CII 10070 manufactured in France by the Compagnie Internationale pour l'Informatique, was delivered to the Data Handling Division and installed in Lab.5 during the month of December 1970.

...The software includes a multiprogramming operating system produced by CII (SIRIS 7).

I guess that "multiprogramming" here refers to multitasking; I need a real computing historian here. Twenty years before anybody would offer a desktop alternative to UNIX.

But it is well known that one prefers to run onto one's own mistakes than to learn from the experience of the others; eh Bill? (Ed.)

Programming Enquiry Office.
(CNL 60, 2nd April 1971.)

The Programming Enquiry Office is currently only staffed with two members due to transfer and leaving of staff. In May we expect one new member to arrive while two further posts are still vacant. Be patient, we will try to operate the office full time, but it will be difficult to offer the same service as before.

Wow! Not so much of a change in thirty years!

When I read these statements one profession comes to my mind: public teachers, who are considered an essential foundation of this society but whose vocation is considered near to priesthood (and therefore social recognition goes accordingly).

We have a saying in Spanish which says "Itch taken with pleasure does not bother. Yes, it must be that :-)) (Ed.)

About the author(s): *Miguel is starting to be amused at poking at historical stuff, be aware!*

7.2 Questions and Answers from the Computing Helpdesk

Collected by the IT/User Support Team

Nicole Crémel, (Editor), IT/User Support

Abstract

A collection of Questions & Answers that have been treated by the Computing Helpdesk managed by User Support group.

N.B. The number in parentheses refers to its relative numbering in the User Support "Question-Answer" database, at URL: `http://cern.ch/consult/qa/NNN` *where* `NNN` *is the problem identifier (number).*

Some History

The "*Questions and Answers from the Computing Helpdesk*" is a regular section in the CNL that first appeared 6 years ago, in CNL 219(January – March 1995). At that time the title was still "*Questions and Answers from the UCO*". It was renamed in CNL 233 (October – December 1998), when the "Computing Helpdesk" was officially announced as a one-desk, one-address, one-number for users, with its new and simple telephone number (78888) and e-mail address (`Helpdesk@cern.ch`), replacing the defunct (but well-known and sometimes regretted by users) "*User Consultancy Office*" (see related article in CNL 233).

The idea to introduce such a section in the CERN CNL originally came from browsing the DESY NewsLetter (their CNL equivalent). Also, with the increasing number and diversity of user queries coming to the Helpdesk, it seemed a good idea to print regularly some of the "frequently asked" questions, in the hope of anticipating some small (and known) problems that users may experience (and for which we can already provide a solution).

PC Support

(Hannes Schwarzbauer, IT/User Support)

Question [367] – Inaccessible servers with W2000

Ever since the migration of some home servers to W2000 I encounter a strange phenomenon on my NT computer:

I usually use the PC `phonebook` *to map someone else's home directory, in order to access his public folder. However, sometimes one W2000 home server is unreachable (i.e. asks me for a password) whereas I can perfectly display others.*

Even stranger, some of my colleagues succeed in accessing home servers, some don't.

Answer

Since the migration certain access mechanisms have changed.

If one tries to access someone's home directory via the phonebook and then maps it, internally the account GUEST is used. This account is not valid anymore for W2000, so the user can't access this server and *has to logout/re-login*.

Therefore the *only valid method* to access someone else's directory is to go from the `Start` menu into `Run` and type the account there, e.g.: `\\srv1_home\joepublic`

(Note that this problem will disappear with the introduction of NICE2000).

Question [374] – Access 2000 to Access 97 conversion

I was sent an Access 2000 database which I cannot read under the NICE Access 97 version. Is there a possibility to convert?

Answer

Because of the change in the file format, you cannot open an Access 2000 database in any previous version of Access. You must convert it to Access 97 like this:

```
Tools - database utilities - convert database -
to prior access database version.
```

Type a new name for the database and save it.

Please note:

- you cannot convert a member of a replica set to an earlier version of Access;
- module code that relies on Access 2000 add-ins or libraries (such as ADO 2.1 or later) needs to be modified;
- a database converted to Access 97 loses:
  - data access pages. There is no workaround for this;
  - Unicode support;
  - support for the decimal data type;
  - support for ADO. You must use DAO or RDO instead.

Question [385] – FTP to NICE from outside CERN

How can I "ftp" to NICE from outside CERN?

Answer

General access to files from outside CERN is *not* supported. However, all new home servers (`CERNHOME01 -> 08`) have an incoming access, so that users can use "ftp", like this (example for `CERNHOME01`):

```
ftp cernhome01.cern.ch
```

(login with NICE account name and password)

If you don't know your home directory server, consult the NICE phonebook (type your name, then click on "*more info*"), or, if you cannot do it, contact the helpdesk: `helpdesk@cern.ch` (tel. +41/22/767.8888)

Question [401] – Printing from a NON-CERN (visitors) W2000 portable

Is there a way to print quickly without connecting to the CERN/NICE domain (e.g. when using a Windows 2000 portable PC)?

Answer

The Printing Package does it all for you. You can just download the client package in FTP mode from the following site: `http://cern.ch/printpackage`

Go to the User's Guide and down-load the FTP Mode self-extracting installer for Windows 2000, called `SetupFTPW2K.exe`

Running this program as Local Administrator will install the package and create an icon in the system tray. In this mode you do NOT need to be connected to the CERN domain.

Question [411] – `SQL*Loader` on NICE

Is `SQL* Loader` available on NICE? If yes, how do I invoke it?

Answer

There are two ways to access `SQL*Loader` on NICE:

1. From a normal MS-DOS windows, you can call `sqlldr`, `sqlldr73`, or `sqlldr80` (it depends on the target database version).
2. In the `Oracle Basics` menu (`Start - Database Management - Oracle Basics`), you find an icon called `Oracle Data Manager`. You run that, and you have `Export, Import` and `Loader` in the same tool.

Question [427] – Office 2000 - where have my menus gone to?

Office 2000 sometimes seems to lose some bits of the menus I was used to get on Office 97. For example, in Word 2000, my 'Format' menu only shows 2 entries - where are the others?

Answer

The entries in this and other menus are not lost, they are only hidden. If you move your cursor over the 'double-down' arrows, they reappear.

Microsoft programmers thought this would make life easier for users, as most of them only use a small percentage of all possibilities the items contain. But indeed others get irritated by this behaviour and feel that a quick and efficient usage of Office 2000 is not possible.

To overcome this, here is the recipe:

Within Word, click on 'tools', then 'customize'. In the Options tab, un-tick 'menus show recently used commands first'.

All menus are now back alive and shown. And not only for Word, this also 'reactivates' them for the rest of the Office 2000 package, like Excel, Powerpoint etc...

Question [468] – W2000 portable DHCP connection to CERN domain

I have connected my portable PC (running Windows 2000) to a CERN DHCP socket foreseen for portables, but I cannot access the CERN domain or see any of its applications.

Answer

Computers need to be registered correctly in LANDB for Windows 2000 to work and deploy applications. In this case, the computer had not been registered as portable from the network database.

You can check your computer configuration at the URL `http://winservices.web.cern.ch`: select then

"`Computer account status`", which will display in red incorrect settings.

The PC has to be registered correctly and you will have to wait for the active directory synchronisation to complete.

Question [472] – Restricted login on PCs

Is there a possibility to restrict user logins to certain PC's, so that not everyone can login to a given PC?

Answer

We cannot centrally restrict users to login on certain machines. This was previously possible with Netware on W95, but it is not anymore possible on WNT and W2000.

However, on NT, you can still do that locally, by selecting:

```
Start - Programs - Administrative Tools
  - User Manager
```

Then select the menu:

```
-> Policies -> User Rights...
```

choose "`Log on Locally`" in the drop-down list for the rights, remove "`Everyone`" from the list below, and add the account(s) you want to be able to login to this PC.

UNIX Support

Roger Woolnough and Nicole Crémel, IT/User Support)

Question [349] – LSF requests on Solaris and Linux

Is it possible to select on which architecture some LSF request will be executed (e.g., on the CMS cluster)?

Answer

In the LSF CMS (Solaris/Linux) cluster, all requests submitted from `cms.wgs` are only going to `Solaris` nodes. In the same way, all requests submitted from `cmslx.wgs` are only going to `Linux` ones.

It is possible to force LSF requests to be executed in another type of architecture, with the option `-R solaris`, or `-R linux` (or even `-R hpux`) on the `bsub` command line.

N.B. All requests issued from `hpplus` (to `zh_???` queues) are targeted to run in `shift42`. To force them to run under Solaris systems, the option `-R solaris` has to be used (or `-R linux`) to get them in Linux systems.

Question [352] – Fortran and `hepf77` command on Linux

Is the `hepf77` command available on Linux?

Answer

The port for this command was done in 1999 and it has been available since at least February 2000. It provides easy portable access to g77, the default CERN supported Fortran 77 compiler.

Please note that using this `hepf77` command together with the `cernlib` command is the standard way to link an application calling CERNLIB on all central platforms at CERN (e.g., RSPLUS, HPPLUS, DXPLUS and more recently LXPLUS), to avoid any "undefined" reference (and setting the correct flags and options). For instance, if you want to link an application `psimple.f` with PACKLIB, just type:

```
hepf77 -o psimple psimple.f `cernlib packlib`
```

(Be careful with the back quotes, standard UNIX feature!)

This will produce an executable file `psimple`.

Question [362] – Access to Software repository (e.g. GNU tool)

I would like to install locally on my UNIX workstation a tool which is available in ASIS. Where can I get this tool ?

Answer

All CERN software which are "public domain" (like the GNU tools) are available from the `asisftp` server, at URL:

```
ftp://asisftp.cern.ch
```

Note that GNU and public domain software are stored under the sub-directory `pub` (`ftp://asisftp.cern.ch/pub`). ASIS software is access free (unprotected access).

Question [368] – X11 resources file

What is the default resources file for X11 (X-windows), as set by the HEPiX X11 scripts at CERN ?

Answer

To see the current resources of your X server, type: `xrdb -q`

To see the resources that the HEPiX scripts used the last time you logged in, look at:

```
~/.hepix/generated_xresources.
```

To change some X resources, edit the file:

```
~/.hepix/xresources.
```

Question [372] – Exceed - Cannot open default font

When I try to start Exceed v6(new) I get the following error:

```
Fatal Error: Cannot open default font
```

and then Exceed exits. What should I do ?

Answer

There are several things to test:

- This might be just a temporary problem, and you may try first to reboot the PC.
- Check if the network is OK.
- Another possible explanation from *Hummingbird* technical support is:

```
Apparently, the problem comes from having too many
fonts installed on the Workstation. Windows has a
limit for fonts,  and when you hit the limit,
Exceed can't load the default font (6x13).
--------------------------------------------------
Please make sure he is able to view the font 6x13.
Go to xconfig->font-> font database->misc and make
```

```
sure he is able to view the font 6x13. Also search
for entire system, if you have more than one 6x13.fon
file, please only keep the one in exceed\ font\ misc
directory. Please go to control panel->fonts. If you
find a lot of fonts here, please delete some of them.
Windows 95 has a limit of 1000 fonts. Certain
products like Corel Draw install many fonts. It is
recommended that the number of Corel fonts installed
is not greater than 500. You will have to delete some
of these fonts, preferably leaving less than 500
fonts in the Control Panel. Check if the following
is true under Xconfig->fonts->font defaults:
Fixed = Fixed cursor = cursor
```

Otherwise you should ask the Helpdesk (78888) to send you someone from the PC local support of your division, who will have to come and see your PC.

Question [375] – LSF job and AFS token

Can you please explain why my LSF job failed with the following error:

```
No AFS token
@(#)CERN job starter 2.07 16/4/2000
```

Answer

The current mechanism to retrieve and renew AFS token in batch jobs has some weaknesses, in particular the fact that no retry is performed in case there is a failure in connecting to the authentication server. Any overload of the server, temporary network problem or even AFS cache corruption is likely to be a possible explanation.

The most frequent explanation is that too many jobs start at the same time; then some of them may not get AFS tokens as the AFS server may have problems to handle all the requests. This is a **known* "feature"* of the LSF/AFS combination. The work-around is NOT to submit too many batch requests at the same time. In your submission script, try to leave a 1 minute period between two `bsub` commands.

Some intervention to reduce the probability of failures has already been identified, and is in the action list of the AFS support team in IT, but its feasibility depends mainly on usual constraints as priorities and resource allocation.

Mail support

(Michel Franceschi and Miguel Marquina, IT/User Support)

Question [353] – "pine" from a login different than the mail one

How to use "pine" from a login different than the mail one?

Variant 1

I am a PLUS/WGS user of "pine" and I have several mail accounts I'd like to manage from a "single location". The point is that in spite of defining additional INBOX entries for those accounts, I keep getting questioned ad nauseam *to supply the password for anything (opening a folder, the inbox, etc) related to a login different to mine. "pine" is really unusable in this way! Help!*

Variant 2

Unfortunately my PLUS/WGS loginid is different to my mail one because I just changed of experiment. I can accept that I am asked to supply my password when opening my mail with "pine". But only once! However "pine" keeps asking me for giving the login and password again and again, even to access my local folders. Can I hope that this improves?

Answer

When opening a mail session with an IMAP-compliant client such as "pine", two distinct dialogues get established: one to deal with the INBOX, another to deal with the existing "folder-collections". The point is that "pine" can be told to open an INBOX and/or folders of an account different to the usual one (upon provision of a valid password of course), but this is relatively tricky.

Technically speaking the pine configuration parameters would look something like:

```
inbox-path={loginid.mailbox.cern.ch/user=loginid}inbox
 folder-collections=
    "{loginid.mailbox.cern.ch/user=loginid}~/mail/[]"
```

If you don't want to bother mangling with pine configurations, you might welcome "xpine", a modest CERN wrapping/extension to "pine" to make the above in some transparent manner. Simply execute: `xpine -login loginid`, which will open a second "pine" session (eventually at a separate window) for the wanted loginid. A man page for "xpine" is also available.

Question [391] – What is UALA?

I have heard of a web interface to view the archives of a mailing list, something called UALA. What is it?

Answer

UALA stands for *Unlimited Access to Listbox Archives*

There are several parameters controlling the behaviour of the web archives associated to a mailing list. All of them can be edited by the list owner(s) via the SIMBA interface:

- *UALA Status* (Whether web archives are enabled or not).
- *UALA Template* (HTML Template to use for web archives).
- *UALA Authentication* (Type of authentication required to access web archives).
- *UALA List Access* (to grant access to the archives to people other than list members and authenticated users).

Web Publishing

(Maria Dimou, IT/User Support)

Question [371] – `cgi-bin` directory on Personal Web sites

My Web pages are under AFS and were not created using the new automatic registration. I would like the `cgi-bin` directory to be enabled. What should be done ?

Answer

You should delete and recreate the site using the Web registration form, at URL: `https://web0.cern.ch/WebRegister/RegistrationStep1.asp`

Question [379] – Netscape problems on X-terminal

When I run Netscape on HPPLUS and using an X-terminal I am flooded by this kind of message:

```
netscape:
 X Error of failed request:  BadPixmap (invalid ... )
   Major opcode of failed request:  54 (X_FreePixmap)
   Resource id in failed request:  0x1c004df
   Serial number of failed request:  5742
   Current serial number in output stream:  5817
   Widget hierarchy of resource: unknown
```

This prevents me from using it in a reasonable way. I am using: `/usr/local/bin/netscape` (version 4.7)

Answer

Most probably it is a problem with your X-terminal, and it might simply be a lack of memory in order to run an X-window application (quite consuming) like Netscape.

You should check whether you have the same problem using another device (e.g. X-terminal or PC with Exceed), and if this is happening on all PLUS services. If it works fine on another X-terminal with more memory, then you should contact the Helpdesk in order to have your X-terminal upgraded with more memory (if this is possible).

Text Processing

Question [377] – CERN Conversion Service

What is the "CERN Conversion Service*"?*

Answer

The CERN Conversion Service exists to provide a facility for all CERN members to easily convert a document from one given format to another.

All you need to do is choose the format conversions you desire, leave your email address and upload your file, and the system will do the rest.

Using the Web-based interface it is possible to upload single files one at a time, or several files collectively as a `Tar` or `Zip` archive. All conversions should complete within an hour or they will be cancelled automatically by the system. Notification is sent notification by email when conversions are completed.

See all details, and access at URLs:

```
http://documents.cern.ch/Convert
http://documents.cern.ch/Convert?upload
```

If you have problems you can contact the people responsible at `Convert.Support@cds.cern.ch`.

Question [384] – Acrobat 4 (and `acroread`) on UNIX platforms

What are the plans to put Acrobat 4 in production on UNIX platforms?

It seems that on NICE `acrobat reader` is version 4.0 while on Unix (e.g. LXPLUS) it is still version 3.0.

Answer

Status in March 2001:

The `acrobat reader 4` on Unix with AFS is suffering the same bug as was `ghostview` (and `gv`) some years ago: i.e., when it runs in a directory it tries to discover the full tree structure. This, for a user with an AFS home directory, is fatal (as AFS is too big for that): it takes a huge amount of time, but it can even fail when it tries to follow all links (which causes it to crash in case of 'circular' links).

Conversely to `ghostview` some years ago, *Adobe* (the owner of `acrobat reader`) is not free software (and not open source), thus we cannot correct the problem. Unfortunately it seems that *Adobe* has no plan to correct this.

At CERN, on ASIS, a small wrapper script installed with `acrobat 4.05` tries to prevent this situation: if the command `acroread` is invoked without parameters the script changes its current working directory to `$HOME/.acroread` and only then launches real binary. But this workaround is not fully satisfactory and it has been decided to keep `acrobat reader 3.0` as the production version on Unix.

Anyhow, if you want `acroread 4`, it is available on ASIS but just as "`Certified`". It is possible to get it on your own workstation, by login as `root` and using `tkwsm` (see `http://cern.ch/asis/primer`), select `X11/acrobat` as Latest version and re-run `ASISwsm`.

Question [517] – p-bar (anti-proton) sign in M/S Office documents

I frequently need to use the `p-bar` (anti-proton) character in Word *documents (i.e. a small p with a bar on top of it) but this symbol is not defined on any of the fonts available on PCs.*

Answer

A possible solution/workaround is the following: inside the application, go into `Insert/Object/MS Equation Editor`, and type the letter `p` . Then use the tool `symbol` to insert the bar on top of the p. `Copy/paste` it so that it stays in the clipboard for multiple usage.

You can also use this method for other M/S applications, e.g. *Excel*. Unfortunately, it is not possible to create a macro containing this special character.

About the author(s): *Nicole Crémel is working in the User Support group and, with Roger Woolnough, is responsible for the supervision of operations of the Computing Helpdesk. Editing and revising the "Question and Answers" database is a daily task of most members of the group.*

8. User Documentation

Everything related to "computer documentation": updates to computer documentation distributed by IT (including the User Support reference cards), offerings of the Computing Bookshop, etc.

8.1 Some News and History from the Computing Bookshop

Roger Woolnough, IT/User Support

Abstract

How did the Computing Bookshop start.

The bookshop began modestly in 1994 with the vision of providing up to date documentation for users who were moving towards Unix and C++. It began as an off-shoot of the stores where new books were quickly bought but it took time to replace them. It was quickly clear that this type of service was well received by users and despite in the first year only having an agreement with what was *Addison Wesley* we managed sales of 350 books.

What started as a small operation quickly grew as we saw the need to add new publishers. In 1996 we were 23rd in European sales for *Addison Wesley* and 11th for *O'Reilly*. Remember we had no shop, just shelving in what was the defunct "*User Consultancy Office*", and books were just in piles for people to buy as required. The thought of a 30-45% discount was also a reason for good sales. Notwithstanding this here are a few interesting facts:

1. The best selling book we have is written by a CERN staff member, Michel Goossens, "*The Latex Companion*", with almost 3000 copies sold!
2. The quickest selling book was the first Java book ever produced: we sold 200 in 2 weeks!
3. We have little demand for French books but sold 350 copies of "*Word et Excel Comment faire...*" in 2 months in 1998.
4. Our legacy to the World is that since publishing companies have seen our success, some now propose similar arrangements to large corporations.

By 1997 we were selling up to 3000 books per year and the people working at the *uco* could not do two jobs at once, sell books and provide support! With the closure of the *Computer Science Library* (CSL) we were lucky to start anew with a proper bookshop in 513-1-022 and Jutta Megies the librarian from the CSL taking over the day to day running.

In the last 4 years we have sales of between 2500 and 3000 books per annum and have added to our stock of computing titles with dictionaries and a small library of physics books, some sold in conjunction with the CERN Reception in Building 33. We try to offer a selected choice of specialised books but are always pleased to hear from users when they have 'tips' for new titles. We like to think of the shop as a service for the user's with a choice steered by themselves and current CERN computing policy.

Since the last CNL we have created a new account with *Hungry Minds*. This used to be *IDG* books, famous for their *Bible* series. These books have always sold well as they provide a tutorial and reference in one book and are extremely complete on individual subjects. We have a full range in stock for *Office 2000 Applications* and more and we invite you to scan the stock list for more details.

Other new books available include two new *O'Reilly* titles, *SQL in a nutshell* and *SSH the definitive guide*. One title has been withdrawn by this publisher (*Building linux clusters*) as it contained some wrong information and is now being rewritten. Two other books of interest are now available *C++ Standard Template Library* and *Adobe Framemaker 6, Classroom in a Book*.

We would like to thank all our clients for their kind remarks and their custom over the years and we look forward to improving the service for you still further in the future. Please get in touch with us if you have information on new titles that could be interesting to purchase, by writing to `Bookshop@cern.ch`.

8.2 User Support Book Catalogue

Roger Woolnough and *Jutta Megies, IT/User Support*

Situated in Building 513 1-022 (tel. 74050) is the Computing Bookshop provided by the *IT / User Support group*, where CERN users can find computer books and CDs at discount prices. The service is open weekdays from 8.30 to 12.30 or contactable by e-mail to `Bookshop@cern.ch`.

Books are purchased from some 15 publishing houses and the catalogue offers a selection of documentation aimed at the range of computing utilities available at CERN. The service welcomes suggestions from the user community for new acquisitions. Purchasing may be done internally via EDH or TID or alternatively cash payments via the CERN Bank.

The list, with all the relevant information, is compiled regularly at URL: `http://cern.ch/consult/books`

9. Just For Fun...

Some jokes to end this CNL, and make more fun ... for those who did not read them already, somewhere else!

9.1 Euh? When did the Millennium Start?

The CNL Editor, IT/User Support

Abstract

If you feel like being polemic with your colleagues over a coffee, we offer you something to debate about and warm up your temperaments.

This article is based on information found by browsing the Web, and extracted mainly from the two following pages:

1. "*Start of the next millennium... an astronomer's answer*" (`http://www.fiz.uni-lj.si/astro/Y2K.html`), by Tomaz Zwitter, Astronomical Observatory of University of Ljubljana, Slovenia.
2. "*The New Millennium Started On 1/1/2001*" (`http://staff-www.uni-marburg.de/~schittek/millenni.htm`), a compilation made by Walter Schittek, Philipps-Universität Marburg, Deutschland.

The debate is more or less over, but, for one year and three months, you have probably been involved in some animated discussion around this issue. For most of the journalists in all European countries there was no doubt: January 1st 2001 had signalled the start of the XXIth century and the 3rd millennium. But can we trust journalists? Obviously this choice helped them to produce again, one year after the big events for 2000, attractive titles on their newspapers, always good for marketing and business. We tried, in User Support, to search on the Web in order to get some scientific explanation on the subject and here are some sensible conclusions (historical and scientific) we could extract.

The statement which is made to prove that we started a new millennium on 1st January 2001 (and not 1st January 2000) is that the date calculation has started with 1/01/0001 and not with 1/01/0000, forgetting at the beginning to define a "Year 0". If you start counting at the year 0, the third millennium started with the year 2000. We count hours, minutes, miles and ages of people beginning at 0, why not calendar years?

Gregorian Calendar versus astronomical chronology

In 1582, the Gregorian Calendar (`http://www.ptb.de/english/org/4/43/432/greg.htm`) was created, which defines the chronology still in use today. Counting of the years "after the birth of Christ" (A.D.) was already established at that time: it was invented by a scholar called Dionysius Exiguus in the 6th century (525 AD). Before that, there was (besides others) the varronic calculation which started counting from the foundation of Rome. A.D. 1 is equivalent to the varronic year 754. The varronic year 1 became then 753 B.C., i.e., there is no year 0, the year 1 B.C. being directly followed by A.D. 1 (Sequence ... 3 B.C., 2 B.C., 1 B.C., A.D. 1, A.D. 2, A.D. 3 ...). At first sight, that could be interpreted as an error, simply because the Roman numerals had no number zero! But many people today think it is not a mistake, but that this "ordinal counting" (enumerating the elements and naming them with their order, i.e., the 1st, the 2nd, etc... or element 1, element 2, ...), versus "cardinal counting" (0,1, 2, etc...) was correctly used for calendaring: cardinal counting gives the value 0 if there is not yet one complete element, whereas the first incomplete element already gets the ordinal number 1. Year 0 in cardinal counting becomes 1st year in ordinal counting!

However, the lack of a year 0 was disturbing for the astronomers in their need for a continuously extendible time axis. Astronomers frequently need to calculate how many days passed between the given two dates, for example when calculating times of eclipses of binary stars. Calendar date is not practical for this purpose, because of all the complications with different month lengths and leap years. Consequently they invented the astronomical chronology. And there you have a year 0, which is equal to 1 B.C., there are even negative years, where the year -1 is equal to 2 B.C. and so on.

Everybody agrees however that a century is a period of 100 years, and a millennium a period of 1000 years. Then, this quick and short review of only 2 calendaring conventions (one historical and one scientific) proves that the answer to the question depends on what you consider to be the origin: if you are a 000-fan (i.e., as on your car, with the switching of the counter to "000"), it is 2000, but if you want to wait 2000 years after January 1st of year 1AD, it is of course January 1st 2001.

Note that there are many other calendars in use than the Gregorian Calendar, e.g. Jewish, Moslem, Buddhist, that started the numbering of years at a different time than the Julian/Gregorian calendar, and therefore do not celebrate the 3rd Millennium at the same time!

Historical Statements

The birth of Jesus Christ

Today's knowledge says that Jesus Christ was born at least 2 years before year 1 which means that those who wanted to celebrate the 2000th birthday of Jesus Christ should really have done that during the nineties. This never lead however to a redefinition of chronology in any country of the world.

2nd millennium

Already before the year 1700, people quarrelled about when the new century would begin. Before each turn of the century, there were some intense confrontations between differing majorities. 100 years ago, nearly all official fixings were made for 1/1/1901, except in Germany, which was caused by a preference of Kaiser Wilhelm II. On December 14, 1899, the German Bundesrat decided that the 1/1/1900 had to be taken as the beginning of the 20th century, which means that the end of the 2nd millennium must have been on December 31, 1999. Ultimately, that was a redefinition of the term "century": e.g. the (German) "dtv Brockhaus Lexikon" writes about the term "century": ... *the 20th century began on January 1, 1901 and will finish on December 31, 2000.*

3rd millennium

For the present 3rd millennium, the U.S. government issued a statement referring to the U.S. Naval Observatory (`http://psyche.usno.navy.mil/millennium/whenIs.shtml`), with the result 1/1/2001. Similar statement were made by the Royal Greenwich Observatory (`http://www.compinfo.co.uk/y2k/greenwch.htm`), UK, and most other countries. The German Bundestag did not have to decide this question, because the Physikalisch-Technische Bundesanstalt has already given clear statements, with the same result!

What about the *4th millennium*? Unfortunately, or fortunately, we will not be there to launch the same polemic again.