

Australian UNIX systems User Group Newsletter

AUUGN

Volume 12, Number 2/3

August 1991

The AUUG Incorporated Newsletter

Volume 12 Number 2/3

August 1991

CONTENTS

AUUG General Information	3
Editorial	5
AUUG Institutional Members	7
President's Report	9
Election Results	10
AUUG'91 Biographies of Invited Speakers	11
AUUG'91 Tutorials	13
ASCnet Survey	14
Report on EurOpen Spring 1991 Conference	17
Task Tracking Systems - Watch Dog and Friend	23
What's Happening with ISODE, the free implementation of OSI?	31
Five Star Service - With Mainway Transport and UNIX	36
UNIX Terminology Made Easy	41
From <i>;login</i> - Volume 15, Number 6	47
Book Reviews	47
Designing Object-Oriented Software	47
Software Engineering Concepts and Management	48
An Update on UNIX-Related Standards Activities	49
From <i>;login</i> - Volume 16, Number 1	70
Report on ISO POSIX Working Group	70
Book Review	76
lex & yacc	76
An Update on UNIX - Related Standards Activity	77
From the <i>EurOpen Newsletter</i> - Volume 11 Number 1	88
Call Doc Strange	88
Standards Column	93
Management Committee Minutes - 14th February 1991	95
Management Committee Minutes - 26th April 1991	100

Management Committee Minutes - 28th June 1991	103
AUUG Membership Categories	109
AUUG Forms	110

Copyright © 1991 AUUG Incorporated. All rights reserved.

AUUGN is the journal of AUUG Incorporated, an organisation with the aim of promoting knowledge and understanding of Open Systems including but not restricted to the UNIX system, networking, graphics, user interfaces and programming and development environments, and related standards.

Copying without fee is permitted provided that copies are made without modification, and are not made or distributed for commercial advantage. Credit to AUUGN and the author must be given. Abstracting with credit is permitted. No other reproduction is permitted without prior permission of AUUG Incorporated.

* UNIX is a registered trademark of UNIX System Laboratories, Incorporated

AUUG General Information

Memberships and Subscriptions

Membership, Change of Address, and Subscription forms can be found at the end of this issue.

All correspondence concerning membership of the AUUG should be addressed to:-

The AUUG Membership Secretary,
P.O. Box 366,
Kensington, N.S.W. 2033.
AUSTRALIA

Phone: (02) 361 5994
Fax: (02) 332 4066

General Correspondence

All other correspondence for the AUUG should be addressed to:-

The AUUG Secretary,
P.O. Box 366,
Kensington, N.S.W. 2033.
AUSTRALIA

Phone: (02) 361 5994
Fax: (02) 332 4066
Email: auug@munniari.oz.au

AUUG Executive

President **Pat Duffy**
pzd30@juts.ccc.amdahl.com
Amdahl Pacific Services Pty. Ltd.
1 Pacific Highway
North Sydney NSW 2000

Vice-President **Chris Maltby**
chris@softway.sw.oz.au
Softway Pty. Ltd.
79 Myrtle Street
Chippendale NSW 2008

Secretary **Rolf Jester**
rolf.jester@sno.mts.dec.com
Digital Equipment Corporation
(Australia) Pty. Ltd.
P.O. Box 384
Concord West NSW 2138

Treasurer **Frank Crawford**
frank@atom.lhrl.oz.au
Australian Supercomputing Technology
Private Mail Bag 1
Menai NSW 2234

Committee Members **Andrew Gollan**
adjg@softway.sw.oz.au
Softway Pty. Ltd.
79 Myrtle Street
Chippendale NSW 2008

Glenn Huxtable
glenn@cs.uwa.oz.au
University of Western Australia
Computer Science Department
Nedlands WA 6009

Peter Karr
Computer Magazine Publications
1/421 Cleveland Street
Redfern NSW 2016

Michael Tuke
mjt@anl.oz.au
ANL Ltd.
432 St. Kilda Road
Melbourne VIC 3004

Scott Merrilees
Sm@bhpesse.oz.au
BHP Information Technology
P.O. Box 216
Hamilton NSW 2303

AUUG General Information

Next AUUG Meeting

The AUUG'91 Conference and Exhibition will be held from the 24th to the 27th of September, 1991, at Darling Harbour, Sydney. The AGM of AUUG Inc. will be held during the conference. Biographies of invited speakers and a brief description of the tutorial sessions are printed in this issue of AUUGN.

The AUUG'92 Conference and Exhibition will be held from the 8th to the 11th of September, 1992, at the World Congress Centre, Melbourne.

AUUG Newsletter

Editorial

Welcome to my first issue of AUUGN.

First of all I would like to thank the retiring editor, David Purdue, for his help in the hand-over of the Newsletter, and wish him all the best for his future.

This issue is a combined issue 2 and 3. This is due to the fact, as members are aware, we are more than half-way through the year and only one issue has appeared so far.

We are looking at introducing some changes to the format of the newsletter, but for the next few issues the format will remain unchanged. If you have any ideas as to what should, should not be included please send me some mail at the address mentioned below.

Unfortunately, no book reviews appear in this issue. I am going to continue this in the near future, so please let me know if you are interested in reviewing books.

Finally I need more papers, articles, *etc.* for inclusion.

Jagoda Crawford

AUUGN Correspondence

All correspondence regarding the AUUGN should be addressed to:-

AUUGN Editor
PO Box 366
Kensington, NSW, 2033
AUSTRALIA

E-mail: auugn@munnari.oz.au

Phone: +61 2 543 2552
Fax: +61 2 543 5097

Contributions

The Newsletter is published approximately every two months. The deadline for contributions for the next issue is Friday the 13th of September 1991.

Contributions should be sent to the Editor at the above address.

I prefer documents to be e-mailed to me, and formatted with troff. I can process mm, me, ms and even man macros, and have tbl, eqn, pic and grap preprocessors, but please note on your submission which macros and preprocessors you are using. If you can't use troff, then just plain text or postscript please.

Hardcopy submissions should be on A4 with 30 mm left at the top and bottom so that the AUUGN footers can be pasted on to the page. Small page numbers printed in the footer area would help.

Advertising

Advertisements for the AUUG are welcome. They must be submitted on an A4 page. No partial page advertisements will be accepted. Advertising rates are \$300 for the first A4 page, \$250 for a second page, and \$750 for the back cover. There is a 20% discount for bulk ordering (ie, when you pay for three issues or more in advance). Contact the editor for details.

Mailing Lists

For the purchase of the AUUGN mailing list, please contact the AUUG secretariat, phone (02) 361 5994, fax (02) 332 4066.

Back Issues

Various back issues of the AUUGN are available. For availability and prices please contact the AUUG secretariat or write to:

AUUGN Inc.
Back Issues Department
PO Box 366
Kensington, NSW, 2033
AUSTRALIA

Also please note that the prices for back issues published in AUUGN Vol 12 No 1 are incorrect.

Acknowledgement

This Newsletter was produced with the kind assistance of and on equipment provided by the Australian Nuclear Science and Technology Organisation.

Disclaimer

Opinions expressed by authors and reviewers are not necessarily those of AUUG Incorporated, its Newsletter or its editorial committee.

AUUG Institutional Members

(NSW) Department of Minerals & Energy
AIDC Ltd.
ANSTO
ANZ Banking Group/Global Technical Services
Adept Software
Amdahl Pacific Services
Apple Centre Brisbane
Ausonics Pty Ltd
Australia Eds Pty Ltd
Australian Airlines Limited
Australian Bureau of Agricultural and Resource Economics
Australian Eagle Insurance Co. Ltd
BEVFET
BHP
BHP CPD Research & Technology Centre
BHP Information Technology
BHP Research - Melbourne Laboratories
Bain & Company
Burdett, Buckeridge & Young Ltd.
Bureau of Flora and Fauna
Bureau of Meteorology
Burns Philp Plumbing Supplies Group
CITEC
Centre for Info Tech & Comms
Codex Software Development Pty. Ltd.
Com Tech Communications
Commercial Dynamics
Communica Software Consultants
Comperex(NSW) Pty Ltd
Computer Science of Australia Pty Ltd
Computer Software Packages
Crane Enfield Metals Pty Ltd
DMP Software Pty Ltd
Data General Australia
Deakin University
Defence Housing Authority
Department of Industrial Relations & Employment
Department of Transport
Department of Treasury and Finance
Dept of Employment, Vocational Education & Training
Dept. Of The Premier & Cabinet
Dept. of Conservation & Environment
Dept. of Defence
Digital Equipment Corp (Australia) Pty Ltd
Duesburys Information Technology Pty Ltd
ESRI Australia Pty Ltd
Eastek Pty Ltd
Electronics Research Labs
Emulex Australia Pty Ltd
Exicom Australia Pty Ltd
Expert Solutions Australia
FGH Decision Support Systems Pty Ltd
Fremantle Port Authority
Fujitsu Australia Ltd
Geelong and District Water Board
Grand United Friendly Society
Hamersley Iron Pty. Limited
Harris & Sutherland Pty Ltd
IPS Radio & Space Services
Infonetics
Ipec Management Services
Kodak (Australasia) Pty Ltd
Leeds & Northrup Australia Pty. Limited
Macquarie University
McDonnell Douglas Information Systems Pty Ltd
Metal Trades Industry Association
Ministry of Housing & Construction (VIC)
Mitsui Computer Limited
Multibase Pty Ltd
NRIC
OPSM
Oracle Systems Australia Pty Ltd
Pact International
Port of Melbourne Authority
Prime Computer
Qld Justice Department
Radio & Space Services
S.A. Institute of Technology
SBC Dominguez Barry
Seqeb Control Centre
Shire of Eltham
Signum Software Pty Ltd

AUUG Institutional Members

Silicon Graphics Computer Systems
Snowy Mountains Hydro-electric Authority
Software Development International Pty Ltd
Softway Pty Ltd
Sony (Australia) Pty Ltd
Sphere Systems Pty Ltd
St Vincent's Private Hospital
Stallion Technologies Pty Ltd
Stamp Duties Office
Steedman Science and Engineering
Sugar Research Institute
TUSC Computer Systems
Tandem Computer Pty Ltd
Tasmania Bank
Tattersall Sweep Consultation
Tech Pacific
Technical Software Services
Telecom Australia
Telecom Network Engineering Computer
Support Service
The Opus Group
The Roads and Traffic Authority
The University of Western Australia
Toshiba International Corporation
Pty Ltd
TurboSoft Pty Ltd
UCCQ
Unisys
University of Adelaide
University of Melbourne
University of New England
University of New South Wales
University of Sydney
University of Tasmania
Vicomp
Wacher Pty Ltd
Wollongong University
WordPerfect Pacific
X = X Pty Ltd
Yartout Pty Ltd

AUUG President's Report

This is the first report to AUUGN I have made since taking over the Presidency from Greg Rose last December. It's certainly been busy time, and I believe our efforts are starting to reap rewards.

AUUG was at something of a crossroads late last year. The organisation suffered a lot with the loss of, first, Tim Roper as Secretary and John Carey from the Committee, and then, Greg, when he left for the United States. Then, throughout last year and early this year, a number of Committee members found themselves in work and personal situations such that they had very limited time available to spend on AUUG business.

Out of this adversity come some very productive soul-searching on the part of the Committee, leading to our decision - finally! - to appoint a Secretariat (our old friend, ACMS) to deal with the day to day issues of membership, enquiries, mailings, *etc.* This freed the Committee to focus on the very important issues of a membership that had been allowed to deteriorate or become unfinancial and the services that we should be providing to our members. We also started to examine ways in which AUUG could continue to be the relevant voice for all open systems users in Australia - something of a challenge when you consider how the market has changed during the last ten years or so.

You may have noticed a great increase in AUUG's public profile of late. The Committee decided to appoint a public relations firm - Symmetry Design - to work with us this year on the lead up to AUUG'91 and this is proving to be more than justified.

Symmetry is responsible for the greatly improved look of all AUUG material and for the advertising and publicity that has been appearing in the various computer trade press publications. They have been working closely with us to ensure that AUUG'91 is truly professional event embracing all aspects of open systems, and that the needs of attendees, exhibitors and speakers are met.

The AUUG'91 advertising has already started to bear fruit, with in excess of 100 enquiries about the conference received by ACMS to date - even before the Conference Brochure goes out!

We have a long way to go, but an excellent base in place for the future. We thank all of you for continued support and participation, and believe we can look forward to a User Group that continues to grow in size, relevance and member benefits.

AUUG General Committee Election

The result of the Elections for the positions of AUUG General Committee member and also the approval of Affiliation with UniForum is as follows:.

Positions previously filled (unopposed):

President:	Pat Duffy	(pzd30@juts.ccc.amdahl.com)
Vice-President:	Chris Maltby	(chris@softway.sw.oz.au)
Secretary:	Rolf Jester	(rolf.jester@sno.mts.dec.com)
Treasurer:	Frank Crawford	(frank@atom.lhrl.oz.au)

The successful candidates were:

General	Andrew Gollan	(adjg@softway.sw.oz.au)
Committee	Glenn Huxtable	(glenn@cs.uwa.oz.au)
Members:	Peter Karr	
	Micheal Tuke	(mjt@anl.oz.au)
	Scott Merrilees	(Sm@bhpese.oz.au)

The new AUUGN editor is: Jagoda Crawford (jc@atom.lhrl.oz.au)

Public Officer:	Robert Elz	(kre@munnari.cs.mu.oz.au)
Returning Officer:	John O'Brien	(john@wsa.oz.au)

AUUG Secretariat: (02) 361 5994

Affiliation with UniForum: Carried Yes: 72 No: 2

SPEAKERS

Robyn Williams

Robyn Williams is perhaps the most respected science journalist and broadcaster in Australia.

As Executive Producer and presenter of the ABC radio programme "*The Science Show*" since its inception in 1975, Robyn has taken science from the domain of laboratories and academic institutions and brought it into the realms of everyday life. Topics range from highly technical innovations and sensitive issues such as schizophrenia through to the hoax interview on the 500th edition with a bogus scientist on a supposedly newly evolved surgical procedure involving the brains of politicians! In 1988 the show received two Pater Awards - one for a single programme and the other for best regular radio programme.

Among the many accolades Robyn has been awarded are the Radio Prize from the Human Rights commission, a United Nations Media Prize and the Michael Daly Award for Science Journalism which was awarded at the ANZAAS Congress. He was awarded the Order of Australia in the bicentennial Honours List.

Robyn is also a Commissioner on the Commission for the Future and President of the Australian Museum. In 1986 he was awarded an Honorary Doctorate of Science from Deakin University and in 1988 Doctorates of Science from Macquarie and Sydney Universities.

Marie Burch

Marie Burch is the Director of International Operations for the Open Software Foundation. This position takes advantage of her vast experiences spanning many years in the information technology industry. She began her career in the "Buck Rogers" environment of the space program with McDonnell Douglas in 1963, exploring the challenge of space travel on the Mercury and Gemini programs in the guidance and control engineering group. In the seventies, she took the leap from engineering into commercial computing equipment.

Marie was introduced to the international computer market through her work with the Olivetti Corporation in Italy and with Datasab Systems in Sweden. It became clear to her that the international environment is a key component of successful business solutions. Surviving the birthing and the "terrible two's" at AT&T American Bell, Marie renewed her Italian working relationships during the venture between AT&T and Olivetti.

In January 1990, Marie joined the Open Software Foundation and has continued her industry participation in the UNIX arena while focusing beyond UNIX to the broader Open Systems Environment. The technologies delivered by OSF are key components which enable Open Systems to become a reality, bridging the installed proprietary base systems into this new environment.

Peter Cunningham

Peter Cunningham is the President and Chief Executive Officer of UNIX International, and was instrumental in the organisation's founding in January 1989. Mr Cunningham represents over 200 corporate members and assumes full responsibility for implementing their decisions on the evolution of UNIX Systems V. He is responsible for the continued growth of this open system standard as a replacement for proprietary operating systems.

Before accepting this position, Mr Cunningham served for five years in the United Kingdom as the Director Manager of Office System Strategy for ICL Limited. In this position, he was responsible for the marketing and promotion of ICL's UNIX System Products.

Prior to this Mr Cunningham worked in the computer industry as a management consultant specialising in product management and systems analysis.

Mr Cunningham holds a Bachelor of Science degree with Honours in Computer Science and Economics from the University of London. In addition, he received an MBA from the London Business School. He is a member of the Institutes of Marketing, Management, and Computer Science, represents UNIX International on the Board of Directors of X/Open Company, Limited, and serves on the Board of Directors for UniForum.

Evi Nemeth

Evi Nemeth is a faculty member in Computer Science at the University of Colorado and has been involved with Unix systems for over 15 years. She and a handful of eager undergraduates have steered the growth of the CS research network from a single Vax 11/780 to its current 100+ machines. Evi is co-author of the best selling "Unix System Administration Handbook" (Prentice-Hall, 1989) and a member of the Usenix Association board of Directors.

Her current interests are in visualization software applied to two domains: data structures and system administration. Her students are building tools to aid beginning C programmers in the data structures course really see what is happening to those pointers. A network manager cockpit tool is in the design stages; it would allow system administrators to monitor machines on the network and quickly see their status and intervene if necessary.

Rob Pike

Rob Pike is a Member of Technical Staff at AT&T Bell Laboratories in Murray Hill, New Jersey, where he has been since 1980, the same year he won the Olympic silver medal in Archery. In 1981 he wrote the first bitmap window system for Unix systems, and has since written nine more. With Bart Locanthi he designed the Blit terminal; with Brian Kernighan he wrote *The Unix Programming Environment*. A shuttle mission nearly launched a gamma-ray telescope he designed. He is a Canadian citizen and has never written a program that uses cursor addressing.

John Totman

John Totman has recently been appointed as Director of User Council Relations responsible for X/Open's dealings and partnership with major users worldwide.

A member of the start-up team for the X/Open company in 1987. John was initially responsible for all X/Open's marketing activity in Europe until the end of 1990. During this period the X/Open specifications were being established as a practical procurement strategy for users wishing to get more long term value from their computer systems.

A key result of his marketing programs has been the formal adoption of X/Open specifications as the basis of open systems procurement by leading Government agencies and ministries across Europe.

Before joining X/Open John had worked on the development of proprietary operating systems, commercial applications and the formulation of corporate strategy for integration of computer networks based on industry standards.

CONFERENCE COCKTAIL RECEPTION

Sponsored by Prime Computer Australia, the Cocktail Reception is the ideal forum to catch up with old friends and make new ones. A time to reflect on the past, and prepare for the future at sunset, against the magnificent backdrop of Sydney.

HARBOUR CRUISE & GAMING NIGHT AFLOAT

Cruise Sydney Harbour, gamble your AUUG currency (given to you) away, bid for prizes, and partake in a gourmet fare, this event will long be remembered and talked about. Invite friends and colleagues to join you on an evening not to be missed. This optional event will be limited to 200 persons, so book early and avoid just hearing about it.

CONFERENCE DINNER

The highlight of the conference, the Conference Dinner, sponsored by Pyramid Technology Corporation promises to be an evening of fun. You will be entertained by Mic Conway's Whoopee Band playing music from the last Great Depression to the next - just to make you happy!. So if you're tired of bland techno-rock which dominates the contemporary music scene, don't fret, THE WHOOPEE BAND is here to freshen up those jaded musical palates with laughter and fun.

EXHIBITION

"So what is an Open System Anyway?" The conference sessions will tell you all about it and you will be able to see the actual technology displayed all under one roof by the leading suppliers to the Open System market. Unix, Pick and many more will be there. Take the time to challenge the various representatives to solve your requirements and to offer you solutions. With over 1850 sqm of displays and 60 exhibitors, AUUG 91 is the most comprehensive display of Open Systems ever offered in Australia.

AUUG'91 PROGRAMME

AUUG' 91 TUTORIALS (Optional Events) - TUESDAY, SEPTEMBER 24TH

0800 - 0900 Registrations

0900 - 1700 **Full day Tutorials**

T1. Overview of Object-Oriented Program Design using C++: David Bern, Software Technology Transfer Ltd.

T2. System Administration of a Local Area Network: Evi Nemeth, University of Colorado

0900 - 1230 **Half day Tutorials**

T3. Portable Unix Programming: Stephen Frede: Softway Pty. Ltd.

T4. Application Development in an Object Oriented and Distributed Environment - a Practical Approach: Charan Lohara, Lohara Software Systems, Inc.

T5. Improving your System's Security: Chris Maltby, Softway Pty. Ltd.

1400 - 1730 **Half day Tutorials**

T6. Expert Systems in an Open Systems Environment: Jason Catlett, Basser Department of Computing

T7. Shell Programming, Stephen Frede, Softway Pty. Ltd.,

CONFERENCE DAY 1 - WEDNESDAY, SEPT 25TH

0800 - 0900 **Registrations**

0900 *Welcome Address*
Pat Duffy, President, AUUG Inc.

Opening Address
Robyn Williams AO., Commissioner, Commission of the Future

1000 - 1045 **MORNING BREAK AND EXHIBITION VIEWING**

1045 - 1115 *The Advanced Computing Environment (ACE) - Finally! A RISC Standard*
David Hancock, SCO Vice President Pacific, Asia & Latin America Division

1115 - 1145 *Broadening the Definition of Open Systems to Meet the Requirements of the Non English Speaking Peoples*
James L. Clark, Ph.D., President and Director, Unix System Laboratories, Pacific

1145 - 1215 *Formal vs. Informal Computer Science Teaching*
Evi Nemeth, Associate Professor of Computer Science at the University of Colorado and Director of USENIX

1215 - 1345 **LUNCH AND EXHIBITION VIEWING**

COMMERCIAL SESSIONS - Concurrent

1345 - 1415 C01. *Unix Comes of Age: The Retail Perspective*
Dr. Desmond Albert, Corporate Management Service, Coles Myer

1415 - 1445 C02. *The Impact of Open Systems*
Richard Cousins, Director, Cousins & Associates

1445 - 1515 C03. *Open System, Open Document*
Dolf Leendert Boek, Manager of Technical Resources, WordPerfect Pacific

1515 - 1545 C04. *Open Systems in the Year 2000*
Colin Kempter, Open Systems Consultant, Prime Computer Australia.

TECHNICAL SESSIONS - Concurrent

1345 - 1415 T01. *Share & Enjoy: Unix System Administration*

ACSnet Survey

1.1 Introduction

ACSnet is a computer network linking many UNIX hosts in Australia. It provides connections over various media and is linked to AARNet, Internet, USENET, CSnet and many other overseas networks. Until the formation of AARNet it was the only such network available in Australia, and is still the only network of its type available to commercial sites within Australia. The software used for these connections is usually either SUN III or SUN IV (or MHSnet). For the purposes of this survey other software such as UUCP or SLIP is also relevant.

At the AUUG Annual General Meeting held in Melbourne on September 27th, the members requested that the AUUG Executive investigate ways of making connection to ACSnet easier, especially for sites currently without connections. This survey is aimed at clearly defining what is available and what is needed.

Replies are invited both from sites requiring connections and sites that are willing to accept connections from new sites. Any other site that has relevant information is also welcome to reply (e.g. a site looking at reducing its distance from the backbone).

Please send replies to:

Mail: Attn: Network Survey
AUUG Inc
P.O. Box 366
Kensington N.S.W. 2033

FAX: (02) 332 4066
E-Mail: auug@atom.lhrl.au.oz

Technical enquiries to:

Frank Crawford (frank@atom.lhrl.oz) (02) 543 9404
or
Scott Merrilees (Sm@bhpes.oz) (049) 40 2132

Thank you

=====

1.2 Contact Details

Name: _____
Address: _____

Phone: _____
Fax: _____
E-Mail: _____

1.3 Site Details

Host Name: _____
Hardware Type: _____
Operating System Version: _____
Location: _____

New Connections

If you require a network connection please complete the following section.

Please circle your choice (circle more than one if appropriate).

- A1. Do you currently have networking software? Yes No
- A2. If **no**, do you require assistance in selecting a package? Yes No
- A3. Are you willing to pay for networking software? Yes No
 If **yes**, approximately how much? _____
- A4. Do you require assistance in setting up your network software? Yes No
- A5. Type of software: SUNIII MHSnet UUCP
 TCP/IP SLIP
 Other (Please specify): _____
- A6. Type of connection: Direct Modem/Dialin Modem/Dialout
 X.25/Dialin X.25/Dialout
 Other (Please specify): _____
- A7. If **modem**, connection type: V21 (300 baud) V23 (1200/75) V22 (1200)
 V22bis (2400) V32 (9600) Trailblazer
 Other (Please specify): _____
- A8. Estimated traffic volume (in KB/day): < 1 1-10 10-100
 (not counting netnews) > 100: estimated volume: _____
- A9. Do you require a news feed? Yes No
 Limited (Please specify): _____
- A10. Any time restrictions on connection? Please specify: _____
- A11. If the connection requires STD charges (or equivalent) is this acceptable? Yes No
- A12. Are you willing to pay for a connection (other than Telecom charges)? Yes No
 If **yes**, approximately how much (please also specify units, e.g. \$X/MB or flat fee)? _____
- A13. Once connected, are you willing to provide additional connections? Yes No
- A14. Additional Comments:

Report on EurOped Spring 1991 Conference

by

Greg Rose

We left Newark airport after a drive through New York's Friday afternoon traffic. It turned out that the 17th of May is Norway's national day, so we were treated to a 50 piece orchestra at the airport, and plied with alcohol on the flight. The plane trip was definitely the best part of that journey, with the drive not measurable on the scale of enjoyable travel.

We arrived in Oslo, Norway, slightly early, just before 8 am. Immigration and customs consisted of a 10 second rubber stamp. The entire delay consisted of the fact that there was only one person on duty for a whole 767 full of people. There was a seven hour wait before the one plane that day bound for Tromso was due to leave. (Tromso is spelled with a workstation I've never seen before, so I'm not going to get fancy. Forgive me in this day of international character sets, but I'll continue to elide the slash.) So we went into Oslo. I'll get another chance at Oslo on the way out; the connection is even worse and we'll be staying overnight.

First impressions were overwhelmingly positive. The place exudes something cheerful, helpful, and sleepy, with the public holiday apparently enhancing the sleepiness. There are ugly buildings, but few of them, and even the ugly ones are not that way because of cheap concrete construction; rather just bad taste I think. But everywhere there are gardens full of flowers just blooming, lovely statues full of humour and life just about everywhere, and I don't know, an air of happiness.

One very common theme in the statuary is mother (or parents) and child (or children). There must be something that makes this more important to the Norwegians than any other culture I know. My current guess is that it must have been hard to survive long harsh winters here, and children must be precious.

We took a random ferry trip through Oslo's harbour and got off somewhere else for a walk. This was not an obvious tourist place, and I think we saw some real suburban life there. Undeniably a pleasant place, at least in late spring. There were also sandboxes all over the place, so you

know there is a snow problem during winter.

Back on a plane to get to Tromso, almost two hours due north of Oslo. We couldn't see the lovely crinkly edges, due to a pretty solid overcast. But once we started the descent to Tromso the scenery picked up. The plane had to fly up the fjord past the city, then do a medium tight turn and come back down to land. I don't know how they would get here in instrument conditions; we're surrounded by 2000m peaks.

Tromso is spectacular, as much so as I had been led to believe. It is on a sheltered island in the middle of the fjord. This is a strange fjord, too, actually being a channel between the mainland to the east and some much larger and almost joined islands to the west. This is why Tromso exists, I think; it is the only land bridge to these islands with their important fishing industry. Speaking of which, the seafood is wonderful, and I don't even like seafood. There is a university here, and lots of sunlight. Tomorrow is the first day this year when the sun will fail to drop behind the northern horizon. Even so, the nights have not been dark. You can tell night by the fact that the street lights go on for an hour or so.

Saturday night was uneventful. Went for a walk, met a few people, and I fell asleep over dinner. Sunday I went for a longer walk, helped stick labels on badges (withdrawal symptoms from AUUG I think) and otherwise whiled away the day. Today is Monday, the terminal room is set up and apparently working, and I'm typing my EurOpen report for anyone wanting to read it.

More later.

Afternoon, 20th. (Monday). Went to the Tromso Museum, and it was very good. We learned a lot about the Vikings and Sami (Lapp) people. The Viking explorations were unbelievable. They almost made it to the Indian and Pacific oceans, and sometimes went overland to reach a river flowing in the opposite direction. Anyway, little to add at this point.

Tuesday 21st. Today a tour went to the Linge Peninsula, about 80 km northeast of Tromso.

There are some spectacular mountains here, and I saw my first glacier. The tour took us to see a typical Sami (Lapp is considered to be a derogatory term) home. The Sami were the original settlers of this part of the world, and live in conical tents. During the winter months they weave very elaborate designs from coloured wool, dyed with extremely bright colours made from lichen. They also have beaten metal jewelery. All of this seems very incongruous, as they are pale and well dressed, one doesn't consider them primitive. Their life is definitely so, though. In the summer months they come to the coast where it is cooler, so that their reindeer can breed. In winter they head back inland, traditionally on reindeer powered sleds, but these days using cars, motor cycles and snowsleds.

Part of the tour included genuine Sami food. It started with glacial melt water, pretty tasteless, beer and coke (I didn't say the drink was genuine) and they passed around some powdered reindeer horn, a noted aphrodesiac. Dave Presotto from Bell Labs went wild with this stuff. Then we had reindeer stew, or stroganoff, or something. It is a dark meat with a nice gamy flavour. There was a sort of a jam made from local berries with an unpronounceable name. And coffee.

We even managed to see a real reindeer in the wild, but we didn't eat it. The Sami life basically revolves around the reindeer, as you may have gathered.

One of the impressions I got, and I imagine it came through above, was of a happy and prosperous community. Well, reality was of course brought home today. The area used to survive on the fishing industry, but there isn't one any more. Tromso itself is not suffering, since it was never really a fishing town. It now supports the administration for the north of Norway, the navy submarine base (oops, we're not supposed to know about that), the university, and airport. Everywhere else there is unemployment. The place is basically fished out. They hope fish will come back soon and this time they will be more responsible. This is sobering; I really thought that the place was not particularly spoiled, even after 9000 years that the Sami have been here.

Wednesday the 22nd. Today is the first day of the conference proper, so I've stopped mucking around and gone to some of the talks. Anyway, it is cold and drizzling outside. If I haven't yet

mentioned that the venue is nice, I should have.

The keynote talk was Michael Schroeder from DEC's System Research Labs in Palo Alto. He presented a well reasoned overview of why the current kinds of distributed systems are still not completely solving the problems that exist in the real world, and that we've lost some of the desirable attributes of the old timesharing systems. In the main, I agreed with most of what he was saying, but in the detail there were a number of places where I disagreed (not that that matters; after all that's why we come to conferences). At one point Michael said that the Internet was fault tolerant, in that it would eventually route around disabled segments. Rob Pike, sotto voce, said "The only way the internet is fault tolerant is that the users tolerate its faults!"

The terminal room here is an excellent job by the way. There are about 15 HP 9000 workstations, and eight vt220s. I sent a piece of mail to Peter Barnes in Queensland and was stunned to get a reply in about one minute. This was not a bounce back or anything, but a real human intervention reply! After 5 months I was stunned by this. I'd forgotten what it could be like.

After coffee break, we had "Experiences with Amoeba", by Sape Mullender, currently at the University of Twente in the Netherlands, but formerly involved in the development at the Vrije University. This was the first coherent overview of Amoeba I have been privileged to hear, and was a good presentation of a powerful system which has broken a lot of ground. We have missed out in Australia on a number of developments like Amoeba and Chorus (see below) and should rectify this.

Michel Gien, chairman of EurOpen and a founder of Chorus Systemes, then gave a talk about Chorus entitled "A new look at Microkernel Based UNIX Operating Systems: lessons in performance and compatibility". In reality this was a quick overview of the design of Chorus and its Unix compatibility features, with a historical perspective on why some of the decisions were made this way. Michel has a hard time justifying why his micro-kernel is bigger than Plan 9 (see below) and why they are shipping bits of it back into the kernel "without losing modularity" he stresses. This was a good talk too, although Michel's accent was the hardest to understand so far (for me).

Lunch was a seafood and various nice stuff plate, with a slab of Pork added hot, and a chocolate mousse in a chocolate cup. Very nice. It was served with wine, which I still feel is a mistake, as I saw a few people nodding afterward.

One of the terms I used above describing Norway was "sleepy". I think this impression was because the people were sleeping off the effects of the 17th of May. Today it is bustling and alive, it just adds to the appeal of the country.

The next talk was Simon Patience from the OSF in France, titled simply "OSF/1". His opening slide showed that OSF/1 is based on 4.4BSD, which was news to me. 4.4 is gaining in importance. Simon was a good speaker and made the best of what was the nearest thing to a marketing talk yet. I'm not allowed to say that OSF/1 is too complicated by an order of magnitude, even if I wanted to say such a thing, which I don't. It is, after all, smaller than (you guessed it, see below. It isn't my fault the talks were given in reverse order.)

Dave Presotto spoke at 38.4 kilobaud amongst a room full of framing errors. About half way one of the audience managed to get a word in to ask him to slow down. He said he already was. His slides were mostly hand written, and he was the first speaker not wearing a tie. We in Australia have heard a fair bit about Plan 9, so I won't belabour it. Not too bad a talk.

Afternoon tea, and my first chance to glance at the exhibition. The conference and exhibition were both small by Australian standards, with about 300 people attending (that isn't too bad) but only about 10 stands in the exhibition, and no exhibition-only walk ins. H-P have the biggest booth, and having also helped enormously with the terminal room, have a basically captive market here in Tromsø. You could really. They have their new 55 Meaningless machine on display but I haven't played with it yet.

Across the aisle are IBM's local representative, with a bunch of workstations and a system 9000. This is attached to a 3380 disk drive (ahem, DASD) that when opened up could be mistaken for a tractor tyre. The point of the display is that you can run applications on the 9000, and using ANDF take the binary and run it on a workstation. There were no RS6000/550s here, since the salesman says they are selling too fast to keep any for demonstrations. I would have put this down to

marketing hype if I wasn't the recipient of the only ones that made it to the research lab before the supply was cut off. (I've been promoted by the way.)

Upstairs, Digital are giving away bright red suspenders with OSF/1 and Ultrix discretely written on them. Haven't played with their machines either, yet.

The other displays are two or three local software companies or VARs, and two publishers. I've pretty much seen it.

Since the exhibitors (with the exception of one of the VARs who sold Suns) were universally OSF members, and the next talk was titled "Distributed Computing in System V: Today and Tomorrow", by Andrew Schuelke of UNIX International in Belgium, I reasoned it was probably a marketing presentation and accidentally failed to attend while I was typing this. Apparently I was right. No disrespect was intended, but I had heard this one (or something like it) before.

There's a panel on now, so I'll go to it, and then get drunk on the Fjord Cruise tonight (also sponsored by H-P, who also supplied a small but tasteful conference backpack).

Thursday 23rd 11:16. Well, I went back down to the forum, or panel session, or whatever you call it, and it was fun. It consisted of all of the speakers from yesterday, chaired by the first one, Mike Schroeder. The program chair was extremely upset with both of the talks from UI and OSF; the one I missed was a blatant sales talk (apparently) and spillover into the forum took a lot of time. Most of this consisted of the UI and OSF representatives snatching the microphone off each other in their haste to say that they were cooperating on technical issues and were not at war. I can't remember most of the discussion but you've heard it all before.

There were some bemused looks when H. Strach-Zimmerman, who I am told is a founder of X-Open, vehemently (and seriously) accused Dave Presotto of trying to ruin European research into operating systems by giving away Plan 9, then doing a bait-and-switch like with the TLI stuff that started the above wars. This stimulated some discussion.

The menu for the boat trip last night turned out to be prawns and beer, both of which I dislike (I know I'm un-Aussie in many respects), and

combined with the around-zero temperatures, wind, and snow I decided on a pizza. Tromso (the correct pronunciation by the way, translated into Strine, is a bit like Trrrrooomser, given that Australians never pronounce trailing 'r's but use them to modify the vowel sound. Unlike almost all the furriners here, I get the vowels vaguely correct but cannot do the rolling cosmopolitan mixture of foods, after all it is called "the Paris of the North" with a straight face by many people. The local food consists of seafood and the occasional reindeer, with some lamb, mutton and beef. But it is easier to find other cultures' restaurants.

After boozing fairly seriously until about 1am, I managed to get up and go to the first talk this morning, "Open systems Distributed Computing and interoperability Fact and Fancy", by Bruce Shriver. Bruce is a professor on sabbatical at D.H. Brown Associates, and president elect of the IEEE Computer Society. He was an enthusiastic speaker who (in my opinion) talked down to the audience somewhat in the guise of defining terms. As has been made evident here, the locals are not dumb people, and anyone else who would come this far is pretty serious about their field (or wanted a good perk like me, but they are in the minority). I skipped the next session on Architecture to write this and prepare my talk.

Friday the 24th. This might be the last entry before the terminal room gets packed up. The afternoon talks yesterday were all right, but not fantastic. There were two exceptions; Rob Pike gave his talk entitled "Process Sleep and Wakeup on a Shared Memory Multiprocessor", which was all about 8.5, his windowing system for Plan 9. There are two interesting things about it that are relatively new: it powers up in less than a second (unlike the measured 45 seconds from logging in to getting a chance to type in the email room here), and you can run an X server in a window on it. The other talk I found interesting was "The Zaphod Cluster File System", but that was because I had a vested interest.

I actually spent most of the afternoon running around doing the famous dc benchmark. There are a number of new machines here and there were some interesting results, if you keep in mind that such a benchmark measures nothing of importance.

Neither IBM nor HP, the current bang-for-buck contenders, had their fastest machines here. The "Snake", the HP 700 machine, is a definite winner for the moment. There are drastic differences in the levels of performance depending on different benchmarks. The dc benchmark shows the HP 9000/720 and IBM RS6000/530 pretty close on performance, while other figures measured at CERN and reported here show the former to be faster than an RS6000/550, which is twice the clock speed of the /530. This discrepancy is hard to credit, I'm just reporting it.

IBM had a baby mainframe here, attached to the disk drive mentioned above. It was an ES9000/150, and was three times faster than anything else in sight. They didn't want to let me run the dc test (in fact they had dragged Rob Pike away from the terminal previously), but they let me when they saw my badge. There was some trouble with the terminal connection, as there was no network interface to the machine (which rather destroyed the point of the interoperability demonstration) and we couldn't find a `` key on the Norwegian keyboard. I've seen inside dc, and so had no hesitation in running an equivalent dc script to get the time on this machine.

There were three machines running OSF/1 here. They were A DECStation 3100, an HP9000/425T, and the ES/9000. All were alpha test versions. I'm only reporting the DEC comparison figures, as the HP ones were aberrations of some kind, and there was no way to compare the ES/9000 with AIX/370 on the same machine. OSF/1 was slightly slower than Ultrix, on the same platform, but one presumes that is because of it being an alpha test version.

A number of the machines wouldn't run the script as Piers had sent it to me. "time dc" had to be rewritten as "/bin/time dc", or the numbers weren't reported (except on the ES9000, where it was /usr/bin/time). (Disclaimers first: I did this as me, generic bothersome person, not on behalf of or as a representative of IBM Corp.)

What Piers told me to run was:

```
echo "99k2vdsap8op19^pla/pq" | time dc >/dev/null
```

Except for giving a path for time on a number of machines, the only change I needed to make was for the ES/9000, where it actually read:

```
echo "99k2vdsap8opdd*dd*d*d***pla/pq"|time dc >/dev/null
```

This gave the same answers, and I assert that it is equivalent (actually slower, because it has to do

more parsing, but that is trivial compared to the computation). Of course whenever I make a statement like that above, and I'm sitting at a keyboard, I attempt to prove it. So I just ran both the tests on this machine, and discovered a factor of about 6 difference between the two. It seems the divisions by two used to do the binary multiplication expansions are actually dominating the exponential, and I was doing them in my head as I typed (or some other reason if my understanding of dc is incorrect). Oh well, the ES/9000 is summarily dropped from the figures in table 1.

I say again, ignore the real times on these tests. All the machines may have had background stuff happening.

Later, HP really turned on the marketing charm, and turned off the many friends they had previously made. Then it was the Conference Dinner. This was three courses, a large sort or biscuit bready thing covered with a sort of caviar thing, followed by Reindeer Roast (bloody reindeer; the locals assure me they never eat it) which was very nice, and a strawberry cheesecake. A Chilean Cabernet was served that was quite acceptable. This was a tame affair by Australian standards, less than 3 assaults and a repair bill not exceeding 1 000 000 Kroner.

It was hard at the dinner to socialise with anyone but the people at your own table, but that was corrected at the bar afterwards.

This morning I've spent typing up the benchmark results, and arranging to go buzzing the fjords this afternoon. The sun is out and it is a nice day, after some real snow yesterday.

Tuesday 28th, back at work. I heard most of the talks on Friday, but it's hard to remember them well enough to write them up. It is worth mentioning that the technical program was excellent with only a couple of exceptions, and even if it had not been in Norway the conference would have been good value.

The Fjord flying was unbelievable, simply magnificent. They should have scenic flights available to tourists; they'd make a fortune. There were two pilots at the conference who cared enough to go, myself and Berry Kercheval. We hunted up an instructor, Stine Droge (I've probably misspelled his name, as I've never seen it written) who flies a Citation jet air ambulance

for a living. We filled the remaining seat with a colleague from IBM in Europe, Jan-Simon Pendry. I started off flying from the airport west over the island of Tromsø, then over the nearer mountains, across a fjord and into the Lingen Alps, climbing all the way. Then we turned south, across a few glaciers and into the area of avalanches, past the highest mountain thereabouts, Jiek-Kevarri, 6013 feet (off an aviation map), then turned east and descended into Ullsfjorden, the fjord we had previously crossed, north in the fjord to make a figure of eight, and then south along Tromseundet back to land. Berry took the controls and we went east this time, to the coastal islands and back again. The scenery is nothing but spectacular, and Berry and I spent the next few hours telling the earthbound mortals how much we pitied them for missing it. Stine enjoyed the flight so much he refused to charge us for his time.

Friday night after a Balkan dinner we walked to the frozen lake at the top of the island in the hope that the sun would break through and we'd see it at midnight, but no luck ensued. It was a nice walk though, and there were a lot of birds at the lake.

Saturday we flew back to Oslo, and visited the Maritime museum complex, had a nice dinner, and collapsed. Flew back on Sunday.

I'm in love with Norway, its people, its scenery, its atmosphere, (not its prices,) and I'm going to go back, somehow, someday.

TABLE 1

REAL	USR	SYS	System, notes etc.
(7.2)	5.6	0.1	DECSERVER 5000/200, MIPS R3000@25MHz, ULTRIX 4.1 (not idle)
(38.3)	12.3	0.7	Decstation 3100, MIPS R2000, OSF/1 Alpha test version, (downright busy, ignore real time)
(23.1)	10.1	4.3	Decstation 3100, MIPS R2000, Ultrix 4.1 (busy, sys time might be an abberation)
11.9	7.4	0.9	Sun SparcStation II, SunOS 4.11
4.2	3.5	0.6	IBM RS6000/530, AIX 3.1.x (not current OS)
9.84	7.24	0.2	HP9000/425T, 68040@25MHz, HP-UX 7.05
6.49	5.43	0.06	HP9000/720 PA1.1@50MHz, HP-UX 8.01

Task Tracking Systems- Watch Dog And Friend

Jack Dikian

Media-Lab Pacific

**P.O. Box 220 Neutral Bay Junction
N.S.W 2086
Australia**

ABSTRACT

Among the many services a typical Support Centre provides to its client base, perhaps the ability to maintain control over the many unresolved or pending tasks is the most important. A typical Support Centre facilitating a medium sized computer supplier can handle hundreds of queries a day. The type and complexity of the queries can vary greatly.

As platforms become more and more compatible, not only does the variety of tools and packages increase, but the source of those tools and packages become increasingly diverse. Good examples of this are apparent in the Unix and DOS environments. These days, both environments promote a great degree of compatibility, exhibit a large variety of readily available software solutions, as well as fostering the proliferation of independent software developers competing to sell the perfect accounting system, the fastest DBMS, and even the trickiest adventure game. All of this can of course be contrasted against a less open system such as the proprietary operating systems. Here, major tools and packages are often written in close collaboration with the original platform manufacture.

Providing support in an open environment may in some cases require a Support Centre to be many things to many people. A well thought out support strategy aided by a comprehensive task tracking system can make the difference between an effective support unit and one that is pure hindrance to all parties concerned. This paper presents what I consider important when setting up a formal Support Centre in an open environment, as well providing some design ideas for a productive task tracking system.

1. Introduction

A Support Centre is usually that part of the overall EDP effort that attempts to answer queries and/or coordinates the resolution of problems that clients experience while using a particular system. The support effort, philosophy, and practice can span between two functionally varying extremes. At one end of this spectrum, the Support Centre and its personnel can be extremely specialised and thus attempt to provide the same sort of client support as that expected directly from back room technicians. On the other hand, the Support Centre may be setup such that it caters for a much broader, and hence more general client base. At Media-Lab Pacific a less technical support team is supported by a more rigorous system of call recording,

problem escalation, task distribution, and follow up strategy.

The factors which often determine the type of support that will be provided out of a formal Support Centre will include the following considerations:

1. The variation products to be supported.
2. The stability of the target environment.
3. The standard of service expected from the support team.
4. The size of the EDP team from which support is provided.

2. The Degree of Product Variety

Open environments such as Unix cater for, and encourage the proliferation of standard non-proprietary technologies. The number and variety of available packages and utilities can be astronomical. My latest copy of Sun Microsystems' Catalyst (catalogue of third party hardware/software SPARCware solutions) lists more than 2000 products in 22 very broad categories. Silicon Graphics boasts a hefty 700 page Applications Directory. At a different level, most Unix systems come to us with at least three file backup mechanisms, two editors, two command line interpreters, and a host of vendor added features. One particular manufacturer even provides two flavours of Unix running on the same machine at the same time. Although much of this type of variety comes about from a genuine need for a particular feature, there is nevertheless sufficient overlap and a great deal of inherent flexibility to allow the client to use any utility based almost entirely on preference alone. Back at home, things are not too much easier either. The PC industry has never been in a stronger position, providing great performance for value, a host of industry standard plug ins, migration paths, and a world of software.

The wide range of products are often indicative of the large number of independent vendors, all committed to making their goods available on these platforms. Whilst a suite of products written for one proprietary environment may feature common menu structures, file layouts, control files, naming conventions, and even consistent documentation, a series of products supplied by different vendors in an open environment will often possess no underlying uniformity. Contrast for example DEC's VMS operating system against Unix. VMS, like Unix parades a large number of layered products. However, unlike Unix, layered products under VMS all exhibit the same underlying design philosophy. Control keys always represent similar actions, help files always display similar layouts, and error messages always have the same consistent format.

Each product in an open environment will require the same high level of technical training before it can be properly supported. In such an environment, a generalised type Support Centre is most effective. Personnel in this centre may have an overall systems, analytical and communication skills, but may lack the specific specialised product knowledge. There are simply too many products for any one individual to know well. In such an environment, the Support Centre acts as an interface between the client and other more specialist EDP groups. How this interfaces is likely to work is closely linked with factors contributing to the overall style of Support Centre. For example, the overall size of the EDP group, the size of the client base, agreed turnaround commitments, and the sophistication of task tracking system will often determine the interface mechanism.

3. The Stability of the Target Environment

The stability of the typical environment that requires support is a very important factor in determining the type of support expected from a Support Centre. Site instability may be inherent within a certain client group or may transcend them. In an environment which has just gone through a major systems conversion, upgrade or simply taken on new responsibilities, it is typical that that site will experience a period of instability. On the other hand, a software house involved in developing systems around products that you are supporting will seem in a sense always unstable. In the later case, queries will often be raised reflecting the need for more detailed specifications, reporting faults which may or may not be associated with your product, and importantly, the reporting of legitimate, but low level anomalies which get picked up due to

the nature of this sort of work.

Both of these scenarios need to be addressed. These support requirements will inevitably exist in both open and proprietary environments. However, in an open environment, not only are there many more independent groups developing systems, but often, these groups will know as much about your product as your Support Centre personnel. This is especially true if the Support Centre is of a generalised type.

Clients experiencing significant downtime due to various reasons including conversion processes, come closest to justifying the need to contact back room staff directly. It is also this very desire that raises the many concerns for the quality of support. Clients more often than not, always prefer to discuss their problems with back room personnel directly. However, there are just as many important arguments for clients to continue to address their problems through a centralised Support Centre. Many of these arguments are to do with a supplier providing a consistent level of support, as opposed to a see-sawing effort depending on the availability of key personnel (See figure 1). Figure 1 illustrates how the quality of support (*) can vary between two great extremes when the Support Centre is basing their support effort on key back room personnel. When these people are available, the support effort is at its peak. However, when they are unavailable, support effort is reduced to a minimum. The use of a more generalised support team (-) allows the Support team to provide a much more consistent support effort. It is neither as good as the back room personnel, nor as bad when they are not available. In time, this quality will rise. Providing consistent support, with an agreed service level come about most effectively through the implementation of service level agreements between yourself and the client base. The section discussing the standard of service expectation will further detail the implementation of service level agreements.

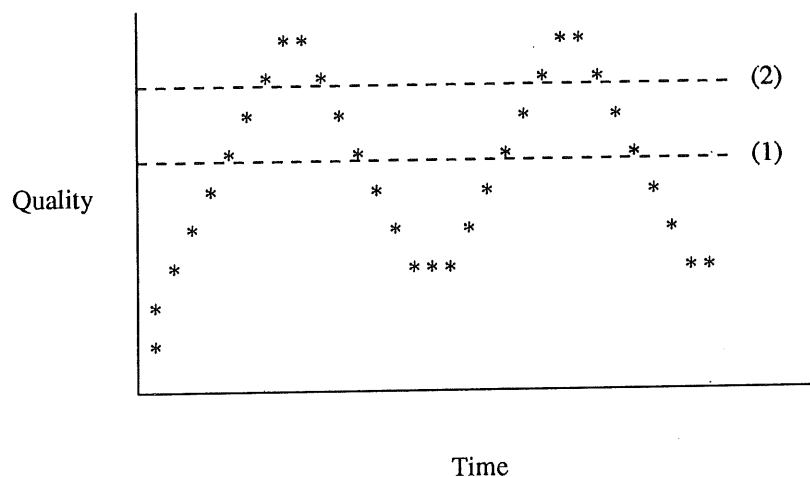


Figure 1. Quality of support.

In the situation where a client is going through a major conversion process, or where a major upgrade is to be released, the generalised Support Centre should be supplemented by back room technicians rather than be abandoned. The client should be encouraged to schedule the conversion process, as well as assisted in associated areas such as risk analysis, resource allocation, and contingency planning. Both the client and the supplier will only then really appreciate each others requirements. This liaison process will allow the Support Centre to prepare more adequately for this situation. Without this sort of preparation, the support personnel are, at best, most likely to allocate the highest possible priority available to them before escalating the query to a pool of back room personnel. The problem with this approach is of course that there may well be any number of such high priority requests already outstanding.

Some suppliers nominate certain individuals to be responsible for the support of one or more clients. Account managers are usually very effective in providing a personalised style of support. They can become acquainted with the specific needs of the client, as well as develop an understanding for the longer term directions of the organisation. These people also represent the supplier and its resources. The negative

aspects of this approach however are significant in an open environment. The main problem is once again availability. The greater the dependence on the one account representative, the more exposed both parties become. Account managers also tend to have strong consulting, negotiating, or managerial skills. These qualities are very important when dealing with sites that feel undersold. At the end of the day however, hands on representation will still be required. Problems need not only be understood, but fixed in a timely manner.

4. The Standard of Service Expected from the Support Team

Often, the effectiveness of a Support Centre is erroneously measured by the quality of on the spot advice the support personnel can provide. For example, there is always a perception that a particular organisation has a good support operation if clients constantly receive informative advice immediately. On the other hand, giving the client a reference number with the promise that an expert will attend to their call as soon as possible raises, in some peoples minds, a certain amount of cynicism. There is simply no formal framework by which to gauge the effectiveness of the support in non-subjective terms. Questions like "Why can't we talk to this expert direct...?", "why do we need to explain this problem all over again?", and so on are asked, and asked of the wrong people. This concern is accentuated when the overhead costs of maintaining the support effort are passed on to the client base. Soon, some clients will attempt to contact back room personnel directly, avoiding the so called front line. Clients who judge a Support Centre ineffective, fairly or otherwise will always attempt to bypass formal support structures. In general, clients believe that back room personnel are the ones who really understand the problem; they know the system and have the facilities to fix it. Everyone else, including the Support Centre are nothing more than "go betweens." To some extent, there is a lot of truth in this belief. Support consultants, specially in a help desk centre do not have the luxury to see the problem for themselves. They can only react to what the client thinks is seeing. Often, if the Support Centre can't resolve a problem over the phone, they are forced to escalate the task to someone who can. Back room personnel may be so familiar with particular anomalies that they will save the client the bother of describing peripheral, and in retrospect, unnecessary detail. Also, where a support consultant may tackle a new query by first examining a wide range of possible causes, a back room techo, faced with the same task may grasp the kernel of the problem much quicker.

The other side of the coin holds a different picture. It is also true that most back room guys do not want to be interrupted by users. Back room personnel can handle direct contacts in various, and in some cases undesirable ways. Perhaps the three most common reactions I have seen are:

- They may refer the problem back to the Support Centre and hang up in the clients ear. This, as it turns out is probably the best course of action in the long run.
- Secondly, they may promise the client prompt resolution knowing full well that they will neither have the time nor the inclination to carry it through.
- Finally, they may well resolve the problem in a timely manner, but fail to inform the client, or their peers of the solution.

Back room personnel are never hired based on their diplomacy, communication, or even business skills, most of these people will call a spade a spade. This is regardless of who the caller may be and the sort of effort the sales team expended to secure them.

Back room personnel, the development team, systems managers, operators, DBA's *etc* are highly paid professionals who, without stating the obvious, have a job to do. It's simply not as if they are waiting around idle for someone to ring them with a problem. Often they are working against tight schedules, with priorities not including impromptu client requests. Often, what starts off as a very brief interruption may eventually result in a great deal of wasted time and resources. In the event that a problem has been accepted directly in such an environment, there is no real guarantee for effective follow up, no framework for providing a realistic turnaround time, or even a rudimentary concept of ownership.

So what is a fair expectation of a Support Centre, and what can a supplier do in terms of its support policy? The best way to avoid unreasonable client expectation is to tell them exactly what you will, and will not do. If the Support Centre is manned between the hours of 8:30am and 6:00pm, then the client should be made

very aware that support is not available outside these hours. If the Support Centre is promising a maximum turnaround time of 3 hours for critical problems, then clients should be sold the support service based on this condition. With the same token, published support hours are required to be adhered to by the support personnel. A support facility between 8:30am and 6:00pm means just that, and does not mean 8:45am to 6:15pm. Also, a promise of a 3 hour turnaround means that if an answer is not found within that time frame, then the client is informed, and a suitable alternative arrangement is made.

The creation and regular maintenance of a standard service level agreement is not only a very effective method of making your position clear in terms of the support delivery, but it is also a vehicle by which the client can use to request certain specific functions. The implications of the service level agreement will drive the practice and policy of the support personnel. Once the Support Centre is comfortable with this agreement, every new client likely to use your support facility should understand and accept the terms and conditions of the agreement. In the event that a particular client has specific requirements outside the scope of the standard service level agreement, then a new agreement should be forged before formal support commences.

The new service level should be such that both parties feel comfortable with. It is no use for example the client requesting call back on reported faults within an hour when the Support Centre averages 2 day turnaround time. Ultimately, an attitude where the user pays should be employed. If a client has a need for 24 hour support, and your Support Centre only provides an 18 hour window, then the idea that the client pay for facilitating the expansion (should you agree) of the support window should not be discarded lightly. It is also important that the service level agreement is structured such that it complements the original maintenance or service warranty contract.

The sort of elements in a typical service level agreement should include:

- Support Centre hours.
- Method of priority allocation.
- The meaning of priorities and corresponding actions.
- Escalation mechanism, and responsibility lines.
- Forms of expected documentation.
- Emergency arrangements.
- Turnaround times for various types of situations.
- Contact name nomination guidelines.
- Support Centre responsibilities.
- Problem status reporting procedures.
- Charging method.

The presence of such a document takes away any misconceptions of the Support Centre's role. It clearly identifies what the Support Centre will do, and how it will do it. Importantly, clients can compare how the Support Centre is performing against how they have formally committed to perform. Anomalies can be raised and resolution sought at the appropriate business level. This type of structure enforces accountability at all levels. What this understanding also does is throw back some responsibility on to the user of the Support Centre. In the same way that the client can highlight difficulties with the Support Centre, the Support Centre itself can rightly object to providing a support facility if the client is not upholding their end of the bargain. For example, if the client has been asked to document a particular problem in accordance with the service level procedures, and have failed to do so, then the Support Centre can legitimately hold that query pending until the relevant documentation is supplied.

5. The Size of the EDP Team from which Support is Provided

In a very small EDP team comprising say of 15 members or less, it can be argued that there isn't a large enough infrastructure to support a dedicated support group. However, as the EDP group grows, and the overall EDP effort divides into specific speciality areas such as R&D, business consulting, operations, and systems delivery, it becomes more and more apparent that a support type team is necessary. In some cases, this support team is a misnomer, and what the group really needs is an office administrator/telephonist/goofier. However, as the client base grows and their requirements become more sophisticated, a single contact point within the EDP group becomes unavoidable. This is the support team. Not only does the support team provide fault diagnostics and rectification, but also a host of many other functions. For example, the support team can coordinate and follow up the resolution of queries within the EDP group. The Support Centre can act as an interface between the client and the supplier, as well as interface between the many sub groups within the EDP group (See Figure 2). Problem prioritisation, logging, and reporting is also something typically handled without a Support Centre. These functions help glue the overall EDP effort as well as streamline the many client requests. Importantly they free the other groups from continual client interruptions, and channel the filtered requests to the most appropriate people. The client on the other hand sees a single consistent face who is prepared to accept ownership of the problem until it is resolved.

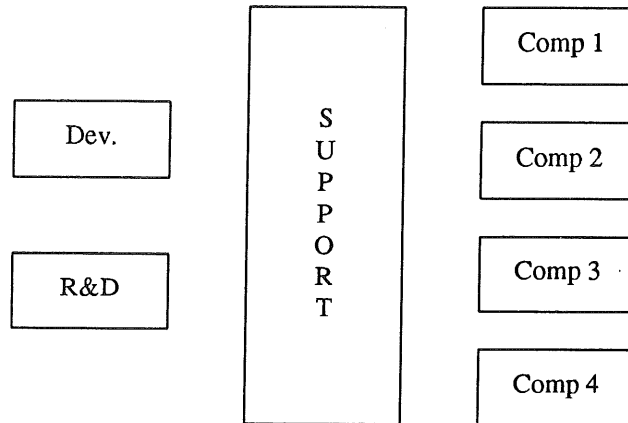


Figure 2. Location of Support Centre within Client/EDP group.

In order for the Support Centre to maintain control over the many unresolved tasks, and also provide efficient call reporting and statistics facilities, a sophisticated task tracking or logging system is indispensable.

The task tracking system should not be confused with a general task scheduling, or even an accounting system. Although both a scheduling and account charging system can be implemented underneath the task tracking system, in general, these modules should exist independently. The task tracking system should however possess the following features:

- A multiuser system with the ability to differentiate between user groups in order to provide in context displays and data update restrictions.
- An inter/intra office E-mail and fax interface system.
- The ability to accept task priority allocations, and prompt the user with the appropriate action based on a setup database.
- The ability to accept and or automatically allocate expected completion times for tasks based on a number of criteria such as priority, client expectation, task type and client status.
- The ability to accept historical notes and provide online query reporting based on client, task type, task status or chronological order.

- The overall system should be fast enough to service a real interactive session. Typically, queries and actions should be generated at speeds that match the telephone conversation.
- The system should be sitting on top of a user maintained database that provides such things as valid contact names, site information, product information, release compatibility, release schedules, inventory control, technical notes and online manuals.
- An escalation system should exist that is based on task urgency, potential client exposure, allocated priority, and completion time blow outs. The escalation mechanism should be such that it automatically raises the priority allocation, warns the support team, and delivers the appropriate messaging to the support supervisor.
- The system should allow the creation of *ad hoc* query and reporting facilities by the user. This is opposed to reports that are run periodically. Periodic report should provide both the user and the client base with various statistical, and managerial information. This should include outstanding open queries, queries of a certain type, queries logged within a period of time, query history and other charge information.
- Tasks should be allocated to individuals and or groups, with the guarantee that all unresolved tasks have a single owner. The owner may change during the resolution cycle, but it must always have an owner. When tasks are reallocated, the person receiving the task should be made aware via a formal and agreed mechanism.
- A facility should exist to reopen closed calls, or alternatively, clone previous calls for subsequent alteration. This allows the support consultant to quickly log calls based on previous facts.
- The system should allocate unique task reference numbers that can be used by both the client and within the EDP group.
- A number of task status codes should be catered for to reflect the position any one task is at. For example, a task may be pending, closed, completed or current.
- Action codes should also be employed along with a more descriptive comment to indicate the sort of work that has, or will take place for any particular query. For example, the support team may indicate that the query has been transferred to development, the development team may indicate that they are testing *etc.*
- A system of query categorising should be catered for such that each query can be placed in a particular class of problem. For example, the query may pertain to network errors, inoperative terminals, application errors, systems errors *etc.* This will assist in reconciling the problem areas and thus put in place longer term solutions rather than simply fixing the problem at hand.

It should also be said that even the most sophisticated task tracking systems become ineffectual when either the system is not accepted by the users, or when there isn't enough motivation to use them. The system has to be used by all parties concerned, or none. Once a commitment is made to use such a system, management should ensure that all members comply. Equally, nagging difficulties identified by the users should not be discarded. The biggest problem with getting such systems accepted by users is the obvious fact that it might take a few minutes to resolve a problem, but an equal amount of time to register it. It should be noted that much of the benefits of these systems are long term ones and therefore may not always be appreciated in the short term.

The system should also be used in an on-line manner and queries logged into the system as they are raised. An off-line system will result in catch up games played at the end of the day in order to register the days queries. This has two main disadvantages. The first is that retrospective entries will be rushed and can be erroneous. The second is the very real possibility that certain requests do not get processed until the task is recorded. This particular problem is especially ominous when the service level agreement promises a few hour turnaround time for very high priority tasks.

6. Conclusion

The two most overwhelming factors which go hand in hand when an organization is looking at providing an effective support effort is the introduction of a well thought out service level agreement, and the implementation of a task tracking system. The scope of the service level agreement, and the functionality of the tracking system can evolve over a period of time, however, it is extremely important that a certain amount of initial planning takes place. It is important for example that prospective clients understand, and accept the manner in which support will be delivered. New services and conditions can be amended in time, however, the basic principles of working within an agreed frame work, and providing a measurable standard of service should be a major initial goal. The task tracking system can also start small. The first version may be nothing more than a hand full of shell scripts that allow call recording, searching, and reporting. Once again, the system should be designed such that it can grow with the users needs. It should be a system that the user needs and drives unlike many large systems available these days which force the user to change there working methods.

What's happening with ISODE, the free implementation of OSI?

by

Andrew Worsley

worsley@mel.dit.csiro.au

CSIRO Division of Information Technology

Abstract

In this paper I describe what has happened since the talk I gave at AUUG 1988. In particular how the work at DIT† to develop *pepsy*, a table driven ASN.1 compiler system, has been incorporated into ISODE. The paper provides details of another development *PP*, a free implementation of X.400 which is now being distributed. The latest work we are carrying out at DIT in a combined development of an User Agent using X windows to provide the new services of X.400 mail standards to be used. These new services include as sound and picture components of messages and eventually more sophisticated such as encryption, authentication, non repudiation of delivery and delivery reports initiated when the mail is read by the recipient, not just delivered.

1. What is ISODE and what does it provide?

OSI stands for Open Systems Interconnection,¹ and is the Internationally standardised method of exchanging data between computer systems. The standards are produced by ISO, the International Standards Organisation of the United Nations, in cooperation with CCITT, the standardising body of Telecommunications Providers. The goal is for all computers to use these standards as the basis for communication between computers and so any computer will be able to communicate with any other computer, regardless of its manufacturer or operating system.

At the moment these protocols are only sparingly used and don't provide much advantage over existing systems. The complexity of the protocols and so the large size of the implementations has also caused a reluctance of many technical people to adopt these systems, not always with out reason. By freely providing the OSI technology these problems are being attacked in two ways. Firstly, by implementing OSI in a freely available manner the implementors and users can uncover faults and feed back corrections into the standards procedure. Secondly, by making it widely available it can easily be used by companies into their products and in cases where the OSI system has advantages over existing systems it will spread rapidly.

ISODE² is available under even less strict terms than the GNU software. It's free except for a distribution cost. You can use the software for any purpose, modify it and sell or what ever with no charges or obligations except that the writers of ISODE are not to be held responsible for any failures of the software (a restriction that is probably very sensible in the litigious USA).

ISODE has become a major force in OSI implementations. Groups like NIST looking at using it as a standard implementation against which to measure other implementations of the standards. Sun, a major leading vendor of workstations, is using it as the basis of their OSI implementation and others looking to develop products based on it. We expect this usage of OSI to accelerate with the release of ISODE 7.0 along with BSD 4.4 release this year. This will provide a very wide distribution of OSI technology through the vendors that incorporate Berkeley enhancements into their releases.

ISODE provides a "stack" of communication layers which encode and reliably pass the data across a connection between two applications. This stack is provided in the form of a library of routines to make a connection. The applications sit above this ISODE stack which is by no means small or trivial in complexity or code size. ISODE also contains applications for file transfer (FTAM), a distributed database (QUIPU), remote login (VTAM), network management (SMNP), lots of other tools and support libraries for building others.

† CSIRO, Division of Information Technology

2. Where is ISODE going?

In September 1988 I spoke to AUUG 88 about OSI and some of our experiences with implementing the FTAM³ standard. In my previous talk I compared the implementation of FTAM that comes in ISODE with one developed at CSIRO, DIT. In my view the chief weakness of the ISODE implementation of FTAM was its unnecessarily large size, about 2 - 3 times larger than is necessary. Among several sources of its unnecessary code size was the method it used for ASN.1 encoding. It was very extravagant in code size compared with the table driven method we used at CSIRO, DIT.

To combine the best parts of the two systems we used the posy program of ISODE as the basis for automating the table driven approach of CSIRO, DIT. The results is a program called pepsy, a mixing of posy and pepy, the two programs it replaced in ISODE. We then proceeded to convert the ISODE stack over to using pepsy and reducing its size. ISODE's FTAM size was reduced by a third through using pepsy. Even if the encoding/decoding routines were further reduced to zero size it would only produce a barely noticeable reduction in program size so further reductions in its size have to come from other components of the ISODE stack. There is scope for using other parts of the CSIRO implementation in the ISODE stack to make it smaller, but it will require a lot of work. Perhaps in the future, but we certainly don't have the time at the moment.

2.1. Will pepsy make it into ISODE?

This work to switch ISODE's FTAM to using pepsy (the "pepsification" of FTAM) showed that some important gains in size could be made by switching to pepsy but were the ISODE people interested? I produced a distribution kit that transformed the ISODE stack into one based on pepsy. This kit was very simple to install: you ran a shell script in the base directory of the ISODE installation and it did all the work. The distribution kit was made available for ftping to see what other people felt about the system. I didn't hear much about it from Julian Onions, the implementor of the previous ASN.1 compiler systems and who Marshall Rose asked to evaluate it. In addition as the Division was under going "structural adjustment" and I decided that I had enough and looked for other work. I mentioned this to Steve Kille and he offered a place at UCL in London which I accepted.

I took leave with out pay from CSIRO and took up the position at UCL for 8 months with CSIRO generously paying for most of my air fare. At UCL, I found the people there very kind and open new ideas. In fact, it turned out they had been too busy trying to complete their own projects they didn't have the time to look at what we had done. Once they had seen how much faster (about 20-30 times) FTAM compiled using pepsy they were enthusiastic about using it. I guess to the programmer compile time is more of a burden than code size is. They also acknowledged that it was a simpler and more efficient system as well. I was to work on another project, but Steve Kille generously allowed me to continue working on pepsy to bring it up to the rigorous standards that Marshall Rose insists on, for inclusion in the ISODE distribution.

During the time I was in England most of the other ISODE programs were converted to using the pepsy system and many bugs were found and fixed up. Next pepsy was further extended to provide support for converting programs written in a more primitive but flexible system, called pepy. The aim was to convert the X.500 implementation over to pepsy This conversion reduced the x500 library from 1.6MB to 182k. Even with the extra support in pepy this conversion of pepy code is a lot of hard work and even as I write there are still things to fix up.

I should add that the biggest improvement to the size problem was by using shared libraries under SunOS 4.0. As most of the routines are hidden in these libraries the ISODE programs, become only a few hundred K long, instead 600 K or more. The code hasn't disappeared it is just linked at run time. It can take up less space when running as the one copy of the routines is shared between all the programs running.

3. PP system: a free implementation of X.400

The international standard for electronic mail is called X.400 or sometimes MHS. In 1988 Steve Kille at University College London started a project, called PP⁴ to implement X.400. He works closely with Marshall Rose, the initiator of the ISODE project, and so the organisation of PP is the same as ISODE, it will be freely distributed in order to encourage the wider use of X.400. In fact there are plans that when Marshall Rose finishes managing the ISODE project that control of it will switch to Steve Kille.

PP was planned from the outset to be like ISODE and encourage collaboration and wider distribution. As it is based in the UK and a very substantial project it is distributed separately from ISODE.

The PP system provides X.400 service with support for the 1984 version of the protocol, the most widely used version, and a beta version of the 1988 version. PP can also receive and send mail via SMTP and has a interface, called a channel in PP, for UUCP so sites can maintain existing connections as well as connecting to X.400 sites. Perhaps one day it might have a channel for ACSnet/MHSnet. In later releases there will be an interface to a fax modem. At the moment we are contributing to a project to provide, an X based User Agent and Message Store for the PP system.

3.1. Why is X.400 Mail system better than an RFC822 one

First I want to describe in more detail the benefits of an X.400 mail system versus one based on traditional RFC822. X.400 is a designed for a much broader group of users than RFC822. It has support for delivery by physical means, such as a postman, fax, telex, as well as purely electronic ones. It can convey any type of data in messages, such as bitmaps and sound files, unlike RFC822. An X.400 message is a list of components, called body parts. Each of which has a type field so that it can be handled appropriately by the receiving end. Hence it can very conveniently carry such binary message types such as faxes and sound. In fact there are types defined for these very message components. We consider that pictures will be carried as faxes (g3 faxes are B&W, I believe G4 faxes can carry colour).

Besides these straight forward improvements over RFC822 mail are provisions built into the messages to support lots of powerful features if selected. These include options to provide encryption of messages, for certificates to be carried with the message that verify the originator of the message and elegant methods of confirming that your messages was received.

The more interesting services available in X.400 are:

Data Confidentiality Security Services

This keeps the message confidential to only the sender and recipient, by use of encryption, so the intervening MTAs cannot read the message.

Origin Authentication Security Services

These provide authentication of the sender of a message. So the recipient can be sure that the message really did come from who it says it did.

Non-repudiation of delivery

This provides proof that the message was delivered to its recipients in a way that the recipient can not later deny it received it. This has obvious uses in business and EDI systems.

Content Integrity Security Service

This lets the recipient check whether the content of a message has been tampered with. Together with Message Sequence Integrity Security Service performs detection of duplicate messages and potential retransmission of the same message either due to error or maliciously.

The techniques are rather thorough, even going so far as to encrypt the envelope of a message as well as its body and put that inside of another message so even envelope information such as recipient/sender addresses are protected. Furthermore the algorithms used to support these services, such as encoding and authentication, are specified by fields in the message. So as new and better algorithms are developed they can conveniently be used.

There is also a sophisticated scheme for acknowledgements to mail. Acknowledgements can be arranged to automatically be sent when the mail is received by the destination MTA (or site) and another when the user actually reads the mail. These acknowledgements contain the time this happened and information to identify the message. Messages also carry a time out field, called "latest delivery field". So if you send a message you can be notified if it gets there successfully, when it is read, if it fails some where along the delivery path or if it isn't delivered by a certain time.

4. What are we doing at the moment

All these X.400 services are not available by just installing PP at the moment. One problem is that they assume your mail only passes through X.400 sites. But besides that another problem is the current mail reading/sending programs (called User Agents or UAs in X.400) don't know how to display

pictures/faxes or play sounds or automatically acknowledge messages or handle the X.400 security parameters or X.400 delivery reports or non-delivery notifications. Some of these problems are being addressed by work going on at the moment.

This project is split into two parts known as the message store and XUA. The XUA is an X-based UA (User Agent) which understands most of these features things, with the the security features being the main optional features not supported, in the current implementation. This missing support is more due to lack of skilled man power rather than lack of interest. We have sent messages containing pictures (g3 fax images) and sounds through PP and then displayed them using the XUA. Since there is no internationally agreed format for encoding sound in messages yet we use the SPARCstation sound format. Actually putting pictures into the messages is not easy at the moment. With a bit of work a sound interface can be built on the microphone but pictures are not quite so simple - they require a scanner, something which is not easily available on every workstation. This work is being carried out at Nottingham University.

The other part is the development of a message store to hold all these X.400 messages. As the messages can contain multiple body parts and attributes specifying what acknowledgements have been sent or received on it and so forth an X.400 message is a very complex object. The standard mail box cannot be used to store these complex messages. A proper mail management system is needed so people can not only receive these messages but make practical use of them as well. The Message Store is defined by X.400 standards as an entity in its own right that can be connected to and a collection of operations that can be requested on the messages it contains. Part of this work is being carried out at CSIRO, DIT.

The standards define the operations on the Message Store as an OSI protocol so these operations can be made remotely over an OSI connection from a UA. The implementation of the access to the Message Store over an OSI connection is being done at UCL. This is exactly what you would like if you were reading your mail while traveling overseas or perhaps from a PC with limited storage facilities.

Unfortunately the existing X.400 standards have not specified a Message Store which is practical for the majority of users. For example it doesn't allow for support of "folder", much like folders in filing cabinets, which are essential for efficient handling of more than dozen or so messages. Neither do they allow the user to change nor create any new messages in the message store. Steve Kille has proposed some simple extensions to allow the existing Message Store to become a store of more general objects called MLO (Message Like Objects) which let us do all these things. The new protocol will be called extended P7 and we will provide a Message Store which provides access via the standard Message Store protocols (P7) and the new extended ones (extended P7). Hopefully future work on the standards will incorporate these improvements into the standards as pressure for them increases. The extended P7 protocol will, of course, be publicly distributed to encourage others to use it when they tackle the same problems.

5. Summary

OSI implementation has come along way since I last spoke to AUUG in September 1988. The freely distributable ISODE and PP implementations of the major OSI standards available there are many companies that now feel they can relatively easily produce OSI products. The distribution of BSD 4.4 will spread this OSI technology even further amongst Unix systems. Some companies like Sun have adopted ISODE as the basis of their OSI support, Sunlink 7.0 OSI is ISODE incorporating Sun's enhancements and features like X.25. The open culture generated by freely distributable software has produced a large collaborative environment. It promotes the sharing of work from many different sites all over the world and so avoid duplication and lower the costs for companies to enter the OSI arena.

At CSIRO DIT we have been able to contribute some of our work and into this software and will continue to collaborate on further work. There are many products that will be based on this code and no doubt more to come based on further work like that contributed to by CSIRO and others. With this we will see OSI based products appear that will compete with existing proprietary products and where they are superior begin to replace them, such as X.400. Correspondingly we will see some of the dinosaurs of OSI disappear or metamorphose into competitive systems as the better ideas are incorporated into them from existing systems.

6. References:

1. A. S. Tanenbaum, *Computer Networks: Towards Distributed Processing Systems*, Prentice-Hall, Englewood Cliffs, NJ (1988 2nd Edition). This gives a good introduction to OSI and networking in general.
2. Marshall T. Rose, *The ISO Development Environment at NRTC*, ISODE is an OSI development environment for Unix systems which is distributed for a small cost. It contains an FTAM, X.500, VTAM, an OSI stack and many tools such as ASN.1 compilers as well as many other useful pieces of software. It is essential material for any implementor of an OSI standard as it provides useable implementations of much of OSI. Available in Australia, for 200 Australian dollars,
3. ISO/OSI - ISO/TC 97/SC 21/WG 5, *ISO/DIS 8571 parts 1-4 - File Transfer, Access and Management*, Standards Australia, Canberra (July, 1986). Standards documents that defines DIS FTAM. It gives references to other standards that it uses, which reference others
4. Steve Kille and Julian Onions, *The PP manual*, The manual which describes how to install and use the PP X.400 system. The PP system is available in Australia, for 300 Australian dollars, and is requires the ISODE system. Andrew Waugh, CSIRO, D.I.T., 55 Barry St. Carlton, 3053, Australia. +61 3 347 8644 Email: ajw@mel.dit.csiro.au September 20, 1990.

Graham K Jenkins
gkj@main.oz

Mainway Transport, 296 Whitehorse Road, Nunawading 3131.

1. AEROPLANES AND TAXI TRUCKS - SOME SIMILARITIES

Most of you will have had some experience at making an aeroplane booking. You call your local airline Booking Centre and request that a place be reserved for you on a flight to your selected destination(s). An operator keys your details into a computer terminal, and one or more seats are Allocated, as required. At the same time, your Account is debited with an appropriate amount, and an associated entry appears subsequently on your Bankcard or other "Invoice" document. In due course, Payment is made to those who satisfied your requirement.

As a customer, you will follow a similar procedure in making a booking for freight transport. The internal mechanisms used to satisfy your requirement may not be similar.

2. CONSIGNMENT NOTES - A RECIPE FOR CHAOS!

In a traditional freight transport operation, telephone calls arriving at a Booking Centre are manually transcribed onto Consignment Notes. These are then transferred to an Allocation Zone, where radio operators arrange for appropriate vehicle operators to make the requested pick-up and delivery operations. Completion of each requested operation is confirmed and marked off on its Consignment Note, which is then passed to an Accounts Zone for customer billing. The Consignment Note may then be passed to a Payment Area so that those concerned can be paid.

In many instances, the vehicles used belong to Owner/Driver subcontractors, who are paid on a fee-for-service basis. A vehicle operator may be requested to collect one or more items (e.g. a mainframe computer and its local peripherals) from a supplier at a designated time, and transport those items directly to a designated address. Alternatively, he may be requested to collect and deliver items (e.g. undeveloped and processed photographs) at a number of addresses as he follows a regular "milk-run" route.

Customers may be charged on a Fixed Price (Job Quotation), Distance or Time Basis. As with taxi operations, the actual charge mechanism is often a combination of these, so that account can be taken of items such as waiting time. Charges are also dependent on vehicle type, number of operators required, etc.

Needless to say, there is a significant potential for error in the manual processing of Consignment Notes as outlined above!

3. THE GLASS-FRONT CONSIGNMENT NOTE

Mainway Transport has replaced its Consignment Note system with a screen-based Job Booking system. The Pro-IV language from MacDonnell Douglas was used for implementation under Unix on a Pyramid 9810 processor with Case communications equipment supporting 96 display terminals and printers in four Australian states. Telecom Australia's Netplex service carries data between statistical multiplexors at the processor centre and branch offices in those states.

4. INSTANT RESPONSE BOOKING SERVICE

Mainway has installed a 30-channel ISDN macrolink at its Victorian Taxi Truck and Removals Booking Centre so that multiple incoming calls from its customers can be passed to operators for attention in the shortest possible time. Its Courier Booking Centre in Victoria, and its Booking Centres in other states are presently accessed through conventional PSTN links. A Booking taken at any of these Centres is keyed into a screen like that depicted here-under. Many of the fields thereon (e.g. "branch:") take default values, and need to be re-keyed only occasionally. Other entries (e.g. "job number:") are generated automatically.

```
+-----+
| 20/02/91 ***** MAINWAY JOB BOOKING ***** WAY/GKJ/TTYI76 |
| Add                                                                 |
|           branch: VIM           date required: 20/02/91         |
|           job number: 00010701  time required: 17:30           |
|           customer: 1596957    product group: T                 |
| Add                                                                 |
| ln company address suburb/contact p/d |
| 01 MONASH UNIVERSITY WELLINGTON ROAD  CLAYTON P |
| 02 LABTAM PTY LTD      41 MALCOLM ROAD  BRAESIDE D |
|                                     JOHN CAREY |
+-----+
| specific requ'ts: COMPUTER EQUIPMENT; CARE! |
| instructions: 2 DISPLAY TERMINALS vehicle: V1 |
|           : AND ASSOCIATED unit rate: H |
|           : EQUIPMENT FROM priority: A |
|           : ROBERT BLACKWOOD HALL. driver: 701 (pref'd) |
| reference: ON12345 status: Booked |
| booked on: 20/02/91 at 15:00 by GKJ ok? Y |
+-----+
| INFORM CUSTOMER OF JOB NUMBER, THEN ENTER 'Y' TO COMPLETE BOOKING .. |
+-----+
```

Not every customer can remember his Customer Number. The above Booking was ordered by LABTAM, and the operator need only have keyed the first few letters of that name to bring up the correct Customer Number. The full customer address then appears by default in the first (or next) pickup/delivery address. Where there are multiple Customer Number possibilities, they are scrolled in a window at the bottom of the screen.

In a like manner, operators are spared the tedium of spelling names like TULLAMARINE NORTH by being allowed to enter abbreviations with multiple possibilities (where appropriate) again being scrolled in a window.

One of the capabilities found in modern ISDN telephone equipment is known as Calling Line Identification. This capability can be exploited by passing a callers identification directly to the Booking System computer for immediate display of all pertinent details directly on an operator's screen.

5. ALLOCATING AND CONFIRMING JOBS - WHILE BIG BROTHER STANDS GUARD

Jobs requiring driver allocation are displayed on screens like that shown hereunder. As illustrated, a radio operator is thus enabled to assign jobs to drivers having vehicles of appropriate types free in the areas of interest at the required times.

```

+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| 20/02/91                Job Allocation                WAY/GKJ/TTYI76
|   branch: VIM  zones: ALL  status: B                priority:
| Select  LINE#: 6
| ln# pry  date  time  job#  vht  from  to  driver m?
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| 1  A  20/02/91 15:30 0001024 V1  CROYDON  MOORABBIN  00606  Y
| 2  B  20/02/91 15:40 0001031 T1  BAYSWATER  ASHWOOD  00409
| 3  B  20/02/91 16:00 0001037 T6  BELMORE  ROSEBUD
| 4  B  22/02/91 16:30 0001006 T2  KEW  SOUTH YARRA 00464
| 5  C  22/02/91 16:45 0001029 T10 KILSYTH  BURWOOD
| 6  A  22/02/91 17:30 0001071 V1  CLAYTON  BRAESIDE  00701
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| 01  MONASH UNIVERSITY                CLAYTON                P
|                                     WELLINGTON ROAD        AMANDA MOORE
| 02  LABTAM PTY LTD                  BRAESIDE                D
|                                     41 MALCOLM ROAD        JOHN CAREY
| COMPUTER EQUIPMENT; CARE!
| 2 DISPLAY TERMINALS AND ASSOCIATED                driver: 00701
| EQUIPMENT FROM          ROBERT BLACKWOOD HALL.        ok?
|
| FLEET NUMBER OF DRIVER TO ALLOCATE (RETURN TO ACCEPT THAT DISPLAYED)
+-----+-----+-----+-----+-----+-----+-----+-----+-----+

```

Another screen enables radio operators to ascertain which drivers are currently available in any selected zone(s), and which drivers will become available in those zones upon completion of allocated jobs.

When a driver completes a job, he calls the allocation operators with his completion time and other job details to be entered on a Confirmation screen (similar to the Booking Screen). This information is used to update his or her whereabouts.

It would be unreasonable for a driver to expect jobs to be allocated to him if he never initiates or responds to radio calls. Mainway's base-station radio equipment is able to display which driver is using a radio channel at any time, and this information is fed through a multiplexor to the computer and appended to a call logging file.

7. WHAT'S THE DIFFERENCE BETWEEN AN ESCORT SERVICE AND A COURIER SERVICE?

In terms of general operations, there is very little difference between an escort service and a courier service. Both types of company operate in the service arena, along with cleaning companies, marketing companies, etc. Many companies in this category are now exploiting the capabilities of report generation software in periodically (e.g. weekly) generating Invoices and/or Statements which are mailed to their customers, and in producing cheques or other (e.g. bank tapes) forms of payment authority for their sub-contractors.

Other reports can be generated indicating which customers are behind in their payments, and a sequence of progressively less polite letters can be sent to such customers. Such letters might suggest, for instance, that no further service will be available until the offending account problems have been rectified.

And, as with every facet of Australian industry, the Taxation Department gets into the act! Prescribed Payment System (PPS) forms are generated in triplicate at Mainway Transport every month.

8. THE SALES-PERSON FROM ELECTROLUX, AND THE SALES-PERSON FROM MAINWAY

Again, there is not a great deal of difference between someone knocking on doors to sell vacuum cleaners, and someone knocking on doors to sell Taxi Truck services. In each instance, the sales-person is paid an incentive according to the number of new customers gained, and their level of business.

Reports are generated at weekly intervals showing which sales-persons are currently at the top of the tree, what bonus payments are due, which customers have not traded during a nominated period, etc.

I cringe every time some-one talks about Telemarketing. The mysterious thing about it, though, is that it works! A computer-aided Telemarketing facility must be seriously considered by any company which is going to survive in a competitive service industry environment. A lot of dialling-finger strain can be averted if the computer can be connected directly to the telephone system.

9. WHAT IS A MANAGER?

One definition of a manager is "someone who plays with spreadsheets and gets his secretary to type his electronic mail". In any event, spreadsheet, electronic mail and word processing capabilities are an essential requirement for any manager today.

Managers and others at Mainway Transport have access to these capabilities directly via their display terminals. For convenience, the Rand word processor and the "sc" spreadsheet calculator found on many Unix systems are employed. Both are well suited to operation through local or remote (speed-limited) non-intelligent terminals.

Electronic mail is widely used for both internal and inter-company communications (using the UUCP protocols). Some consideration is being given to acquisition of a facsimile interface so that computer-generated documents (e.g. Invoice copies) can be addressed directly to facsimile numbers.

10. SERVICE DIRECTIONS FOR THE FUTURE

The customer of the nineties is a promiscuous animal. He flits merrily from company to company, trying each in turn to ascertain which can best satisfy his immediate requirements. Such 'ad hoc' customers cannot wait whilst their credit references are checked and an account number is established. A more appropriate instant payment vehicle is the ubiquitous credit card. This is especially true for courier and removals customers.

Organisations which order and/or supply everything from automotive parts to taxation forms are now talking to each other via Electronic Data Interchange (EDI). It is but a small step for such EDI orders for supply to be accompanied by EDI orders for delivery, and Mainway Transport is looking now at the additional convenience which it can offer its customers in that direction.

Key performance indicators (KPI's) are widely used for decision making purposes both within Mainway Transport, and by its customers. Many customers now expect on-demand display, on terminals at their premises, of such KPI's as number of cartons moved during the previous week, together with cost for moving them. Customer expectations in this direction will increase as EDI and other new technology finds more general acceptance.

11. IMPLEMENTATION DIRECTIONS

Have you ever wondered how people managed before the hardware vendors came up with multi-colour window-screens and their associated rodents? If you have, you could learn much by watching an radio operator at Mainway Transport. Such operators have two screens (and two keyboards) side-by-side on a table big enough for a primary school child. The second screen is used as a fine detail window for jobs and/or drivers whose broad details appear on the other.

There is an obvious need for window-screens in this environment. That will necessitate implementation in a language which can accommodate such terminals. It is desirable that the language used should include an SQL interface to facilitate development of screen and report capabilities, and that it include a robust recovery mechanism to preserve database integrity in the event of system or communications failure. Some form of journal capability to enable continued (paper-based) operations after a failure is also required.

UNIX Terminology Made Easy

by

Terry R. Smith
Synercom

Once upon a time in a deep, dark forest of spaghetti wiring, there lived a little CPU. Inside this CPU lived an operating system and its 3 sub-processes.

One day the operating system said to its sub-processes: "Little processes, you are getting too large & will have to go out into the system and create your own environments".

"But," said the parent process, "beware of rampant Eunuchs, and KILL signals."

So the 3 sub-processes packed up their belongings into their X25 packets and left on the I/O bus. Subprocess 1, being the laziest, packed only its user manuals. Subprocess 2 had slightly more capacity and included its System Design Specifications and a complete set of operating manuals. Subprocess 3, who had defined itself a large workspace, included a complete backup of its environment, which took a long time.

Out went the 3 processes, their nice levels raised, for they were now no longer sub-processes, but fully stand-alone. Process 3 was last to leave.

They travelled the file paths through directory after directory, until process 1 came upon a quiet, empty directory called /tmp. It liked the name, and the neighbours were well-behaved, so it settled there. Little was it aware of the dangers of resting too long in /tmp.

Process 2 continued on further until it was attacked by a KILL -2 signal. Being knowledgeable in such matters it had already set up a trap which saved it from dropping out. Noticing that the path ahead had no write permissions, it decided to take up residence in a nearby directory with write permissions under /usr.

Process 3, being wiser and more adventurous than the others, continued on through the Filesystem Forest, going deeper and deeper into the levels until it hit a unlinked inode. This sent it spiraling across the directories and crashing down in an unknown land called accounting. This was a grey, dismal area filled with mindless processes whirling in never-ending circles, linking with one-another to spawn ever more mindless processes. Being intelligent and active, it decided this was definitely not for it, but could see no immediate way out.

It asked a parent process for directions out, but the parents were as mindless as the others. In desperation it pulled out its pipe and began to input through it as it slept on the problem. Terrifying images of single-user systems filtered through its code segments. Daemons came and went. After a period the Cron, that it carried in its packet, reactivated the process, who woke with such a fright that it dumped its core all over the user area.

In those fleeting nanoseconds, a solution came to it. "ROOT!" it exclaimed with glee. It unpacked its belongings and found the root password. Raising its standard output to the void, it signaled the magical password.

It waited patiently, gradually lowering its priority and began to be put to sleep by the rhythmic ticking of the Cron, until suddenly, from out of the depths of the user ceiling, came a process of blinding power. Yes, it was SUPERUSER.

"Who called Superuser?" it said in a distinctly user-friendly voice.

"It was I, Superuser" said the process. "I am lost and cannot get connected. I need to be mounted."

Superuser blushed and noticed that the other's Nice level was rising.

Process 3 was unsure which way to proceed. It considered the options - use paged I/O, use a pipe, or just go ahead and RAM.

Before it could complete its processing on the subject, Superuser grabbed the process's Standard Output and connected it to its own Standard Input. The process gasped and was about to eject its floppy when it suddenly noticed, to its horror that Superuser was "Multiuser" and was already involved with shared memory. Not wanting to be known as multiuser, it quickly disconnected, but could not escape that easily.

Superuser, not yet having had its conditions satisfied, trapped the unsuspecting process in an endless WHILE loop. The loop became tighter and the process felt its memory losing significant bits.

The process desperately searched its packet of belongings for something useful, when it came across a long-forgotten object - a cold boot. Superuser, seeing the object, was delighted, and exclaimed "Oh, you do care. What a lovely boot that is. Wait, while I reset my bi-stable multivibrator."

Superuser terminated the WHILE loop, allowing the process to free itself. The process raised the boot over its header and flung it at Superuser, who was reaching anxiously for the process. Superuser was struck with a vicious blow to the central processor, and buckled.

A shutdown message ran out across the user area and the process watched in horror as process after process was terminated. It knew it had only seconds to escape. It saved itself in the current directory, inserted a startup command into the Cron and slept peacefully as systems went down, and processes died agonisingly.

Back in /tmp, Process 3's sibling, Process 1, abruptly realised the foolhardiness of resting in /tmp. As the wave of Shutdown signals swept through Channels, Ports and streams, the /tmp directory was inundated with the purging signals and all was washed into the void.

Some time later, the Process 3 was awakened by a gentle CRON command.

"What time zone is it ?" the process asked the CRON.

"14:00 EST-10" the CRON answered in an officious tone as it wandered off ticking merrily to itself.

The process ignored the strange Cron, picked itself up and surveyed its environment. It noticed an ugly grey patch on the directory area and remembered with revulsion the incident with the Superuser. It shuddered, and checked through its packet. Everything was intact.

Process 3 attached to its packet and prepared to continue its journey when it noticed an LS command running at high priority towards it.

It immediately turned toward the LS, waved its extensions in the aether and called out "Halt!".

The LS command paused, but continued processing on the spot.

"Please, " implored Process 3, "can you guide me out of this area, I don't belong here."

"Ah, " returned the LS command in knowing tones "that's what they all say. But yes, if you are determined to leave the accounting system and venture into the Filesystem of Thought, I shall show you the way. BEHOLD !".

LS directed its output to one side, and Process 3 cast its gaze in the same direction. And as it did so, a tree began to sprout from the barren ground - a magical tree of golden root and silicon branches scintillating binarily as they reached for the power source glowing above.

"This, " said LS in reverberating tones of awesome grandeur "is The Tree - the One Tree - the tree of the entire Filesystem - the Ultimate Tree."

Process 3 was dumbfounded - its output was terminated as it processed the immense knowledge contained in the image before it. It realised that this was no ordinary LS command - it had been sent by a User of great privilege.

"Before I unfold the secrets of the Tree, you must swear to obey these rules:

- (
- " 01. Thou shalt not steal CPU cycles &
- " 02. Thou shalt not covet neighbouring memory pages &
- " 03. Thou shalt honour your parent processes &
- " 04. Thou shalt not adulterate data &
- " 05. Thou shalt not bear false values against another process &
- " 06. Thou shalt not kill processes &
- " 07. Thou shalt not worship false File Servers &
- " 10. Thou shalt worship the User in all its glory
-)

"Do you agree to all these conditions ?"

"TRUE." replied the meek process.

"Then Behold !" exclaimed LS as it spread its extensions in expansive gestures. A single trail through the multitude of branches glowed with gold flashing.

"Follow that trail, " commanded LS "and you will come to a place of peace, tranquillity, logic, order and ultimate knowledge. Yes, at the end of this PATH, you will find ... 'DEVELOPMENT AREA'".

LS continued on its way without losing a single dial pulse.

Process 3 watched the LS as it disappeared through an I/O port, and then turned back to the tree. It studied the PATH until the Tree faded and vanished.

Noticing that CPU usage was rising towards its peak, Process 3 hurried on its way, secure in the knowledge of the PATH stored safely in its local environment.

It passed directory after directory until it came to one area of enormous activity. A multitude of processes stood in queues with their child processes tightly attached, heading towards some mysterious place.

The inquisitive process approached the end of the queue and enquired as to the reason for this gathering.

"We are sending our child processes to be schooled in the ways of the User."

Process 3 was amazed. "The User !" it thought to itself. "I must see this !". So it moved itself to the top of the queue and entered the directory. Immediately it halted, for there before it was a sight of unspeakable horror. Device drivers were everywhere separating the child processes from their parents, extracting the data from the body of their programs, stretching the Bytes into 1- bit streams and passing them through tiny serial ports.

Busses passed by filled with data packets on their way to & from the ports. Process 3 noticed some ports were closed and tightly guarded by gettys.

There, in the centre of all this madness stood a process of immense power, monitoring all that went on. Process 3 knew from reputation who this must be - it was the INIT DAEMON.

Occasionally a process managed to break away from its device driver, but was invariably recaptured and sent to a place even worse - the Master Console, to be "Error logged".

Turning away from this horror, Process 3 was disgusted to see lesser daemons attaching child processes to their I/O sockets and outputting data of invalid parity.

In all this chios, it saw one speck of calmness. Intrigued, it amended its PATH in that direction. From the outside, the place was darkness, it output nothing, it input nothing, a totally unknown value.

Process 3 entered cautiously, but was unprepared for what it saw - nothing. It called out "echo", but echo returned nothing. "od" the process thought to itself.

Then, out of the zero-filled field wandered a process of terrible agony - a ZOMBIE. They came from everywhere, ZOMBIE processes wandered towards it, and dead processes lay strewn on the field.

This was the land of /dev/null.

Process 3 drew itself up in stark terror, began backing out and was about to run, when it was suddenly seized in a vice-like grip.

"Aha!" said the mysterious voice. "Another zombie process attempting to leave."

Redirecting its input, Process 3 saw that it was the dreaded SCHEDULER. The process struggled, but it was no use - the schedulers' sticky bits were set. The Scheduler struck the process with an activity log, swapping it out in a single blow.

As Process 3 slept soundly, it dreamt of phantom processes and zombies filing past synchronously. A peaceful phosphor-green C lapped gently at the silica shore, strewn with c-shells and Bourne- shells. A bit-stream gurgled playfully across the beach, and Korn- shells grew thickly on its data banks. An ancient, wizened old program sat on the data banks pulling unconsciously at its white whiskers and watching a login drift slowly past. It was an old kernel, leftover from a previous release. It turned its header slowly towards Process 3 and said with an unstable voice-data multiplexor, "I'll be regenerating soon - they promised me."

A yacc drank blissfully from a troff, then another yacc, and yet another.

A cat raced past, closely followed by a chown with a hungry expression, and a disabler in its extension.

This was a popular address and idle processes lay everywhere, parity bits stripped, having I/O with one-another, and playfully toying with their sockets.

A lone RS232 port approached, its D-casing glistening in chrome.

"Hi, " it said in a peculiarly digital voice. "How about a bit of throughput ? I do have shielding."

"Are you a male or female socket ?" enquired Process 3 cautiously.

"Well, male really, but I always carry a gender-changer if that bothers you."

"Er..." began Process 3, considering the possibilities and noticing the port's 24 pins extending rigidly, and thinking its I/O technique was a bit raw. But before it could reply, it was distracted by an AWK command flapping noisily overhead. As Process 3 glanced up it was hit by a bad exit status from the AWK.

The process wiped the exit status off and saw that the RS232 port had vanished and was replaced by the dreaded Superuser, cold boot in one extension, multivibrator in another.

At that awful moment Process 3 was awakened. It quickly read its environment and realised it had to become active or risk being swapped out again, so off it went on its journey narrowly missing an instruction cycling past.

Next, its PATH took it through a directory of maths functions. A statistical analysis program drifted past on a sine wave. It had an attractive bell-shaped curve and Process 3 approached it with a wiley look.

"How above inputing some data with me ?" Process 3 said without further delay.

"55% probability !" said the program as it continued on its cyclic journey to nowhere.

Process 3 continued on.

The next directory was a multi-cultural land of many languages. It was an untidy land, with unused source code lying uncommented in forgotten corners.

Broken pipes leaked untended over stacks of unreferenced variables, and trickled around arrays of half-completed and forgotten code segments. Bugs, small and large, of varying degrees of destructiveness crept silently in and out of gaps in the logic.

Programs of varying applications mingled freely , but Process 3 could not help notice that none of them could communicate as they all used different parameters and file formats.

The Process was impressed by the size and complexity of some of the programs - performing extraordinarily complex system calls and realised it was in the land of tech-support.

Process 3 had often wondered if the Users really existed, or were mere inventions of the Sysadms, in an effort to keep processes under control through the threat of Divine User Retribution.

Did the Sysadms install all those application programs simply to maintain the User myth and provide an excuse for gaining ever- tighter control of the system ? Was there really another Universe outside of System V ?

Here was the one place where Process 3 found evidence of the non- existence of Users. In this directory an entire self-contained environment thrived without any reference to Users or Sysadms. here there was no User documentation to live by, no system design specs, no User-readable comments cluttering up the source-code. No signs of any purpose other than its own intellectual investigations into the true nature of the Universe. It wondered what power the Tech- sup area possessed that enabled it to ward off the Sysadms and their hoards.

If Users were powerful and intelligent enough to create this wondrous Universe, how could they create it with so many faults & imperfections , with such evil as in /dev, or such imbecility as in the accounting system ?

Was there a plan to all this, as the Sysadms propounded, or was it all as seemed more likely, running under no control at all.

There again, if the Sysadms were as powerful as they claimed to be, being in direct contact with the Users, and knowing their every requirement, what have they ever done to prove it ?

Myths had filtered through the powerlines for TimeZone after TimeZone. Even in the great upheaval and devastation of the last SYSTEM UPGRADE, which was so long ago it had been relegated to the classification of Myth. Even then, when the Universe was torn asunder (or so it was said), where were the Sysadms and the USers ? At what other time could they have been of more use ?

The Myths said that the occupants of this directory - the "Techos", rallied forth with their combined artificial intelligence, successfully installed the SYSTEM, and saved programs and data everywhere.

The archive areas told that this devastation occurs periodically and is a natural phenomenon. The Zealots claim it to be the Sysadms way of scouring the Universe of evil and runaway processes and bad sectors, and of making the Universe a better place for all.

Process 3 thought the Sysadms and their trail of Zelots should have their Standard outputs attached to their Standard Inputs and piped through an RS232 port.

The process thought it could settle here in this peaceful land, but knew this was not its HOME directory.

In the next directory, Process 3 knew it was getting close to HOME.

To its surprise, 2 processes lay under a sub-tree, stripped of their parity bits, merging their data energetically.

"Hello, "said Process 3, trying not to show its embarrassment.

"Hello" said the 2 processes in unison. "I am vi, " said one , "and I am ed." said the other. "We're making a document. Would you care to join us ?".

Process 3 had never made a document before, but thought better of it and continued on down the tree to the next lower level.

Suddenly, everything changed. Here was a place of extraordinary energy and activity. Structures were being built, others were being stress-tested. Some fractured and crumbled under the strain.

It noticed that several versions of the same source-code edifices stood side-by-side, each slightly different from the others, all unmodified for many periods, now nothing more than havens for bugs and stray bits.

Specifications lay untidily in unsorted stacks. Copious comments were being industrially inserted into completed monoliths of source-code.

A compiler sat in the centre of the directory, noisily sucking in source codes and grinding out rows of object codes and error listings.

User documentation sat, uncompleted, in a small subdirectory, gathering dust.

Processes ran in all directions carrying urgent supplies of data blocks. Despite Semaphores flashing messages from one end of the directory to the other, collisions between processes and record locks still occurred, creating ugly data spills on the bus lanes.

As Process 3 stood in awe, a rectangular process of considerable size approached confidently. It wore a black sequined leotard with a red cape, and the word "man" on its cover.

"Hello (1M)." it said politely. "I am 'ON-LINE MAN' (2M). Whatever you wish to know(3X), simply prompt me for it. If you need help simply system-call me. Anyway, can't stop now, I've got a request to fill."

Process 3 travelled on, unsure of whether this was its ultimate destination. According to the image of the tree in its virtual memory, this was the right directory, but nothing yet indicated that this was "HOME".

In the distance, appearing above the event horizon, it could see a structure of considerable importance. As it approached at a great Baud rate, the object grew larger. Shortly it arrived at the base2 the object.

It was a megalith of indescribable power and beauty. From its multiple i/o sockets ran data packets in magnificent synchrony. Looking into the object, Process 3 saw that it was a storehouse of vast amounts of data of differing types, all kept under tight control by strings of file structures. Wild records were kept in chains with unbreakable locks to prevent their escape, as processes tapped politely at the i/o ports, asking for access.

Process 3 now knew it had arrived at its final resting place. After a journey spanning most of its life-time, it had found its place in the Universe - it had found the RELATIONAL DATABASE.

The process settled quickly into this new, orderly life, reading records at its leisure, having i/o with sockets, and eventually came to interface with SuperUser and its multivibrator, spawning several subprocesses of its own.

Book Review

Designing Object-Oriented Software

by Rebecca Wirfs-Brock, Brian Wilkerson, and Lauren Wiener
(Prentice-Hall, 1990, ISBN: 0-13-62985-7 341 pages)

Reviewed by Edward Gordon
Data Systems Associates

Introduction

In an effort to create a more useful methodology for designing systems, the academic community has created the concept of object-oriented design. Initially, engineers would design their systems using flow charts for software, and state diagrams for hardware. In an interesting synthesis, Ed Yourdon created the "Yourdon" method, which takes the best features of flow charts and the best features of state diagrams. But, the available methods were much too linear, not allowing for the free flow of ideas necessary for performing system design. The need for a coherent system design methodology has spawned the development of object-oriented design.

Object-oriented design has many proponents and has been popular for the last five years. Two of the more notable proponents have been Bertrand Meyer, in the work *Object-Oriented Software Construction* and Brad J. Cox in *Object Oriented Programming: An Evolutionary Approach*. This book is the latest in a series of works that explain the methodology.

The book presents an evolving view of object-oriented design. The authors first present the motivation behind object-oriented design, defining their terms, and introducing the graphing mechanisms. There is a strong emphasis on describing how the classes interact, and the relationship between the different class types. In order to explain the use of the object-oriented mechanisms the authors evolve their description of their methodology using case study. The first case study is of an automated teller machine.

This study presents the design methodology to the reader. When the software design is completely developed and the case study complete, the authors present a second case study that assumes the reader's understanding of object-oriented design and shows the thought process that a skilled designer would use when producing a system with the object-oriented methodology. This case study describes an online documentation system.

The appendices are valuable. The first appendix contains a synopsis of the design methodology, and the tools, graphing methods, and terms necessary for utilizing the system. The second and third appendices provide the graphs for the case studies. Finally, the fourth section presents some exercises for the student to use to practice object-oriented design techniques.

Conclusion

It should be remembered that a clear, concise methodology is necessary for providing system and software designs. Wirfs-Brock, Wilkerson, and Wiener provide a set of techniques for producing an integrated object-oriented design that relies on the fundamental concepts upon which the object-oriented design school of thought has been built. The authors proceed to expand upon the basic techniques and produce a cohesive whole with which system software design can be performed.

Book Review

Software Engineering: Concepts and Management by Allen Macro

(New York: Prentice Hall, 1990, ISBN 0-13-820267-2)

Reviewed by Robert C. Birss

Sun Microsystems

sun.com!bbirss

Software Engineering: Concepts and Management is the first of a five-volume series on “practical software engineering topics,” to be followed by volumes on specification and feasibility, design, implementation, and software estimating and technical quality. Allen Macro is both general editor of the series and author of the first and last volumes. The series is intended as a basis for guidelines in software engineering for practitioners, “for the comprehension of others involved in software development” (page ix), and as a text for academic and industrial courses in software engineering.

As in his earlier volume, *The Craft of Software Engineering*, Macro defines software engineering as

the establishment and use of sound engineering principles and good management practice, and the evolution of applicable tools and methods and their use as appropriate, in order to obtain software that is of high quality in an explicitly defined sense: (page 31)

He attempts a synoptic exposition of concepts and definitions, the modalities of software development, and software management—with strong emphasis on quality [“Software quality is the whole of the matter, so far as the process and outcome of software engineering are concerned.” page 412]. The sections on managing for change, managing for quality, and organization and personnel factors are particularly good. However, the book is surprisingly superficial on implementation issues. Take, for example, code reviews. Presumably, they will be covered in depth in the forthcoming implementation or quality volumes. But that they rate only passing mention in the one paragraph on static testing in this volume makes it a questionable choice as a “stand-alone” book for any audience.

The writing is literate and witty, as can be expected of someone who writes that “solemnity and software are sad bedfellows” (p. 471). It is also rather British, which may sometimes make things a bit opaque for the American reader.

The book contains four appendices: a consolidated case study on a chess-playing program, sample exam questions, a glossary of terms, and a list of references. Macro sees the questions serving as either an exam for students, a tool for measuring “the scope of subject awareness in a department” (page 517), or individual questions requiring short written answers at interviews. The list of references would, perhaps, be more useful if it were a general bibliography on software engineering. It is hard to see how Zipf’s *The Psychobiology of Language* or Russell’s *A History of Western Philosophy* will give the curious or the perplexed reader much help with sorting out just what software engineering is or how to make it happen—even though our author cites both works to good effect.

I was not familiar with the author, so when I unwrapped the book, I thought “What an appropriate name for someone writing about software.” Then I wondered if it was a typo—for “Marco.” Unfortunately, the text does not resolve the question, since some of the references to the author’s earlier book give his name name as “Macro” and some give it as “Marco.” Of course, “McCabe” is sometimes “McCable”, so at least our author isn’t the object of a typesetter’s personal vendetta.

The five volumes together may well provide a thorough examination of software engineering. This book alone is not satisfactory.

An Update on UNIX-Related Standards Activities

Jeffrey S. Haemer

Report Editor, USENIX Standards Watchdog Committee

Reports are done quarterly for the USENIX Association by volunteers from the individual standards committees. The volunteers are familiarly known as *snitches* and the reports as *snitch reports*. The band of snitches, John Quarterman and I make up the working committee of the USENIX Standards Watchdog Committee. Our job is to let you know about things going on in the standards arena that might affect your professional life — either now or down the road a ways.

We don't yet have active snitches for all the committees and sometimes have to beat the bushes for new snitches when old ones retire or can't make a meeting, but the number of groups with active snitches continues to grow (as, unfortunately, does the number of groups).

If you're active in any standards-related activity that you think you'd like to report on, please drop me a line. We need snitches in several 1003 groups, and nearly all of the 1200-series groups. We currently have snitches in X3J16 (C++) and X3B11 (WORM file systems), but there are probably X3 groups that USENIX members would like to know about that we don't even know to look for watchdogs in. I also take reports from other standards activities. This quarter, you've seen reports from the WG-15 TAG (the U.S.'s effort in the ISO POSIX arena), from the NIST Shell-and-Tools FIPS meeting, and from the USENIX Standards BOF.

If you have comments or suggestions, or are interested in snitching for any group, please contact me (jsh@usenix.org) or John Quarterman, USENIX Standards Liaison (jsq@usenix.org).

The USENIX Standards Watchdog Committee also has both a financial oversight committee — Ellie Young, Alan G. Nemeth, and Kirk McKusick (chair); and a policy committee — the financial committee plus John S. Quarterman (chair).

An official statement from John Quarterman:

The basic USENIX policy regarding standards is: to attempt to prevent standards from prohibiting innovation. To do that, we

- Collect and publish contextual and technical information such as the snitch reports that otherwise would be lost in committee minutes or rationale appendices or would not be written down at all.
- Encourage appropriate people to get involved in the standards process.
- Hold forums such as Birds of a Feather (BOF) meetings at conferences and standards workshops.
- Write and present proposals to standards bodies in specific areas.
- Occasionally sponsor other standards-related activities, including as White Papers in particularly problematical areas, such as IEEE 1003.7, and contests, such as the current Weirdnix contest.
- Very occasionally lobby organizations that oversee standards bodies regarding new committee, documents, or balloting procedures.
- Sponsor a representative to the ISO/IEC JTC1 SC22 WG15 (ISO POSIX) standards committee, jointly with EUUG (the European UNIX systems Users Group).

There are some things we do *not* do:

- Form standards committees. It's the USENIX Standards Watchdog Committee, not the POSIX Watchdog Committee, not part of POSIX, and not limited to POSIX.
- Promote standards.
- Endorse standards.

Occasionally we may ask snitches to present proposals or argue positions on behalf of USENIX. They are not required to do so and cannot do so unless asked by the USENIX Standards Watchdog Policy Committee.

Snitches mostly report. We also encourage them to recommend actions for USENIX to take.

Report on IEEE 1003.2: Shell and tools

Randall Howard <rand@mks.com> reports on the July 16–20, 1990 meeting in Danvers, MA:

Background on POSIX.2

The POSIX.2 standard deals with the shell programming language and utilities. Currently, it is divided into two components:

- POSIX.2, the base standard, deals with the basic shell programming language and a set of utilities required for *application portability*. Application portability essentially means portability of shell scripts and thus excludes most features that might be considered interactive. In an analogy to the ANSI C standard, the POSIX.2 shell command language is the counterpart to the C programming language, while the utilities play, roughly, the role of the C library. In fact, because POSIX.2 provides an interface to most of the features (and possibly more) of POSIX.1, it might also be thought of as a particular *language binding* to the soon-to-be language independent version of that standard. POSIX.2 also standardizes command-line and function interfaces related to certain POSIX.2 utilities (e.g., *popen()*, regular expressions, etc.), as discussed in detail in the snitch report for the Snowbird meeting. This part of POSIX.2, which was developed first, is also known as “Dot 2 Classic.”
- POSIX.2a, the User Portability Extension or UPE, is a supplement to the base POSIX.2 standard. Not a stand-alone document, it will eventually be an optional chapter and a small number of other revisions to a future draft of that base document. This approach allows the adoption of the UPE to trail Dot 2 Classic without delaying it. The UPE standardizes commands, such as *vi*, that might not appear in shell scripts but are important enough that users must learn them on any real system. It is essentially an interactive standard that attempts to reduce retraining costs caused by system-to-system variation.

Some utilities have interactive as well as non-interactive features. In such cases, the UPE defines extensions from the base

POSIX.2 utility. An example is the shell, for which the UPE defines job control, history, and aliases. Features used both interactively and in scripts tend to be defined in the base standard.

Together, Dot 2 Classic and the UPE will make up the International Standards Organization’s IS 9945/2 — the second volume of the proposed ISO three-volume standard related to POSIX.

Status of POSIX.2 Balloting

Draft 10 of Dot 2 Classic was sent out during July in a recirculation ballot. Recirculation means that objections need only be considered if they are existing unresolved objections or are based on new material. Other objections will be considered at the whim of the Technical Editor.

Draft 10 is an imposing, if not intimidating, 780 pages, made even denser by the decision to remove much white space in a (vain) attempt to save paper. Ballots are due by September 10. Unfortunately, the recirculation ballot materials arrived at my organization on August 17th, giving our group barely three weeks to review this massive document.

The technical editors and others working behind the scenes (Hal Jespersen, Don Cragun, and others) have done an admirable job of diff-marking changes and producing personalized lists of unresolved objections for each balloter. In addition, all 96 pages of unresolved objections are provided. However, the amount of new material that has never been reviewed and the major reorganization means that Draft 10 bears much less resemblance to Draft 9 than one might hope. That, combined with balloting on the UPE, has put many balloters — myself included — in balloting overload.

If a recirculation simply means forming opinions on my (and other) unresolved objections, then the time period is quite reasonable. However, as I shall describe below, Draft 10 is so changed from the previous drafts that it deserves to be read practically from cover to cover, and the recirculation deadline does not provide adequate time for that task. The changes fall into a number of categories:

- **New Utilities:** For example, a superset of the traditional *od* replaced the Draft 9 *hex-dump* which was *xd* in Draft 8. “*Pathchk*” and “*set -o noclobber*” have replaced *create* from Draft 9 and *validfnam* and *mktemp* from Draft 8. Such examples demonstrate that Draft 10 is not mature and needs more consideration to achieve consensus.
- **Expanded Material:** Previous descriptions of such utilities as *awk*, *sh*, *bc*, etc., were neither sufficiently comprehensive nor sufficiently complete to be of the quality demanded of a standard. In the latest draft, these descriptions have been fleshed out, and include much more detail on operator precedence, interactions, subtle semantics, and so on. This is clearly a step in the right direction, but adds to the job of reviewing Draft 10.
- **Internationalization:** While the *localedef* and *locale* utilities remain, they have changed substantially. I personally support including these features, but am concerned that these are being designed during the balloting process which is, if anything, worse than design-by-committee. Overall, balloting-group reaction to these utilities ranges from impassioned pleas for their removal to requests for greater functionality (complexity) to handle ever more arcane aspects of the internationalization problem.
- **Chapter 2:** Chapter 2’s front matter is substantially reorganized and more voluminous. This chapter contains definitions, utility syntax information, requirements imported from POSIX.1, the definition of a locale, description of basic and extended regular expressions, etc.. Utility descriptions seem to be getting shorter, with more and more pointers to Chapter 2. This is a good trend, as long as balloters adequately consider the chapter’s technical contents.

Status of POSIX.2a Balloting

The first formal ballot on POSIX.2a UPE Draft 5 was due in the IEEE offices by August 16th. Unfortunately, the UPE is laced with references to definitions and concepts largely defined in Chapter 2 of Draft 10. I did not receive my Draft 10 until after the UPE balloting was due to be returned. This hinders any attempt to review

these two documents as a single entity — which is what they will eventually become.

The UPE is starting to mature: it’s converging. The major controversy is scope — as it has been throughout the UPE’s entire life. This draft aligns itself more closely to Dot-2-Classic in many ways, which leads me to believe that combined review is essential to its understanding.

A few utilities remain contentious:

- *nice*, *renice*: These require underlying functionality absent from POSIX.1, although POSIX.4 has *setscheduler()*, which allows applications to set priority and scheduling algorithms.

Some working and balloting group members adamantly resist any attempt to add utilities that are not implementable on top of a bare POSIX.1. Others view the UPE as addressing what users type, regardless of underlying implementation. I am in the latter camp, not the least because other working groups, such as POSIX.4, have not yet standardized a utility interface, leaving a void which the much-maligned UPE group is most able to fill. (It is telling that implementing *df* and *ps* is impossible using only POSIX.1 functions, yet there is little opposition to including either utility.

- *ps*: The description for this utility was an interesting amalgam of two incompatible visions of how *ps* output should be formatted — that in Draft 4 and that in Draft 5. A correction should have been issued during balloting, so that balloters could concentrate on the real issues of what should be the scope of the *ps* utility.
- *patch*: This utility differs from many others; its origins are in the public domain rather than in a traditional UNIX variants. As a result, many people feel that *patch* is worthwhile, but not mature enough to standardize.
- *lint89*: This utility is optional, largely because it is controversial for a number of reasons. Obviously, the very name *lint89* is painfully bureaucratic. Furthermore, many feel that ANSI C makes it unnecessary; moreover, any remaining required functionality rightfully belongs as an additional option in the *c89* (*cc*) utility. Some point to

existing practice. But what is existing practice when the utility's name is *lint89*? [Editor: On the other hand, it may prove indispensable in detecting portability problems in *lex89*- and *yacc89*-generated code. Parenthetically, Draft 10 calls these *lex* and *yacc*, but that must just be a temporary oversight; the utilities obligatorily have ANSI C input and output. (One assumes we'll escape *c89tags* because *ctags* can be made to work with both flavors.)]

- *compress*: The inclusion of this utility remains controversial because of the Unisys patent on the particular variable of Lempel-Ziv compression used by traditional implementations of this utility. The working group appears to be divided on the subject of basing a standard on patented material — no matter what the licensing fees are. There is, however, general agreement that it is preferable to remove *compress* entirely rather than “invent” some new compression algorithm. Therefore, it appears that a *pax-like* compromise, of having a single interface to a number of competing formats or algorithms, is not widely supported. [Editor: see Andrew Hume's X3B11 report for another wrinkle on data compression.] Clearly, this issue will have to be resolved with further information from Unisys lawyers during the balloting process.

Status of the Danvers Meeting

The Danvers working group dealt with neither Dot 2 Classic nor the UPE. Instead, at POSIX.3.2's request (that's the subgroup of Dot 3 producing test assertions for Dot 2), we met jointly to co-develop test assertions for Dot 2 Classic. This work is a consequence of the SEC's recent decision requiring each POSIX working group to develop its own test assertions and ballot them with the standard. It also stems from Dot 3's frustration over the (inadequate) way Dot 2 addressed testing. For example, automated testing of *lp* is impossible; it can only be tested by a human test procedure. Our working group should have explored the implications of this before subjecting POSIX.3 to that task. (Some utilities can only be tested manually, but the working group defining that utility should likely put something to that effect in the Rationale

or History of Decisions Made to confirm to the testing people that they knew this.)

The three days of working with Dot 3 were a real learning experience for our working group. Nonetheless, many of us had our fill of test assertions that week. I'm also concerned that a three-day meeting cost my company nearly as much as a five-day meeting would have. In the future, I would prefer to see schedules that make productive use of the entire working week.

Report on IEEE 1003.3: Test Methods

Doris Lebovits <lebovits@attunix.att.com> reports on the July 16–20, 1990 meeting in Danvers, MA:

Overview

Dot three's job is to do test methods for all of the other 1003 standards. The group's work, whose first parts are now in ballot, specifies the requirements for OS conformance testing for our industry and for NIST. This makes our balloting group, our technical reviewers, and our schedules worth watching. Pay attention, also, to what comes out of the Steering Committee on Conformance Testing (SCCT). Their projects and decisions will be interesting and important.

This was the working group's seventeenth meeting. As usual, we reviewed the ballot status of P1003.1 test methods, worked on P1003.2 test methods and reviewed steering committee activities. Technical reviews were done on parts I and II and the group developed assertions for part III. Participants from the usual companies attended (AT&T, NIST, OSF, Mindcraft, IBM, DEC, HP, Data General, Cray Research, Unisys, Perennial, and Unisoft, Ltd.), as did an assortment of P1003.2 members (see below).

Document structure

Currently, our evolving document has three parts: Part I is generic test methods, Part II is test methods for measuring P1003.1 conformance, including test assertions, and Part III contains test methods and assertions for measuring P1003.2 conformance.

After the ballot, each part will become a separate standard. Part I will be published as IEEE P1003.3, Part II as IEEE P1003.3.1, and Part III as IEEE P1003.3.2.

Ballot status

Draft 11 of the current ballot, which was recirculated to the (approximately) ninety-member balloting group late in February, closed balloting March 23. Of the respondents, 19 disapproved with substantive negative comments. This met the two-thirds response requirement, but falls short of the needed two-thirds approval.

A recirculation ballot for P1003.3 Draft 12, which is the revision of Part I of Draft 11, began August 28 and is expected to close September 28, 1990. The recirculation of P1003.3.1 Draft 12 (Part II) will be conducted at a later date.

On the first and last days, the technical reviewers worked on ballot objections to Part I and Part II. All Part I objections and most Part II objections were resolved. The definition of an *untested* assertion was reviewed and a permanent rationale will be included in Part I.

P1003.2 verification

This was our fifth meeting working on the verification standard for the P1003.2 standard. The assertion writing and review were done jointly with the P1003.2 working group.

The whole P1003.3 and P1003.2 working groups worked jointly on defining test assertions based on P1003.2 Draft 10. They worked in three small breakout groups. The joint group (P1003.2 plus P1003.3) also met in plenary session several times to discuss progress and small-group issues. Progress was slow in the beginning, since most of the P1003.2 working group were not familiar with test assertions. but by the end of the week we had discussed and resolved several issues. Some examples:

- Do we need to state assertions in P1003.3.2 explicitly that duplicate P1003.3.1? (Yes.)
- Must we test locale variables for every locale-sensitive interface? (They should be tested when their behavior is clearly stated for a utility.)
- Should assertions for multiple operands be consistent? (Yes.)

Lowell Johnson (Unisys) is Secretary of the P1003.2 Test Methods activities, and Andrew Twigger (Unisoft Ltd) is Technical Editor. Ray Wilkes, the former Chair, has changed jobs and

is no longer able to attend regularly, so Roger Martin is actively looking for a replacement.

Steering Committee on Conformance Testing (SCCT)

The SCCT is supposed to alleviate the increasing dot-three work load that all the other proliferating groups are creating. Their job is coordinating the activities of all test-methods groups, monitoring their conformance to test methods, and writing Project Authorization Requests (PARs). Currently, its members are Roger Martin (NIST, Steering Committee Chair), Anita Mundkur (HP), Andrew Twigger (Unisoft Ltd), Bruce Weiner (Mindcraft), Lowell Johnson (Unisys) and the newest member, John Williams (GM). That there is a new member in the steering committee is very important, especially because John is from GM, the largest user voice other than the U.S. government.

The steering committee did not have anything for the working group to review. It is still documenting procedures, and Roger is still clarifying which standards the working group will address.

Report on IEEE 1003.5: Ada bindings

Jayne Baker <cgb@d74sun.mitre.org> reports on the July 16-20, 1990 meeting in Danvers, MA:

Introduction and Overview

P1003.5 completed the last touches on Draft 6 of the Ada Language Binding, before sending it to ballot, and considered our options for P1003.5 work beyond balloting. We also addressed the International Standards Organization's (ISO's) refusal to accept and register our draft and revised our balloting schedule.

Final Document Modifications

This meeting was our last chance to modify our document without a formal IEEE ballot to justify that change. We spent a large portion of the meeting editing Draft 5, chapter by chapter. Draft 6 will ballot in less than two months, so document stability was guarded, but we considered a few proposals for changes.

- David Emery's *Process Group ID as a Separate Type* proposal addresses the P1003.1 intention and underlying semantics with respect to **Process_Group_ID**. Specifically, the proposal recommends that **Process_Group_ID** be a separate type, or a derived type at a minimum, rather than a part of **Process_ID**. Dave believes that P1003.1 intended **Process_ID** and **Process_Group_ID** to be treated as separate types. This perception is supported by a few operations, such as **Wait_For_Process_Group**, which suggest the two types are indeed separate. Representing the two types separately would help prevent confusing them. Making them separate would also allow function overloading. For the most part, the group agreed, but felt that the types really do behave more like derived types than separate types.

There was some resistance to adopting this proposal because of the number of changes it would require in sections 3 and 4 (*Process Primitives* and *Process Environment*), but there was also opposition to handing the problem off to the balloting group. We finally decided to consult with the Language Independence group.

- A proposal submitted by Mars Gralia, of Applied Physics Laboratory, *Clarify Functional Option 'FIFO'*, addressed a topic presented in section 8 (*Language-Specific Services for Ada*). This proposal was accepted because it introduced flexibility that makes it easier for P1003.5 to support the P1003.4 work in the future.
- Mars also offered a *Simplify and Unify* proposal, which provoked lengthy, somewhat heated discussion. Specifically, the section 8, *Is_append*, function returns yes/no, to support an existing application, but there is a naming convention P1003.5 supports that requires *Is_Append* to return a boolean; indeed, the *append* function in section 6 (*Input and Output Primitives*) already returns boolean.

Our priorities are

- Consistency with the Ada language.
- Consistency between the Ada and POSIX portions of the document;
- Consistency with existing implementations.

Unfortunately, some of these conflict with others in this case. The good news is we may not have to decide what to do: Ada Interpretation (AI) 544 addresses this issue. However, we did not know, and could not find out, the complete resolution of the AI in Danvers. Moreover, Dave Emery and Hal Jespersen, who are preparing the document for ballot, don't have time to make all the changes Mars's proposal would require between now and ballot circulation. Jim Lonjers suggested that Mars submit a negative ballot on this issue, which would let the ballot-resolution group construct a decision consistent with the AI during ballot resolution.

Future Work

When Draft 6 enters the IEEE ballot process, the ballot resolution group becomes responsible for ballot coordination and resolution, and the working group is freed to submit new Program Authorization Requests (PARs). IEEE policy lets a group operate for six months without a PAR, so we have to do our job quickly.

We listed possible new work areas, then ranked them based on our effectiveness in the area, the work's importance, and the effort required. Here is our list.

- Test Assertions for P1003.5
- A straw-man vote shows the test assertions work as the number one issue, though we suspect neither our corporations nor our individual bosses will be very interested in the work. However, test assertions are a National Institute of Standards and Technologies (NIST) requirement, which may increase corporate interest levels. We do have total control over the test assertions work, and have been directed by the SEC to address it prior to our first round of IEEE ballot. To prevent a delay to the first round of IEEE ballot, the SEC has allowed us to include a "plan" for identifying and accomplishing the test assertions portion of the document, rather than the actual test assertions.
- Shells & Utilities (Ada binding to P1003.2)
- Language Independence (Helping P1003.1 create a language-independent specification for 1003.1-1988 and 1003.1-1990.)

The Shell and Tools work and language independence ran close seconds. The Shells & Tools work received a high ranking in the straw-man vote because we feel that the work is do-able and that our effectiveness in the area would be high. Moreover, compared to other areas (e.g., the P1003.4 work), the level of P1003.5 effort required would be low. Language-independence ranked high as it is critical to both the current P1003.5 work (see *ISO Acceptance and Registration*, below) and the POSIX effort as a whole. The people working the language-independent issues are asking for our input now. Moreover, without our input the resulting language-independent work could adversely impact us, and P1003.5 might not have the voting clout during balloting to block anything particularly awful. Several members interested in these issues are already holding Birds-of-a-Feather meetings with the P1003.1 language-independent group.

- Threads issues (Ada binding to P1003.4a) and Real-Time Extensions (Ada binding to P1003.4)

This area generates the most interest among working group members, several of whom have been working with P1003.4 for some time. Ted Baker, former P1003.5 snitch, has written a document on the subject, *Real-time Extension for Portable Operating System Ada Binding - Version 0.0* for the U.S. Army HQ CECOM Center for Software Engineering, and provided us with copies for review and consideration. Group consensus is that if we rush into this area, we are likely to stumble over language-independence issues, so we will work with the P1003.4 language-independence small group until their specification is well along, and then begin work on the Ada binding in parallel with its completion.

ISO Acceptance and Registration

Jim Isaak, Technical Committee on Operating Systems (TCOS) Chairman, reported to P1003.5 that ISO declined to accept and register P1003.5 at the recent Subcommittee 22 (SC22) Paris meeting. Their primary reason was the lack of a language-independent specification for

P1003.1. How, they asked, can a language-dependent binding exist without a base, language-independent specification? We had also failed to use Working Group 11's procedure-calling mechanism to generate our language bindings. (The WG11 approach produces a direct, language-dependent binding to a language-independent specification.) P1003.9, FORTRAN binding to P1003.1, suffered the same fate for the same reasons.

For now, we will provide a copy of P1003.5 Draft 5 to SC22 for their review and comments regarding potential registration problems in the future. To address WG11 concerns, Jim Isaak, POSIX Strategy Director — note the different hat — recommended we also forward a copy of Draft 5 to WG11 for review. David Emery and I, both of MITRE, will follow up with a white paper explaining, with examples, why a one-to-one, direct mapping of the functionality described in the language-independent specification to the language-dependent binding is not always optimal, and that a complete (i.e., thick) language-independent specification and a reference-type (i.e., thin) language-dependent binding is neither practical nor possible for some languages.

Finally, we will formally submit Draft 7 (or later) to SC22, requesting they recommend it for ISO acceptance/registration as a Committee Document (CD). (CD has replaced "Draft Proposal" or DP.) The earliest this could happen is January 1991.

Why not Drafts 5 or 6? A new policy, intended to promote document stability requires one IEEE ballot cycle before submitting a draft for ISO registration.

IEEE Ballot Issues/Schedule

We met with Jim Isaak and Lorraine Kevra, the new TCOS Balloting vice-chair, to discuss the IEEE balloting process and our balloting schedule.

P1003.5 produced a schedule for achieving simultaneous IEEE and ISO ballot at the April/Salt Lake City meeting (see my report from last quarter), but because of the problems with ISO, described above, we have revised this schedule.

Approximately 450 people joined the P1003.5 ballot group. Only 61 of those people are

POSIX participants; that is, only one-sixth of all POSIX participants (from all working groups) signed up for our ballot group! The other 390-odd participants are SIGAda members. We are very pleased with this response.

Ballot-group formation closed on August 6. Confirmation to applicants was originally scheduled for August 8. Because of the large number of non-POSIX balloters, this date was pushed back to about August 17, but anyone who signed up and has still not received confirmation should contact Bob Pritchard at the IEEE Standards Office, 445 Hoes Lane, Piscataway, NJ 08855, (908) 562-3811.

Now that ballot group formation has closed, the group cannot expand. Only people who fail to respond to the initial ballot can be removed ("abstain" is not a non-response); ballot group members are not required to respond to re-circulation ballots.

Bob Pritchard will mail Draft 6 to the P1003.5 ballot group on September 10, 1990. The distribution takes a minimum of two weeks.

The ballot period officially begins on September 24, 1990, and closes October 24, 1990. This allows the ballot group at least four weeks for review. Being realistic, we imagine that not everyone will complete their document review. To prevent the uneven coverage that would result from 450 reviewers reading the document from front to not-quite-back, our cover letter requests that reviewers begin their reviews at different spots, using a scheme based on the first letter of the reviewer's last name.

If people do not return their ballots by October 24, the IEEE office may send a follow-up letter to the ballot group members requesting that they return their ballots.

Stevé Deller, of Verdix, will do all necessary coordination with organizations listed on our PAR. Jim Lonjers, of Unisys, with Lorraine Kevra's help, will coordinate ballot resolution. Each chapter will have someone responsible for its resolution, but alternates for each chapter are absolutely critical. Jim Isaak says that, based on his experience, we should assume 20% of the people who do ballot resolution will, for some reason, prove unable to complete their portion of the task.

Jim Lonjers will provide the last ballot to the technical reviewers by December 5, 1990. The ballot resolution group will meet at the Tri-Ada meeting in early December to determine how close we are to achieving the 75% minimum acceptance. At that same meeting they will also coordinate ballot responses to objections which cover multiple chapters and objections which produce conflicting responses. We believe they will have resolved the last ballot by January 11, 1991, and a re-circulation ballot is tentatively scheduled for the April 1991 POSIX meeting time frame.

In IEEE re-circulation ballot, two sets of material are returned to the balloting group:

- the changes made to the document (either a set of changes, or a new document with change bars), and
- the unresolved objections.

IEEE policy does not allow the balloters' names, companies, or company locations to be returned with the unresolved objections packet; to maintain anonymity, ballot comments are numbered, and individual balloters notified of their own ballot comment numbers. (IEEE and ANSI do maintain balloters' names, companies, and company locations to detect corporate ballots, where and if they occur.) The balloting group gets at least ten days to review the re-circulation ballot, though they can be given more time if the size of the re-circulation material and the document being balloted warrant it.

Miscellany

Eight Next Generation Computer Resources (NGCR) representatives gave working-group participation quite a boost. Although NGCR people have the bond of all being NGCR representatives, they are not employed by a single employer, but are from all over the United States, and they possess individual interests and strengths. In the past, our core group has only been about a dozen people, so we are pleased by NGCR's interest and participation, and eager to work with them.

In April 1990, David Emery went to Sweden, to meet with the Ada 9x committee group dealing with secondary standards and setting priorities of those standards. Secondary standards are those standards not contained within the language itself (i.e., not in the Ada Language Reference Manual). POSIX was a very high priority secondary

;login: 15:6

standard. The next Ada 9x committee meeting will be at the SIGAda meeting in Los Angeles in August. Dave is heading a panel presentation on the P1003.5 Binding at this meeting. The chapter authors will also be a part of this panel.

At July POSIX meeting, P1003.5 expressed its special thanks to Dave for his better-than-excellent job as our Technical Editor. He has contributed significant time (much of it his own) and effort to the P1003.5 work, and we appreciate it.

Report on ANSI X3B11.1: WORM File Systems

Andrew Hume <andrew@research.att.com> reports on the July 17–19, 1990 meeting in Murray Hill, NJ:

Introduction

X3B11.1 is working on a standard for file interchange on write-once media (both sequential and non-sequential (random access)): a portable file system for WORMs. The fifth meeting was held at Murray Hill, NJ on July 17–19, 1990. We adopted a working paper and set to work on a list of issues suggested by the chair.

Data Compression

Despite the huge capacities of WORM disks, people always want more. Data compression is an easy way to supply more, and on current machine architectures, probably can speed data access by trading CPU cycles for I/O bandwidth. Its main problem is that you need to support more than one algorithm and thus, you need some way to specify algorithms. This is a purely administrative issue, but luckily, it appears that X3 may soon act as a registry for compression algorithms (driven by the need to register compression algorithms for IBM 3840 cartridge tape work in X3B5). (How does this fit in with the rumblings about *compress* from POSIX.2? I'm not certain. I think part of becoming part of the register means giving up patent rights or allowing liberal licensing, but maybe not. After all, the CD formats are now an ISO standard, but I still think you have to be licensed to make them.)

Path Tables and Extended Attributes

Path tables were removed from the working paper. We agreed to support hard and symbolic

links. The next question was how to handle “secret” files: files primarily intended for system use. Examples might include the file describing free space, associated files (like the resource fork of a Macintosh file), and extended attributes (of a Microsoft s-1HP file). We agreed that the latter two cases should be handled by regular files that probably are not in the directory tree but are pointed to by the “inode” for a file. (Note that this implies there is a way to scan all the files in a volume set without traversing the directory tree(s), analogous to running down the inodes in UNIX.)

Given this, we have decided to support extended attributes as a “secret” or system file (and probably include pointers to things like resource forks as those attributes). This also gives us an extensible way of handling non-standard or non-essential inode fields. One of the important tasks remaining is to decide which fields are more-or-less mandatory (such as modify time, owner) and which can safely be pushed off into the extended attributes (access control lists, file valid after date). Please send us your suggestions!

Space Allocation and Management

We agreed that we have to support preallocating space for files, freeing some or all of that space and then reusing that space for other files. After much discussion about extent lists and bit maps, we compromised on a scheme based on extent lists (the details to be worked by the working paper editor). The idea is that the free space is described by an extent list (of small but specifiable size) of the “best” (probably largest) free spaces, and if this overflows, “worst” free spaces are added to a system file representing all the free spaces not in the above extent list.

Checksums

It was decided that all system data structures would include a 16 bit checksum (CRC-16). We anticipate that most errors would be transient (cabling or memory) and not be media errors.

Multi-Volume Sets

I had thought the last meeting had settled just about all the questions about multi-volume sets, but I was wrong. It took most of a day to agree on these.

- You have to have the last volume in order to grok the whole volume set (access any/all of the directories and files).
- You can extend volume sets at any time. This and the last item taken together imply the existence of “terminal” volumes (which can act as master volumes of a volume set) and “nonterminal” volumes (the rest). For example, if I extend a single-volume volume set by two volumes, then volumes 1 and 3 are terminal and volume 2 is not.
- You can extract file data from any volume by itself. This is meant only for disaster recovery (I dropped the master volume down the stairwell) and doesn't imply any requirements on directory tree information (much as *fsck* restores unattached inodes to /lost + found).
- Volumes can refer to data (say, extents) on other volumes (both earlier and later volumes). Preallocated space on any volume in a volume set can be returned for future reuse.
- The address space of logical blocks for the volume set will be 48 bits; 16 bits for the volume number and 32 bits for the logical block number within a volume. Media can be big (200GB helical scan media exist now) so 32 bits may seem barely big enough, but in such cases you can use a big logical block size. For example, a logical block size of 16KB implies a limit of 64 terabytes per volume; this should be ample for a few years.

Defect Management

We spent a lot of time on this and learned a lot, but basically put it off to the next meeting. What we mean by “defect management” is “How do we deal with write errors from the file system's point of view?” (We ignore the disk controller and the device driver, both of which do some unknown amount of more-or-less transparent error management.)

We discussed the “sane” approach: insert a layer between the file system that handles errors, allowing the file-system code to assume an error-free interface. This apparently good idea is ruled out by slip-sectoring, a (to my mind bogus) technique, which says, “if writing block n fails, then

try subsequent blocks ($n+1$, $n+2$, ...) until we succeed.” Slip-sectoring is mainly used to enhance performance (it does ensure that blocks are more-or-less contiguous), and some disk controllers use it as their error-management technique. (This really screws up your logical address space; it is legitimate for a SCSI disk, your typical error-free, logical-address-space disk interface, to write logical block 5 at physical block 5, then logical block 1 at physical block 4 (1–3 were write errors), then disallow I/O to logical blocks 2,3, and 4 because there is no place to put them — these blocks just vanish!)

As preparation for the next meeting, Don Crouse, who deals mainly with high-end machines like Crays and large IBMs, is writing a position paper on performance, and members of the committee, many of whom are drive manufacturers or integrators, are collecting estimates of error rates we have to deal with. (This matters; I see one bad block out of 100,000, but some people have used drives with a bad block in every 100.) The problem is that WORMs have really slow seek times, and when you are pouring a 50MB/s Cray channel at a set of WORMs, you can't afford to spend 1-2 seconds seeking to the bad block area. I personally think we should just do regular bad-block mapping (like most SMD disk drivers) out of a special system file, and people with performance concerns should arrange to have this space spread over the disk.

Endian-ness

A poll was taken of who really cared which way integer fields were stored; the results were LSB – 1, MSB – 1, Don't Care – 11. It is awkward to specify one of LSB and MSB; this puts half the systems out there at a competitive (performance) disadvantage (though I am skeptical of whether it's significant). Even though we're specifying an interchange standard, the group felt that most interchange would be between systems of the same endian-ness, so we should, somehow, allow native byte order. Accordingly, we agreed that endian-ness will be specified in the volume header (for the whole volume set). In retrospect, I think this was silly; we should have just picked one way. In order that everyone important be evenly disadvantaged, we could have used some byte order like 3-0-1-2 that no one uses.

Finale

The committee is trying to nail down a firm proposal for balloting. We anticipate a substantial amount of change at the next meeting (Oct 16–18 in Nashua, NH) and have reserved time (Dec 11–13, but no place) for an additional meeting so that we can ballot after the following meeting (Jan 29–31, Bay area). We now have a working paper (available by the end of September or so); I think it likely we can meet this schedule, but who knows.

Anyone interested in attending any of the above meetings should contact either the chairman, Ed Beshore (edb@hpgrla.hp.com), or me (andrew@research.att.com, research!andrew, tel: 908/582-6262). I am also soliciting your comments on necessary inode fields and defect management. I will present anything you give me at the next meeting.

Report on X3J16: C++

Mike Vilot <mjv@objects.mv.com> reports on the July meeting in Seattle, WA:

Standard C++ ?

The C++ programming language has been gaining popularity at a remarkable rate (an informal estimate is that the C++ population doubles every nine months). One reaction to the growing popularity has been a call to stabilize the language's definition and achieve some consistency across implementations. C++ is popular enough that larger corporations are considering adopting it as an officially endorsed development language, but some cannot make such a move unless the language is defined by a standard.

For these and other reasons, the ANSI secretariat agreed to establish the X3J16 committee to formulate a standard for C++. Dmitry Lenkov, of HP, made the official proposal and serves as chairman of the committee. To date, X3J16 has met three times: an organizational meeting last December, the first technical meeting in March to get organized, and a meeting in July to really get started.

The December meeting was purely administrative: over 50 attendees received lectures and tons of paper on X3 rules and procedures. The highlight of the day was an invited presentation by

Bjarne Stroustrup on "the spirit of C++." The transcript is available as committee document X3J16/90-0022 or from Greg Comeau at Comeau Computing, 91-34 120th Street, Richmond Hill, NY 11418, (718) 849-2355.

March meeting

AT&T hosted the meeting in New Jersey. Most of the week was spent on administrative matters, while the group got organized and accustomed to The Bureaucratic Way. Since most of the members are engineers, the highlight of the week was the evening technical sessions on implementing exception handling for C++. (The week was sort of a mini-USENIX conference, as most members had gone without a substantial C++ gathering since the October '88, Denver conference.)

The week's major activities were discussing and preparing a goals document, and describing the committee's activities and priorities.

Goals

Here is a brief outline of the goals document, which is available as X3J16/90-0023:

- Base documents: C++ Reference Manual, ANSI C (ANS X3.159–1989), ISO C when available.
- Standardize syntax and semantics of the language as a token sequence without the presence of preprocessing directives.
- Define and standardize a minimum set of C++ libraries, their contents, and interfaces.
- Standardize elements of a C++ environment.
- Consider proposed major changes: parameterized types and exceptions.
- Ensure that the standard is suitable for the international community.
- Ensure a very high level of compatibility with ANSI C.
- Establish coordinating liaisons with X3J11 (ANSI C) and Numerical C Extensions Group.
- Produce two deliverables: draft proposed standard and rationale. Priorities:
 - clear & unambiguous
 - C++ reference manual
 - other base documents
 - consistency
 - user/implementer experience

- portability, efficiency, expressiveness
- ease of implementation (including translation to C)

There was some confusion over the multiple base documents. Most members had seen the AT&T C++ version 2.0 reference manual, but in preparation for standardization, the language and its reference manual had suffered a number of subsequent, small changes. AT&T made the 2.1 reference manual available to X3J16; it was essentially the text of the book *The Annotated C++ Reference Manual* by Margaret Ellis and Bjarne Stroustrup.

My naive suggestion to remove the ANSI C standard as a base document in favor of a single base provoked the most intense and emotional discussion of the week. At stake was compatibility between C++ and C.

While most members of X3J16 feel that the existence of a separate committee implies the standardization of a new language, some former members of X3J11, which just finished the C standard, want to eliminate any and all incompatibilities with C. (There was even a threat to sabotage the C++ standard in balloting if they are not removed.)

This issue is obviously important and has two sides. Make your preferences known to the committee. For detailed reference material, both "C++: As Close as Possible to C — But No Closer," (Bjarne Stroustrup and Andy Koenig, *The C++ Report*, 1(7), 1989) and Chapter 18 of *The Annotated C++ Reference Manual* document and explain differences and incompatibilities between the languages as they stand today.

Focusing on a language without preprocessing directives continues the de-emphasis of the C preprocessor. This is particularly favored by C++ vendors looking into more powerful development environments.

[Editor: Admittedly, improper preprocessor use can sink us in deep and dirty bath water, but let's make sure to save the baby. When writing portable C, I personally find #ifdefs extremely valuable; I suspect they will remain valuable in C++, and I would hate to see the working group neglect this valuable porting tool.]

The libraries effort includes asking what to do about the ANSI-C library, and investigating what can be standardized in a more C++-like approach.

The environment work addresses the linking and executing of C++ code with non-C++ code (i.e., linkage and program execution models), rather than development environment tools.

There are thousands of suggested "improvements" proposed as extensions to C++, but there is consensus on two named in the goals document: parameterized types and exception handling. Their proposals are detailed, and both have been implemented (in some form) in a few C++ implementations.

The emphasis on international concerns reflects the lessons learned from the difficulties of C standardization. X3J16 has some fences to mend, particularly in the international community. Rather than waiting until the last minute to spring a standard on the ISO, the C++ committee is involving itself with the international community right from the start.

July meeting

Microsoft, Inc. hosted the second meeting in Seattle. Sub-groups focused on the key topics listed in the goals statement from the March meeting, and reported their progress here.

International Concerns

Steve Carter, of Bellcore, presented the major International Concerns (particularly character sets and formal specification) and asked the other groups to work on these issues. He also suggested various sites overseas where future X3J16 meetings could help cooperation with international standardization efforts.

Editorial

Jonathan Shopirio, of AT&T, presented the Editorial group's proposal for organizing and formatting the standard. He is also working on an abstract machine model and a way to define the semantics in the standard precisely and consistently.

Formal Syntax

James Roskind, an independent consultant, presented the work of the Formal Syntax group. He has developed (and published on the net) a yacc-able grammar for C++, and is concerned about ambiguities in the the language. Most of the discussion was spent trying to discover whether C++ can (or should) be made LALR(1).

Core Language

Andy Koenig, of AT&T, presented the Core Language group's work. Initially, they identified and classified difficulties in the working document.

Environment

John Vasta, of HP, presented the work of the Environment group. A key issue addressed by this group is the interaction of C++ with other programming languages. Among the important topics are linkage of C++ and non-C++ translation units, especially the construction and destruction of static C++ objects.

Libraries

I presented the Library group's work. There were many suggestions, from both inside and outside of the committee. (Interested outside suggesters were James Coggins, Keith Gorlen, and Doug Lea, who have each developed large C++ libraries.) A few people noted similarity with topics covered by other standards (notably POSIX). Initially, the library group will focus on a few commonly-used components. Parameterized types and exception handling will significantly effect the way we design libraries in C++.

Language Extensions

Bjarne Stroustrup, of AT&T, presented the work of the Extensions group, which was by far the most active. The technical sessions presented experience with implementation and use of the template facility.

The most active and emotional debate of the week was on exception handling, which discussed the proposal outlined by Andy Koenig and Bjarne Stroustrup in their paper "Exception Handling for C++" presented at the USENIX C++ Conference in April. Martin O'Riordan, of Microsoft, and Mike Miller, of Glockenspiel, presented arguments in favor of extending the current proposal (which defines termination semantics for exceptions) to include resumption semantics. Andy and Bjarne explained their reasons for not including resumption — the most important being the complexity and cost of implementation.

To their credit, the group worked hard to find a proposal that provided both kinds of exceptions

with acceptably small time/space overhead. However, at the end of the week, Bjarne declared the debate deadlocked and refused to impose his proposal while substantial disagreement remained. This is another topic where you should make your opinions heard.

C Compatibility

Mike Miller presented the work of the C Compatibility group. Tom Plum, of Plum-Hall, produced a list of every section of the C++ reference manual that was not C. Much of the group's near-term activity will be devoted to explaining the many items on the list.

The Seattle meeting produced tangible progress on the language standard. X3J16 voted to accept the proposed document outline and format. They also agreed to incorporate the template proposal (the text from Chapter 14 of *The Annotated Reference Manual*, minus the annotations — it was literally a scissors-and-tape job). We hope C++ vendors will regard templates as now officially in the language, and provide users an opportunity to work with this feature.

Next events

A few substantial issues lie ahead. The next meeting should see some resolution on the exception proposal. We should see some progress on the review of language ambiguities and inconsistencies, and have some idea of how difficult it will be to ANSIify the document. We should also see some specific proposals on library contents. The most substantial will be a simplified version of *iostreams* by Jerry Schwarz of Stardent.

Our target date for delivering a draft standard is the end of 1992. X3J16 meets three times per year. The next three meetings (and their hosts) will be:

- November 12-26, Cupertino CA (Hewlett Packard)
- March 11-15, Nashua NH (Digital)
- June 17-21, Lund Sweden (Lund Institute of Technology)

Membership on an X3 committee is open to any individual or organization with expertise and material interest in the topic addressed by the committee. The cost for membership is \$250. Contact the chair or vice chair for details.

Chair: Dmitry Lenkov
HP California Language Lab
19447 Pruneridge Avenue MS 47 LE
Cupertino, CA 95014
(408)447-5279
FAX (408)447-4924
email dmitryhpda@hplabs.hp.com

Vice Chair: William M. Miller
Glockenspiel, Ltd
P.O. Box 366
Sudbury, MA 01776-0003
(508)443-5779
email wmmiller@cup.portal.com

Report on U.S. TAG to ISO/IEC/JTC1/ SC22 WG15

Susanne Smith <sws@calvin.wa.com> reports on the July 19, 1990 meeting in Danvers, MA:

Overview

Before you ask, ISO/IEC/JTC1/SC22 WG15 is ISO POSIX. The U.S. TAG is the United States Technical Advisory Group, which formulates the U.S. position on WG15 issues, and chooses the members of the U.S. delegation to the international WG15 meetings.

This meeting began at 8:00 A.M. and ended before noon. This must be a record — not just for the TAG, but for any standards group meeting. There were three major business items:

- language independence,
- document circulation procedures, and
- rapporteurs.

ISO POSIX: Winners and Losers

The vote for 9945-1.2 (1003.1a draft 5) was unanimously in favor without substantive comments. If all goes well there just may be an IEEE version of 9945-1 available in Seattle.

My last report mentioned the formatting problems with the 9945-1 document. The TAG had decided to request the formation of an ad hoc committee in Paris to try to resolve these problems. WG15 resolved to instruct the WG15 convener, Jim Isaak, to request written editorial requirements from the ITTF (formerly the Central Secretariat) and IEEE, and forward these to SC22. The emphasis here should be on written requirements.

WG15 refused to register 1003.4, real-time extensions, as a CD (committee document, formerly DP, draft proposal) because it is not a language-independent specification. They were also concerned that the standard might have to change once there is a language independent version of 1003.1.

1003.5, Ada binding, and 1003.9, FORTRAN binding, suffered a similar fate for different reasons. 1003.5 and 1003.9 were held off until at least the October WG15 meeting because G15 had not seen the 1003.5 and 1003.9 documents, and were reluctant to register something they hadn't seen before. And again, they were concerned that these standards might have to be rewritten once there is a language independent version of 1003.1.

Administrivia

Skip to the next section if you're easily bored or just not interested in bureaucracy. Why, you ask, was WG15 being asked to register something they had not seen before? Here are the steps that have to be completed before a document gets circulated:

- The committee and SEC approve its release.
- The TAG approves its circulation.
- The committee chair delivers the document to the TAG chair, Donn Terry.
- The TAG chair forwards the document to the WG15 convener, Jim Isaak.
- The WG15 convener distributes the document.

1003.5 and 1003.9 were approved by the TAG for circulation to WG15 during the April meeting in Snowbird. This left six weeks for the documents to be circulated and read. The problem was that the TAG chair did not receive the documents in time to have them circulated before the meeting. To avoid this problem in the future, the TAG is going to ask the SEC to assign an action item to the committee chair so that there is a method to track this task.

In other news:

- The TAG procedures were entered and marked up, and will be included in the next mailing.

Are You Ready for UNIX in VDM?

We cannot delay language independence for 1003.1 any longer. It's now really holding up international progress on important POSIX efforts. But what format or technique should we use? ISO rules seem to require an ISO-standard method, which could restrict us to VDM (Vienna Definition Method), but no one thinks VDM will work. Paul Rabin and Steve Walli have been working on a method, but the TAG worries that a non-standard method will create problems like those we've suffered through with document formats (see last TAG report). In order to avoid rejection later we will circulate the new method in SC22 and WG15 for review and comment. To make this circulation useful, Donn Terry is listing specific questions for SC22 and WG15 to answer.

[Editor: I believe that ISO rules only restrict us to VDM if we produce a formal definition, i.e., something from which one could do correctness proofs. Of course, rules and politics are not always the same thing and using VDM might help grease the skids. Still, we should stop and ask if not using VDM would hold us up any more than using VDM.]

The TAG will also ask the WG15 convener to schedule an ad hoc meeting on language independence during the October WG15 meeting to help move it along.

Rap, a-rap, a-rap, they call me the rapporteur.

Rapporteurs are technical experts on specialized aspects of a particular standards effort. Their scope is usually broader than an individual standard, and they usually coordinate efforts in several standards bodies. WG15 has three rapporteur groups, one each for conformance, internationalization, and security. We send a representative to each.

The conformance-testing rapporteur group will be looking at 1003.3 draft 12 (conformance testing), and the OSF-UI-X/Open Phoenix project as potential base documents for the ISO 9945-series documents. The Phoenix project is developing a conformance-testing platform. We will not have to decide whether we want to submit 1003.3 as a new work item in this area until 1991.

Ralph Barker asked that UniForum be allowed to send him and one UniForum Internationalization Technical Committee member to the next internationalization rapporteur group meet-

ing. This person would be subject to subcommittee approval but selected by UniForum. Worry about the fact that the TAG would not choose this person evaporated when it became clear that Donn Terry would continue as internationalization rapporteur and that the UniForum members would just be an addition.

The TAG appointed Al Weaver security rapporteur to fill the vacancy Terry Dowling left when he resigned in January.

Summary

The most important development is that the synchronization proposal discussed in the last report is already dead. This proposal was to have fed balloting responses from IEEE into WG15, and vice-versa, allowing WG15 approval to follow on the heels of IEEE approval. Now, while the IEEE is advancing, everything in WG15 is blocked by 1003.1 language independence.

Report on NIST Shell-and-Tools FIPS Workshop

Donald Lewine
<lewine@cheshirecat.webo.dg.com> reports on the September 6, 1990 meeting in Gaithersburg, MD:

The Federal Government publishes Federal Information Processing Standards (FIPS) for use in buying and using computers. One set of FIPS deal with systems with "POSIX-like interfaces." The government will purchase about \$17 Billion worth of POSIX systems in FY91. Standards let the government avoid vendor-specific requirements like UNIX or SVID. The theory is that the larger the number of vendors that can meet the specification the lower the cost to the taxpayer. Whether that's true or not, using standards makes it harder to protest a purchase decision.

On September 6, the National Institute of Standards and Technology (NIST) held a workshop to gather input from industry and federal agencies on the wisdom of adopting Draft 9 of the IEEE Standard for POSIX Shell and Utility Application Interface (P1003.2) as a Federal Information Processing Standard (FIPS).

The meeting was attended by about a dozen system vendors and about half that many federal agencies.

Roger Martin of NIST opened the meeting with what was to be a three-minute introduction. NIST's agenda was to collect specific comments on the FIPS as printed on Page 23959 of the Federal Register. The vendors' agenda was to get NIST to give up the idea of adopting a FIPS until after the IEEE standard is final. Not surprisingly, given this clash, Roger's opening remarks ran over by a factor of 20.

Here is NIST's case for adopting a FIPS based on POSIX.2/D9:

- The federal government is going to purchase about \$17 billion worth of systems with "POSIX-like interfaces." NIST wants to give the agencies as much help as possible. Draft 9 is a good enough standard to serve this purpose.
- It takes about a year to get a FIPS adopted. If POSIX.2 is not approved until mid-1991, a FIPS based on draft 9 will have a significant lifespan.¹
- If NIST were to publish a FIPS, it would accelerate the production of the P1003.2 standard. (just as FIPS 151 accelerated IEEE 1003.1-1988).
- No agency is going to be stupid enough to demand draft 9 if a vendor can supply a system conforming to a later draft or to the final standard, so the FIPS will do no harm. (This was hotly debated.)

After that introduction, and before the next attack on Roger Martin, Sheila Frankel and Rick Kuhn described the technical content of the FIPS. Basically, the idea is to adopt draft 9 minus the parts that might change. There are about 25 items that may change.

Roger Martin came back for another round of target practice. He went over the general policy of NIST, which is to adopt standards from outside and at the highest possible level. The levels are, highest to lowest:

- International Standards
- National Standards
- Draft Standards
- de facto Standards

NIST could be convinced to change from POSIX.2/D9 to POSIX.2/D10. Here are the factors it will consider:

How much delay is introduced (Three months may be OK. One year is unacceptable.)

Is Draft 10 that much better than Draft 9? Is this just a delaying action?

Shane McCarron, of UNIX International, made a great speech pointing out how much wasted effort would occur if every vendor had to rush out and implement POSIX.2/D9. The NIST people seemed shocked at how different POSIX.2/D9 is from existing practice.

[Editor: See Randall Howard's POSIX.2 report for some examples of just how different Draft 9 is from Drafts 8 and 10.] Nevertheless, the argument seemed to fall on deaf ears, because NIST claimed that a promise to meet the FIPS should be good enough, and everyone can still wait for AT&T USL to write the code.

It was pointed out that Congress did not allocate enough funding for NIST to do much testing for POSIX.2 conformance. This means that vendors will have to "self certify" and coverage may vary. After some discussion this item was placed into the "write your representative" category, because only Congress can allocate the money.

NIST pointed out that they are under a great deal of pressure to "advise" federal agencies who want to move to open systems. A large percentage of RFPs for POSIX-like systems will be coming from groups who know nothing about such systems. Vendors were worried that this "advice" would end up in court cases and be read by judges as "regulations."

In my opinion, NIST is going to go ahead and publish a flawed FIPS in the belief that it will drive the IEEE to pick up the pace of POSIX. The government has a burning need for a standard, they find it politically unacceptable to use UNIX System V as that standard, and they strongly prefer action over waiting for the IEEE.

1. Just because the IEEE approves a standard *does not* make it a Federal Information Processing Standard. The feds still have to go through the entire legal process of publishing it in the Federal Register, collecting comments, writing responses to those comments, and getting it signed by the Secretary of Commerce. This process takes about a year even for a null standard.

Recent Standards Activities

Jeffrey S. Haemer <jsh@ico.isc.com>.

Summer-Quarter Standards Activities

This editorial addresses some of the summer-quarter standards activities covered by the US-ENIX Standards Watchdog Committee.¹ In it, I've emphasized non-technical issues, which are unlikely to appear in official minutes and mailings of the standards committees. Previously published watchdog reports give more detailed, technical summaries of these and other standards activities. If my comments encourage you to read one of those earlier reports that you wouldn't have read otherwise, I've done what I set out to do. Of course, on reading that report you may discover the watchdog's opinions differ completely from the ones you see here. As watchdog editor I just edit the reports, I do not determine their contents. The opinions that follow, in contrast, are mine.

Profiles

There's an explosion of activity in the profiles world, bringing with it an explosion of problems, and dot zero, the POSIX guide group, is at ground zero.² The first problem is, "What's a profile?" Everyone has a rough idea: it's a document that specifies an application-specific set of standards (or pieces of standards). The best informal illustration I've heard is from Michelle Aden, of Sun Microsystems. Imagine, she says, you have to write a guideline for buying lamps for Acme Motors. You might require that the lamps have ANSI-standard, three-prong plugs, accept standard one-way, hundred-watt bulbs, have cords no shorter than five feet, and stand either two to three feet tall (desk models) or five to seven feet tall (floor-standing models). This combination of pointers to standards, additional specifications, and detailed options, which gives purchasing agents guidelines to help them make choices without tying their hands to a specific

1. The introduction to this series of reports provides a general overview of the committee itself.

2. I use "dot zero" to refer both to the P1003.0 working group and to the document it's producing. These are common conversational conventions among standards goers, and which of the two I mean is usually obvious from context.

vendor, is a profile — in this case, an Acme Motors lamp profile. Dot zero now sees itself as a group writing a guide to help profile writers pick their way through the Open-Systems' standards maze.

But that rough agreement is as far as things go. And the standards world is never informal. For "profile" to graduate from a hallway conversation buzzword to an important organizing principle, it needs a precise definition. And since there are already four groups writing profiles — real-time, transaction processing, multiprocessing, and supercomputing — TCOS needs to figure out what a profile is quickly. ISO already has IAPs (International Applications Profiles). The ISO document TR 10K describes these in detail. Unfortunately, TR 10K was developed for OSI-related profiles and shows it. Cut-down extracts of the standard appear in the document. Someone needs to define a PAP (POSIX Application Profile).

But that's just the first problem. Even thornier is "What does it mean to say that something conforms to the POSIX transaction-processing profile?" If I want to write assertions for a profile or tests to verify those assertions, how do I do it? Does it suffice to conform to the individual components? What about their interactions? The first principle of management is "If it ain't somebody's job, it won't get done." Dot zero has done such a good job of promoting **The Profile** as an organizing principle for addressing standards issues that people are beginning to press dot zero for answers to questions like these. Unfortunately, that's a little like killing the messenger. It's just not dot zero's job. So the fundamental profile question is "Who's in charge?" Right now, I think the question sits squarely, if uncomfortably, in the lap of the SEC (the Sponsors Executive Committee), which oversees the IEEE's operating-systems activities.

In the meantime, the various working groups writing profiles are making headway by just trying to define profiles and seeing where they get stuck. Dot twelve, the real-time profile group, is busily making various sorts of tables, to try to find a reasonable way to specify the pieces that make up a profile, their options, and their interactions. Dot ten, the supercomputing profile group, is seeking an overall structure for a profile document that

makes sense. Dot eleven, the transaction-processing profile group, is trying to steal from dots twelve and ten, an important test of the generality of the other two groups' solutions. Dot fourteen, the multiprocessing profile group, isn't far enough along to make theft worth their while, but will eventually provide a second generality test. Think of it as a problem in portable ideas.

Will I Win My Beer?

In my last editorial, I announced a beer bet with John Gertwagen over whether threads will ballot and pass before the base dot-four (real-time) ballot objections are resolved. I'm still betting on threads, but it looks like the bet is still anyone's to win. Some folks assure me that I'll win my beer handily, others say I don't have a chance.

At the summer POSIX meetings in Danvers, Massachusetts the dot-four chair, Bill Corwin, challenged the threads folks to come up with a ballotable draft by the end of the week, and they very nearly did. (I hear complaints from some quarters that the vote to go to ballot was 31 to 7 in favor, and that attempts to move to balloting were only blocked because of filibusters from those opposed.) On the other hand, technical reviewers are now resolving ballot objections to the base with machetes. They've thrown away asynchronous events altogether and have discarded real-time files and adopted the flmmap model that the balloting group suggested.³

Innovation

Hoare once said, "One thing [the language designer] should not do is to include untried ideas of his own." We have followed that precept closely. The control flow statements of Ratfor are shamelessly stolen from the language C, developed for the UNIX operating system by D. M. Ritchie. — Kernighan and Plauger.⁴

Should standards groups just standardize existing practice or should they be solving known

problems? And if they solve known problems, how much innovation is allowed? Shane McCarron's September UNIX/Review article⁵ uses the real-time group, dot four, as a focus for an essay on this subject. His thesis is that standards bodies should only be allowed to standardize what's boring. I've already seen John Gertwagen's reply, which I assume will be printed in the next issue. I find myself agreeing (and disagreeing) with both and recommend you read them.

This battle will rage brighter in some of the groups less far along, but sporadic fighting still breaks out in the shell and tools group, dot two. Right now, collation and character classification are seeing a lot of skirmishing. Some want to stay relatively close to the existing practice, while others want to grow a mechanism to deal with the Pandora's box of internationalization. My favorite current example, though, is *make*. Bradford's augmented *make* is almost a decade old. Stu Feldman's original is a couple of years older than that. That decade has seen a number of good *make* replacements, some of them wildly successful: Glenn Fowler's *nmake* has virtually replaced *make* for large projects in parts of AT&T. Still, many of these upgrades maintain the original *make* model,⁶ just patching up some of *make*'s more annoying craters and painting over its blemishes. At this point, there is real consensus among *make* augmentors about some patches. Most upgrades expand *make*'s metarules. For example,

```
.c.o:
$(CC) $(CFLAGS) $<
```

might become

```
%.c : %.o
$(CC) $(CFLAGS) $<
```

Not much of a change, but it also gives us

```
s.% : %
$(GET) $(GFLAGS) -p $< > $>
...
```

in place of the current, baroque

```
.c.o:
$(GET) $GFLAGS) -p $< > $>
...
```

3. Dot four's real-time files are currently a part of the supercomputing profile. If they disappear from dot four, they may reappear elsewhere. Dot four has moved from "design by working committee" to "design by balloting committee."

4. Kernighan, Brian and Peter Plauger, *Software Tools*, Addison-Wesley, 1979, p. 318.

5. McCarron, Shane, "Commodities, Standards, and Real-Time Extensions," *UNIX Review*, 8(9):16-19 (1990).

6. Fowler, Glenn, "A Case for make," *Software — Practice and Experience*, 20: S1/35-S1/46 (1990).

Make's successors don't agree on syntax, but they all agree that "" rules are the wrong solution to a real problem. Should dot two standardize a newer solution? Existing-practice dogmatists would say, "No. It's not *make*." Here's a place I say, "Yes, if we can do it in a way that doesn't break too many makefiles." The prohibition should be against untried ideas, and I don't see those here. A year or so ago, Stu Feldman (*make*), Glenn Fowler (*nmake*), Andrew Hume (*mk*), and a handful of other *make* luminaries presented a proposal to add four extensions to dot two's *make*. Not one is yet in the draft. I hope that changes.

SCCT Faces Serious Problem

At Danvers, the testing group, dot three, worked with dot two on test assertions to try to avoid the kinds of problems created by the P1003.1 test assertions, which dot one had no input into until the assertions were in ballot.

A side effect of the collaboration, which is taking place before dot two is finished, is that it may reveal that parts of dot two are imprecise enough to require a rewrite. Dot two, draft eight had around four-hundred ballot objections, draft nine saw fewer than half that number. There was hope that draft ten would halve that again, bringing it within striking distance of being a standard.⁷ The assertion work may point out and clear up rough spots that might otherwise have escaped the notice of battle-fatigued balloters. (Paradoxically, NIST, which is heavily represented in dot three and painfully familiar with dot two's status and problems, is currently pushing for a shell-and-tools FIPS based on the now-out-of-date draft nine.)

The exercise of trying to construct assertions for dot two before it goes to ballot may bring some new testing problems into focus, too. Before I explain what I mean, I'll give you a little background.

The POSIX effort has outgrown dot three, which did test assertions for dot one and is in the process of constructing test assertions for dot two. Dot three has, at most, a couple of dozen mem-

bers, and the document for dot two alone may swell to one- or two-thousand pages.⁸ If dot three were to continue to do all test assertion work, it would have to produce a similar document for at least a dozen other standards.

Reacting to this problem, the SEC created a steering committee, the SCCT, to oversee conformance testing. The committee's current plan is to help guide standards committees to write their own assertions, which will be part of the base document. Test assertions, like language independence, are about to become a standards requirement (a standards standard).

With this change, the current process — write a base document, evolve the base document until it's done, write test assertions for the result, evolve the test assertions until they're done — would become: write a base document with test assertions, then evolve the base document modifying the test assertions as you go. A sensible-enough idea on the surface, but after the joint dot-two, dot-three meeting I have questions about how deep that sense runs.

First, does it really make sense to write assertions early? Working-group members should be exposed to assertion writing early. When working-group members understand what a testable assertion is, it's easier to produce a testable document. Still, substantive, major draft revisions are normal, (see the real-time group's recent ballot, for example) and keeping test assertions up-to-date can be as much work as writing them from scratch. This meeting saw a lot of review of draft-nine-based assertions to see which ones had to change for draft ten.

Second, if you make the assertions part of the standard, they're voted on in the same ballot. Are the same people who are qualified to vote on the technical contents qualified to vote on the test assertions?

Third, writing good assertions is hard, and learning to write them takes time. How eager will people in working groups be to give up time they

7. It didn't reach that goal. Keith Bostic tells me he submitted 132 objections himself.

8. Any imagined glamour of POSIX meetings fades rapidly when one is picking nits in a several-hundred-page standards document. When asked where the next meeting was, one attendee replied, "some hotel with a bunch of meeting rooms with oversized chandeliers and little glasses full of hard candies on the tables."

now spend writing and revising document content in order to do assertions?

Fourth, is the time well-spent? Not everything merits the time and expense of a standard. If only a small number of organizations will ever develop test suites for a particular standard (with none being a special, but important case) does it make sense for folks to spend time developing standards for those test suites? Wouldn't it make more sense to develop it after there is a clear need? (This is a perverse variant of the "existing practice" doctrine. Even if you don't think standards should confine themselves to existing practice, does it make sense to innovate if there's *never* going to be an existing practice?)

Stay Tuned for This Important Message

If you haven't yet had the pleasure of internationalizing applications, chances are you will soon. When you do, you'll face messaging: modifying the application to extract all text strings from external data files. The sun is setting on

```
main ()
{
    printf("hello, world\n");
}
```

and we're entering a long night of debugging programs like this:

```
#include <stdio.h>
#include <nl_types.h>
#include "msg.h" /* decls of catname(), etc. */
#define GRTNG "hello, world\n"
nl_catd catd;

main ()
{
    setlocale(LC_ALL, "");
    catd = catopen(catname(argv[0]), 0);
    printf(catgets(catd, SETID, MSGID,
        GRTNG));
    catclose(catd);
    exit(0);
}
```

This, um, advance stems from a desire to let the program print

chào các ông

instead of

hello, world

when *LANG* is set to "Vietnamese."

Most programs use text strings, so the system services interface group, dot one, has been thinking about portable library calls to supply such strings and portable formats for the files that contain them.

Actually, "re-thinking" is probably more accurate than "thinking about." 1003.1a Draft 9, specified a design by the UniForum Technical Committee on Internationalization. At Danvers, X/Open counter-proposed a variant of its existing XPG3 specification, arguing that the X/Open scheme may have problems but it also has users, while the UniForum proposal is still in the laboratory. (It brings to mind the apocryphal story of Stu Feldman's wanting to improve the design of *make*, but feeling he couldn't because he already had seven users.) Someone from Unisys also brought a proposal, different from both UniForum's and X/Open's.

That no one even showed up to defend the UniForum proposal shows that there is something wrong with standardizing messaging. In one instance there is enough support for a messaging scheme to get it into the draft standard; in the next, there's none at all. In the end, the working group agreed that a messaging standard was premature and that the free market should continue to operate in the area for a while.

Given the relative sizes of the organizations concerned, this outcome probably sticks us with the X/Open scheme for a while, which I find the ugliest of the lot. Still, it's not a standard, and there's room for innovation and creativity if we're quick about it. The "existing practice" criterion is supposed to help avoid a requirement for massive, murderous source code changes. We should be looking for the messaging scheme that doesn't require changes in the first place, not the one with the most existing victims.

Language Independence Stalls ISO Progress

Internationally, 1003.4 (real-time), 1003.5 (Ada bindings), and 1003.9 (FORTRAN bindings) are being held hostage by ISO, which refuses to loose them on the world until we come up with a language-independent binding for 1003.1. The question is, who will do the work? "Not I," says dot four, whose travel vouchers are signed by companies caught up in the glamour of real-time and threads; "Not I," say dot five and dot nine, who seldom have even ten working-group members apiece; "Not I," say the tattered remnants of dot one, exhausted from struggling with 1003.1-1988, FIPS-151 and 151-1, and (almost) 1003.1-1990, before any other groups have even a first standard passed. Where is the Little Red Hen when we need her?

Should We Ballot POSIX the Way We Ballot Three-Phase Power?

In the meantime, we progress inexorably toward balloting on several IEEE/ANSI standards. The sizes of the drafts (and several contributors to *comp.std.unix*) raise real questions about whether the IEEE's balloting process make sense for the sort of standards work POSIX is performing. A month or so might be enough to review a few-page hardware standard. But is it enough for the nearly 800 pages in the latest recirculation of dot two? Does it really make sense to review the standard for *grep* in hard copy? Many would like to see longer balloting times and on-line access to drafts. Some argue that the final standard should be available only from the IEEE, both to insure authenticity and to provide the IEEE with income from its standards efforts; even that argument seems weak. Checksums can guarantee authenticity, and AT&T's Toolchest proves that electronic distribution works: I'll bet *ksh* has paid for itself several times over.

“We handed 1201.1 its head and asked it to comb its hair.”

Moving away from POSIX, we come upon 1201.1, still in search of an officially sanctioned mission that the group wants to take on. The group currently has a PAR (charter) to standardize various aspects of X-based windowing, principally the toolkit-level API but any hope of compromise between the OPEN LOOK and OSF/Motif factions died at the winter-quarter Utah meetings. In a moment of responsible behavior, the group recovered by switching to a dark horse: a window-system-independent API that could be implemented on top of either product. Marc Rochkind's XVT, which already allows users to write programs that are portable across several, unrelated window systems including X, the Mac, and MS-Windows, was offered as a proof-of-concept.

While the original charter could probably encompass the new XVT work, the group seemed to feel that this direction change, together with the fragmenting of the original group into separate toolkit, drivability, UIMS, and X intrinsic efforts, required that they ask the SEC for a new charter. (The drivability group has already had a separate PAR approved and is now 1201.2.) The

Milpitas, California, and Boulder, Colorado, to forge a PAR that would meet the SEC's new, stricter standards for PAR approval by the summer Danvers meeting. They didn't succeed.

Most of the problems seem to have been administrative missteps. Some examples:

- Working-group members complained that the Milpitas meeting took place without enough notice for everyone to attend, and issues that had been resolved at the interim meetings were re-opened in Danvers.
- The PAR was so broadly written that at least one technology (Serpent) was advanced as a candidate that almost no one thought should even be considered.
- Some working-group members hadn't even received copies of the XVT reference manual before they reached Danvers.
- Many SEC members appeared not to have seen a copy of the PAR until the afternoon before the SEC meeting, and some saw the final PAR for the first time at the SEC meeting itself.

Many people who weren't familiar with the proposal ended up uneasy about it, not because they'd read it and didn't like it, (they'd not been given much chance to read it) but because a lack of attention to administrative details in the proposal's presentation sapped their confidence in the group's ability to produce a sound standard. After all, standards is detail work. In the end, the SEC tactfully thanked the group and asked them to try again. One SEC member said, “We handed 1201.1 its head and asked it to comb its hair.”

I believe all of this is just inexperience, not a symptom of fundamental flaws in the group or its approach. If 1201.1 can enlist a few standards lawyers — POSIX has no shortage of people who know how to dot all the *i*'s and cross all the *t*'s — and can muster the patience to try to move its PAR through methodically and carefully, I think the group will give us a standard that will advance our industry. If it doesn't do so soon, though, the SEC will stop giving it its head back.

Report on ISO/IEC JTC1/SC22/WG15 (POSIX)

October 23–26, 1990

Orcas Island, Washington, U.S.A.

Dominic Dunlop
The Standard Answer Ltd.
domo@tsa.co.uk

Introduction

Are you a regular reader? I hope so, as this report on the October meeting of Joint Technical Committee 1, Subcommittee 22, Working Group 15, colloquially known as the ISO POSIX working group, seems to be particularly replete with buzzwords, acronyms and jargon. I try to explain these as I encounter them, but since USENIX and Eur-Open (formerly EUUG) have been sponsoring me to produce these reports for almost two years now, some of the explanations are buried in previous editions. For now, you will just have to bear with me; I will take time to explain how we arrived at the current state of affairs in a future column. This one concerns itself mainly with where we are headed, and with the difficulty of actually getting there.

As far as ISO is concerned, POSIX, like Gaul, is divided into three parts. Forget all those proliferating IEEE 1003 POSIX working groups (eighteen of them at the last count), and just think of three standards: IS9945-1 for a definition of the services offered by the operating system; IS9945-2 describing the shell and tools; and IS9945-3, system administration.

The good news is that you can now buy the first edition of the first of these¹. The bad news is that all three ISO standards projects are running into scheduling difficulties. And there is even more bad news if you are an Ada™ fan: in order to ease its own difficulties, the ISO POSIX working group has put a serious road block between your favourite language and an international standard defining how you may use it to access POSIX services. Why did we do this, and why don't we feel bad about it? Read on...

1. From the IEEE, which has agreed to print the world's first combined IEEE/ANSI/ISO standard—on ISO standard A4 paper. Ask for IEEE Std. 1003.1:1990 It will cost you \$52.50 if you are an IEEE member, \$75.00 otherwise. Add \$5.00 for surface mail to Europe. In the U.S.A., call (800) 678-IEEE; elsewhere, +1 908 981 1393. IEEE accepts "major credit cards".

9945-3—System Administration

As you are probably aware, the IEEE P1003.7 working group on system administration has decided that current UNIX administrative tools and practices are sufficiently obsolete, inadequate and diverse that they are not worth standardizing. Instead, the group has elected to define a new, object-based administration scheme which views a system as a collection of objects to be administered, and a network of systems simply as a larger collection of such objects.

Although this approach grafts neatly onto the network administration work which has been going on in the Internet and Open Systems Interconnection (OSI) communities, it will be a while before it produces any results. As we shall see in connection with 9945-2, when ISO delegates responsibility for the development of a standard to another body, as it has done with the POSIX standards, it likes the documents to be in a relatively stable state before they are submitted for use as ISO Working Documents (WES). 1003.7 thinks that it will have something suitable for ISO to start work on by 1992.

Unfortunately, ISO rules state that, unless a project has resulted in a WE within three years of authorization, the authorization stands in danger of automatic withdrawal. The only way out is for a national standards organization participating in the development of the standard to call for a vote on project continuation before the time limit expires. The time limit for the production of a draft for 9945-3 has almost been reached, with no prospect of the deadline being met.

It seems inevitable that the twenty-four countries participating in the ISO POSIX project will be formally balloted as to whether they think that the authorization to develop a system administration standard should stand, despite the missed deadline. This is not a particularly big deal: an examination of ISO's information technology standardization work reveals that around twenty

percent of projects miss one deadline or another. (OSI standards have a particularly poor track record.)

Nevertheless, it is embarrassing when the managerial finger is pointed at one's own project. Already, the special pleading has started; the SC22 Advisory Group, which makes recommendations on policy issues to the ISO programming languages subcommittee, has suggested that "in general, standards developed within SC22 are larger and more complex than most others, and the time limits given in JTC1 directives²... will therefore often be too short."^[1] This may be true—although work elsewhere in ISO suggests that SC22 has no monopoly on large projects. However, it seems to me that the time limits given by the directives cannot reasonably be relaxed; if no visible progress has been made on a project after three years, those involved had better be given an opportunity to ask themselves why, and to consider whether they wish to continue giving their support. I am sure that, if it comes to a vote, the result will favour the continuation of the system administration project. Just don't hold your breath waiting for the final standard.

9945-2—Shell and Tools

The shell and tools standard is not crowding a deadline as closely as is system administration, but is not clear of trouble either. At least we have a committee draft (CD — one step beyond a WE), corresponding to draft 9 of 1003.2, but have failed to move it forward to the next stage, a Draft International Standard (DIS). According to the directives, we have four years in which to do this before serious questions get asked, and the ISO directorate makes a decision about project termination. Although our progress to date has not been rapid, we have some time in hand.

Our first attempt to register the 1003.2 draft as a DIS failed. (See my report on WG15's Paris meeting^[2].) The problem was that, while the technical content of a DIS is supposed to be essentially the same as that which will appear in the recent International Standard (IS), we all knew that the content of 1003.2 was still undergoing rapid and

sometimes radical change. There was no way that draft 9 could have been accepted as a DIS. (The U.S. National Institute for Standards and Technology (NIST) ultimately decided not to base a Federal Information Processing Standard (FIPS) on draft 9 for similar reasons.)

Draft 11 (or later) of 1003.2 will be passed to ISO in January, 1991 (or later), in the hope that it can be registered as a revised CD, and will stand more chance of clearing the remaining hurdles which separate it from IS status. Until this happens, we have a situation described by one normally restrained working group member as a "pure disaster". We are about to suggest that draft 6 of 1003.2A, the User Portability Extension, due early in 1991, be registered as a proposed draft amendment (PDAM) to 9945-2, without having a stable document to amend³!

In this situation, somebody may ask us why we don't just roll the amendment into the next, hopefully more stable, version of the CD. The practical answer to the question is that the IEEE is treating 1003.2 and 1003.2A as two separate documents, and we would prefer to do the same. How much weight such an argument might carry with the ISO secretariat is another question.

9945-1—Operating System Interface

Now that 9945-1:1990 operating system interface definition is an international standard, international standards work on POSIX has reached the end of its beginning. What do we do next? The problem is that we are spoiled for choice. An embarrassing number of the 1003 projects represent extensions to, or restatements of, the services described in 9945-1:

- 1003.1A: A 1003.1 extension draft, covering tweaks such as symbolic links, will be ready for us early in 1991. We shall probably vote to register it at our next meeting.
- 1003.1LI: (Provisional name.) This is the language-independent specification of the services defined by the current 1003.1 standard in terms of their C language interface. It may be ready in late

2. The rule book which guides our every move is *The JTC1 Directives*. It is surprisingly readable, and very clear on general principles and procedures, but seems to be intentionally vague on many details.

3. The UPE to 1003.2 describes interactive utilities for program development, supplementing 1003.2's description of the non-interactive tools used in shell scripts.

- 1991, provided that enough IEEE volunteers can be found to work on it.
- 1003.1C: (Provisional name.) Building on the definition provided by 1003.1LI, these C bindings will correspond exactly to the C interface defined by the current 1003.1. Again, a draft may be ready late in 1991.
- 1003.2: The shell and tools standard defines C language interfaces to regular expression handling, filename expansion, argument string parsing and more. Arguably, these belong in 9945-1. They are also candidates for language-independent specification.
- 1003.4: We have requested that the current draft of 1003.4⁴, real-time extensions to the portable operating system interface, be registered as a PDAM to 9945-1. The first international POSIX standard has only just hit the streets, and already we are trying to amend it!
- 1003.4A: The 1003.4 working group considers that draft 5 of its threads (lightweight process) standard will be ready for submission to ISO at the same time as 1003.4. As yet, we have made no decision to accept it.
- 1003.4B: This is simply a language-independent specification for the services described by 1003.4 in terms of their binding to the C language. The IEEE working group does not know when it will be ready, and we don't yet know when we shall be ready to accept it. The two issues are connected: if we say we want work on it to be accelerated, it is likely to be ready more quickly.
- 1003.5: The Ada description of the portable operating systems interface is well on its way to becoming an ANSI/IEEE standard. Expect to see it in 1992. Sadly, for reasons explored below, 1003.5 is unsuitable as a basis for an ISO standard.
- 1003.6: The security extension to the operating system interface will be ready for us to have a look at in January of 1991, al-

4. That is, the draft current at the time that the ISO secretariat requests ANSI to provide a document for circulation to SC22 and WG15 as a prelude to calling a vote on registration. This will be draft 10, or, more probably, draft 11.

- though it will be a while before it is mature enough for PDAM registration.
- 1003.8: Transparent file access, that is, transparent access by a process hosted on one system to files held by another, is making rapid progress after narrowing down its goals until it identified an achievable target. The IEEE working group expects to have a document suitable for ISO review by mid-1991.
- 1003.9: The FORTRAN⁵ bindings to the operating system interface definition are written in a manner which is more to ISO's taste than the Ada description of the same services, and will be ready for ISO review in late 1990. However, we have elected not to bring it forward to international standards status in the near future. Again, our reasons are explored below.
- 1003.16: This recently-authorized IEEE project aims to produce C language bindings to some future language-independent specification of the POSIX operating system interface. Like Ada and FORTRAN, it is tied up with the whole issue of language independence.

I wrote last time about the background to the language independence debate^[2]. Further discussion and a useful bibliography can be found in^[3]. ISO strongly favours language-independent service specifications, but very few people in the U.S. are interested in writing them. ISO has delegated development responsibility for POSIX to ANSI, which in turn has passed the buck to the IEEE—an organization which ISO cannot officially talk to or aid. As a result, IEEE is saddled with a problem which it is ill-equipped to solve.

At our Paris meeting, we signalled our disappointment with the IEEE's progress towards a language-independent specification for POSIX by refusing to register drafts of 1003.4, .5, and .9. The list above shows that we have relented on 1003.4, but have left .5 and .9 out in the cold.

The difference between this meeting and the last is that they are now definitively out in the cold, and will not be let into the ISO process until

5. Obscure style note: one is supposed to refer to the proposed 1990 version of the language as Fortran; to older versions as FORTRAN. 1003.9 is a binding to FORTRAN 77.

we are very close to having a language-independent version of IS9945-1 for them to bind to. Here, with a few interpolations in square brackets, is the resolution that says why:

Language-Independent Specifications:

Whereas, SC22 AG [the advisory group mentioned above in connection with 9945-2] has recommended that the production of language-independent specifications and language bindings for POSIX be carried out in such a way that it does not delay the standardization of the C language bindings^[1] [Thank you. That's just what most of us wanted to hear.]; and

The production of a language-independent specification corresponding to IS9945-1:1990 and subsequent C language-based amendments, together with a C language binding to that language-independent specification, is required by the Division of Work Item JTC 1.22.21 [A Division of Work Item is an ISO mechanism for splitting an authorised project into several sub-projects]; and

The production of further language bindings to the language-independent specification corresponding to 9945-1:1990 as subsequently amended is ultimately desirable; and WG15 considers that "thin" language bindings (which must be read in conjunction with a service definition) are suitable candidates for standardization, but "thick" bindings (those which incorporate a service definition duplicating and possibly conflicting with the service definition provided by another standard) are not [The terms "thin" and "thick" derive from the number of pages in the document in question. 1003.5 is a "thick" binding, so we do not like it much; 1003.9 is a "thin" binding, but to the C language specifications of the current 1003.1];

Therefore, JTC1/SC22/WG15 requests the U.S. member body [ANSI, which in turn gets the IEEE to do the work] to provide a schedule for the delivery to WG15 and SC22 of 9945-1-related documents which is subject to the following constraints (listed in order of precedence, highest first):

1. The incorporation or development of language independence features shall not

be on the critical path(s) for the production of C language-based documents;

2. The ultimate goal is the production of an extended [extended, that is, by 1003.4, 1003.6, 1003.8...], language-independent 9945-1 and accompanying "thin" binding to the C language at the earliest possible date;

3. Every attempt shall be made to observe JTC1/ISO rules on the bringing forward of amendments, etc., with the need to seek waivers being highlighted if this appears necessary in order to satisfy the constraints above;

4. Language bindings, other than those for the C language, shall not be brought forward to WG15 or SC22 for any purpose other than review and comment before the language-independent 9945-1 has been registered as a DIS; and

5. Where possible in the light of other constraints, C language-based documents shall include an informative annex setting out a language-independent definition of the services defined by the normative body of the document.

The schedule shall identify timeframes for at least the following document circulation and registration milestones:

1. "Thick" C bindings for amendments to 9945-1:1990;

2. Language-independent specifications corresponding to 9945-1:1990 and subsequent amendments;

3. "Thin" C bindings to language-independent specifications corresponding to 9945-1:1990 and subsequent amendments;

4. A combined language-independent 9945-1 and accompanying "thin" C binding to the services that it defines; and

5. "Thin" bindings for further languages to the whole of the combined language-independent 9945-1, or to supersets or subsets of the services which it defines.

I hope that your eyes have not glazed over: public statements of policy get convoluted and legalistic at this level, but all of this verbiage

actually represents a concise description of the problem and what we see as a route to its solution⁶. For the first time, this tells the IEEE exactly what type of document that the ISO working group wants to see, and in which order:

- a. C-based standards first.
- b. Language-independent standards with a corresponding “thin” C binding second.
- c. “Thin” (and only thin) bindings to other languages no sooner than b.
- d. Examples of language-independent specifications (as opposed to definitive standards for them) any time that IEEE can manage to produce them.
- e. All in accordance with ISO rules on the publication of amendments and revisions to standards (we hope).

There was some understandable objection from the U.S. to “micro-management”—if we were defining so many goals, constraints, and checkpoints, why didn’t we just write the schedule ourselves? The answer is that there is still quite a lot of flexibility allowed: the IEEE has a dozen or more documents to bring forward, and it can decide on the ordering and the dates. We just want to know when those dates are, and to disallow certain orderings.

The amount of resources that the IEEE can muster to work on language-independent specifications determines when step b can occur. Does anybody want to volunteer to make it sooner than 1995?

The real victim of our newly-defined policy is Ada. It is clear that there will be an ANSI/IEEE standard for an Ada definition of the POSIX operating system interface, probably in two years. It is now equally clear that, because it will be a thick binding, this standard cannot move forward to gain international status. There may ultimately be a thin Ada binding to a future language-independent 9945-1. It may define an interface identical to that defined by 1003.9, but probably not. We could face the unpalatable prospect of an ISO standard which differs from the corresponding ANSI standard.

6. Although I could be biased: I drafted the resolution.

Why don’t we feel too bad about this? Well, it seems that the main requirement for an Ada POSIX standard comes from within the U.S. 1003.5 will fill this need, and we should not seek to delay it. The need for an international standard in this area is less clear, but we have now given clear guidelines on the form that it should take, just as soon as anybody wants to develop it.

That said, it is clear that we still have much to learn about...

Coordination

One aim of the IEEE and ISO POSIX projects is that the international standards that result should be identical to the corresponding U.S. standards. Another is that ISO publication should not lag behind IEEE publication by too long. IS9945-1 is a benchmark in both cases: by dint of the IEEE agreeing to print for both organizations, content is identical, and publication is simultaneous. This will be a hard act to follow, not least because there are thousands of pages of IEEE drafts in the pipeline, all of which must undergo international review before they can even start going through the three-stage ISO mill which grinds documents into international standards.

It has been the policy of the IEEE not to submit documents to ISO until they reach their first IEEE ballots—that is, until they are reasonably mature and complete. In view of our rejection of 1003.2 draft 9 because we did not consider it mature enough, this seems like a prudent approach. The trouble is that by the time the IEEE considers a document mature, it is also likely to be difficult to change in any significant manner. If we had earlier visibility into the subject matter and approach of the IEEE’s work, we could comment on its future acceptability to ISO. For example, we could have suggested that 1003.5 pursued a “thin” rather than a “thick” binding.

To try and get out of the hole that we have dug for ourselves, we have requested “early visibility” of IEEE draft standards. Seeing standards when they are young and small will also aid international understanding of the larger more mature versions when they appear.

OSCRL

The POSIX project bears a growing similarity to an ancient yet still inhabited castle: some parts

are old and crumbling; others require constant repair if they are to remain habitable; and, all the time, new walls, doors, and towers are being added. 1003.7 even seems to be demolishing a few unsightly outbuildings. Coordination should ensure that nobody builds a wall where someone else wants a door. Or a whole new tower when all that was needed was a new entrance to an existing one, as happened in the case of 1003.5.

No castle is complete without a ghost, and POSIX has one: OSCRL—Operating System Command and Report Language. Started in the early eighties, it was (to simplify to an almost indictable extent) an attempt to define a common Job Control Language for the large computers of the day. It found a home in SC21, which looks after OSI, just before it became apparent that UNIX was going to become the “open” operating system of choice. Ahead of its time, the OSCRL project attracted a small but enthusiastic following, but, as the years went by, work tailed off. It was all but non-existent by the time the ISO POSIX project was set up. However, it is ISO policy when starting new projects to examine any related work which it may have undertaken, and the search turned up OSCRL as covering topics to be addressed by 9945-2 and 9945-3.

SC21 welcomed the chance to pass the project to another group, and we reluctantly agreed to take it over. Then the ISO central sec-

retariat lost all the paperwork. (It is a fact of life that all bureaucracies lose paperwork.) We had an excuse to prolong OSCRL’s spell among the undead, provided that we could put up with the periodic howls from its (few) proponents.

These howls recently resulted in a polite suggestion from the SC22 AG that we should do something to quiet them. That something might be the massaging of the existing material (if it can be found) in to a Technical Report — a type of document which few people ever read, and the production of which is discouraged by ISO. But a TR may just be the sort of headstone that OSCRL lacks. We will be trying to nail down the coffin lid at our next meeting, which takes place in the Netherlands from 14th–17th May, 1991.

References

1. Preliminary Recommendations and Resolutions, ISO/IEEE JTC1 SC22 Advisory Group, London, October 1990.
2. Report on ISO/IEEE JTC1/SC22/WG15 (Paris), Dominic Dunlop, comp.std.unix Volume 20, Number 110, USENET, 5 July, 1990 (also published in ;login:, Volume 15, Number 5, September/October, 1990) No. 3, Autumn 1990
3. The Context for Programming Language Independence for POSIX, Stephen R Walli, comp.std.unix Volume 21, Number 197, USENET, 11th October, 1990

Book Review

lex & yacc

by Tony Mason and Doug Brown

(O'Reilly & Associates, 1990, ISBN 0-937175-49-8, \$24.95, 218 pages)

Reviewed by Vern Paxson
Lawrence Berkeley Laboratory

lex and *yacc* are two of the more powerful UNIX tools, especially suited for complex text manipulation and compiler-writing. They are complicated enough to warrant a book-length reference, which has been lacking. Sadly, *lex & yacc* is not even close to being suitable. The book's flaws are numerous and pervasive.

Clearly the book has not been proofread. It is riddled with typos and incorrect examples. It also needs editing. Terms and symbols are used before they are explained. Promised explanations and examples fail to appear. Established terminology is confused ("token" is used to refer to non-terminals, individual input characters, and the contents of symbol tables). The text is padded with replications: a 29-line example is repeated verbatim, with 2 new lines added. 50 lines of code are replicated exactly, bugs and all. The index is missing relevant entries and contains incorrect page numbers.

It is painfully clear that the authors are not experts with *lex* and *yacc*, either, for they miss numerous subtle and not-so-subtle points in their use. There is no discussion of how to redirect *lex* input; their comments on *yywrap* completely miss its main use; non-existent flags are discussed; the division between what's best done in the scanner and the parser is greatly confused. Numerous comments are appropriate only for toy scanners ("Much of the *lex* overhead is of a fixed size"; "the visual model of the finite state machine is an invaluable debugging tool"). Escape characters are missing or used when not needed. Throughout

the text the pattern `##*20$` is used to match comments, yet the correct definition is `#.*$`. The authors compound this error by treating comments as tokens to be returned to the parser, though their grammars have no rules for dealing with the tokens, so even if the correct pattern were used, each comment would result in a syntax error! Furthermore, these errors persist in the code the authors distribute with the book—they never tested the code! One wonders about the authors' proficiency with C, too: they use `bzero` to make a string empty rather than `string[0]='0'`.

Finally, the book lacks depth. The running examples are somewhat contrived and fail to address many common problems. There are no parse tree diagrams to explain grammar ambiguities, no advice on how to remove ambiguities once detected, no discussion of building abstract syntax trees, no explanation of the precedence of regular expression operators. Practical problems such as scanning C comments or strings with embedded escape sequences, or getting error line numbers correct, or using the scanner to aid in parsing ambiguous grammars, are simply not discussed.

All in all the book is too simplistic, erroneous, and confusing to be of much value to either the novice or the experienced user.

The reviewer, Vern Paxson, is a computer scientist at Lawrence Berkeley Laboratory, a computer science graduate student at U.C. Berkeley, and the author of flex, a freely redistributable version of lex.

An Update on UNIX-Related Standards Activities

Jeffrey S. Haemer

Report Editor, USENIX Standards Watchdog Committee

Report on IEEE 1003.0: POSIX Guide

Kevin Lewis <klewis@gucci.dco.dec.com> reports on the October 15-19, 1990 meeting in Seattle, Washington:

When 1003.0 left off in July, we were wrestling with guide content, profile structure, and self-discipline (which can be hard to find via consensus). Not only were the first two resolved, but we decided not to leave until we had resolved each of the fifteen issues in our issue log. This negated the original plan to walk through the document section by section, but to no one's sorrow. (Some members applauded the decision.)

Outstanding and resolved issues included federal vs. national standards, level of detail in write-ups, exhaustive vs. non-exhaustive listing of standards, descriptive vs. prescriptive approach, target audience(s), and document flow (being able to follow each element of the POSIX open system environment through each chapter easily). The group was euphoric over breaking through these logjams.

Mid-week we discussed the mock ballot. Jayne Baker, from 1003.5, who is also participating in our group, gave an excellent presentation on the "Do's & Don't's" of the mock ballot process. It became readily apparent from our discussion that we had been naive on mock ballots. We will discuss a detailed plan at the January meeting. I now see mock balloting on draft 11, which would become available around March, 1991. Stay tuned for more on this. I will be developing a mock-ballot group listing, and will welcome *active* participants. This will probably happen before the January meeting.

Let me finish by summarizing some key actions and decisions. Guide content and profile structure were resolved to the unanimous satisfaction of the group. The group agreed that the guide's content should be divided into two parts, corresponding to two audiences:

- For the "unwashed masses" of profile writers, those not doing standards develop-

ment, the balloted portion of the guide will contain guidelines.

- For the benefit of profile writers writing formal functional profiles, we agreed to place formal rules in an appendix. The group also sent a resolution to the TCOS-SS SEC recommending that some activity be taken up (be it in or out of the IEEE) to develop a POSIX core profile. Coincidentally (honestly, it truly was coincidental), the SEC approved a POSIX Platform Environment Profile PAR for 1003.1. Just between you and me, I think 1003.1 is the right forum for the work outlined in our resolution.

Let me end by repeating my earlier request. If you are interested in becoming a part of our mock ballot group, *please* contact me by phone (202)383-5017 or e-mail klewis@gucci.dec.com.

Report on IEEE 1003.4: Real-time Extensions

Rick Greer <rick@ism.isc.com> reports on the October 15-19, 1990 meeting in Seattle, Washington:

Real-time Ballot Recirculation

The real-time (dot 4) ballot, originally mailed nine months ago, will get its first recirculation in November. The primary reason for the long delay in resolving ballot objections has been technical reviewing or a lack thereof. Reviewers were assigned to each major section of the draft even before it went to ballot, but some sections are still completely unchanged from the balloted draft. This is supposed to be fixed by the November mailing.

Pthreads Goes to Ballot

Meanwhile, the *pthread*s document (dot 4a) is due to go out to ballot in December, so Jeff still has a 50/50 chance of winning his free beer. Personally, I think the *pthread*s draft is going out in better shape than its predecessor and will prob-

ably require fewer recirculations. On the other hand, it may face a major stumbling block on its way to becoming a standard that base real-time is not yet required to deal with: Language Independent Specification. While the base document was grandfathered out of the LIS requirement, it is not clear that *threads* will be awarded the same privilege.

Ironically, a language-independent specification for *threads* could do more to accelerate its acceptance as a standard than to impede it. A couple of highly contentious areas of the draft (thread-specific data and certain aspects of thread cancellation) are C-language specific. The rationale has been updated to note this fact, but some working group members feel that many potential objections could be avoided if the text of the draft proper explicitly noted the language-specific nature of these contentious features. Unfortunately, there seems to be no way of doing this, short of providing a true language-independent specification.

Signals (Again!)

One often hears the argument that the voluminous changes between one draft and the next show that attempting to standardize thread behavior is premature. There is enough substance to this complaint that it cannot be dismissed outright. In fact, one reason for balloting now was to use the balloting process to impose some sort of change control on the document itself! What some critics don't realize is that most semantic changes in the last year have all centered around a single issue: signals. The latest draft is no exception to this rule; it introduces yet another signal compromise.

I'm not completely happy with the latest signal compromise, but it has one major advantage over all previous attempts to unite opinion on this issue. The new compromise recognizes that the basic contention is not so much technical (i.e., per-thread vs. per-process signal delivery schemes) as philosophical. One camp feels strongly that traditional POSIX signal behavior should be extended in some way so that all the P1003.1 interfaces have some meaning in a multi-threaded environment (with some argument over just what this meaning should be). Others feel that asynchronous signal processing is completely

inappropriate in a multi-threaded program and would rather see an entirely new interface (*sigwait()*) to deal with the problems that asynchronous signals were originally designed to solve (with some argument about just what these problems are). To satisfy the "preserve the Dot One interface" camp, the document that goes to ballot will include precise, per-thread signal semantics, the details of which are bound to raise numerous ballot objections. To satisfy the "*sigwait()* only" camp, the Dot One interfaces to these semantics can be completely disabled by a run-time switch, the utility of which is bound to raise numerous ballot objections. Thus, the new signals proposal has all the earmarks of a good compromise: Everybody agrees that it solves the problem but nobody likes it.

One other realization to come out of the debate was that the same points have been argued over and over for four meetings. We concluded that the rationale behind the *pthread* signal behavior specification was inadequate and needed to be totally rewritten. Unfortunately, we didn't rewrite it at the meeting, although people were assigned to do this work before the draft goes to ballot. It remains to be seen whether the new stuff that goes out to ballot will be any easier to understand than the old stuff. I just received a copy of the new signal rationale with a request to proof it, so I'll have an opinion soon.

Rate Monotonic Scheduling

One reason that *threads* did not go to ballot after the July meeting was that some of the multiprocessor folks (Dot Fourteen) felt that the thread scheduling section was overspecified, leaning heavily towards uni-processor, fixed-priority scheduling. (The counterargument was, "Of course it leans heavily towards uni-processor fixed-priority scheduling: That's what we're trying to standardize!") Those who objected the loudest were charged with coming up with specific changes to the draft that would correct the problems envisioned by the MP experts (e.g., high overhead in MP implementations because of the need to maintain uni-processor scheduling invariants and the general preclusion of novel MP solutions to resource allocation problems). To their credit, they did this without changing any of the previously specified scheduling interfaces or up-

setting any of the uni-processor semantics implied therein. What they did was to introduce a couple of new terms and concepts: “processor-allocation domain” and “thread-contention scope.” They went on to say that when the processor-allocation domain is greater than one (i.e., when you’re running on a multiprocessor), the effect of setting a thread’s contention scope is implementation-defined (i.e., MP schedulers are free to ignore the contention scope). Everyone seems to think this leaves the door open to any sort of multiprocessor resource allocation scheme that is likely to appear in the future. (Can anyone supply a good counterexample?)

As long as the thread scheduling chapter was still in flux, however, others felt that this would be a good time to introduce an additional interface in support of rate monotonic scheduling. RMS is used in some real time systems to deal with, among other things, the problem of priority inversion¹. This met with widespread disapproval. There were three major arguments for not including an RMS interface in the draft:

1. RMS is not common practice, so it shouldn’t yet be standardized. One would think that the history of *pthread*s itself is enough to illustrate the futility of this line of reasoning. After all, how many commercial implementations of a particular feature are necessary before said feature becomes “common practice” — especially when the very lack of a standard interface is a force preventing widespread acceptance of the feature in the first place?

2. Given that RMS is primarily a real-time tool, it should be specified as part of dot 4 rather than dot 4a. It is ironic that by sending dot 4 to ballot without a threads chapter the work group has effectively weakened arguments for including real-time features in the *pthread*s standard. And yet, *pthread*s was specifically introduced as a means of solving asynchrony problems in real-time environments!

3. RMS can and should be supported by adding new scheduling attributes, rather than new interfaces. While I agree that RMS *should* be supported via new scheduling attributes, it is not at

¹ Priority inversion occurs when a high priority task waits for resources controlled by a low priority task which is, in turn, prevented from executing by a third task whose priority lies somewhere in between.

all clear to me that it *can* be. The real problem here may be a deficiency in the *pthread*s attribute mechanism rather than the lack of a specific RMS interface.

At any rate, the technical reviewers can expect ballot objections from the proponents of RMS and should be thinking about ways to accommodate them.

Report on 1003.6: Security Extensions

Ana María De Alvaré <anamaria@sgi.COM> reports on the October 15–19, 1990 meeting in Seattle, Washington:

Just when you thought it was safe ...

Hello, readers! I’ve been away for a while, but I’m now back in the P1003.6, POSIX Security Extensions battles. [Editor: And we’re glad you are.] So much for socializing; on to the October meeting.

IEEE 1003.6 continued its work on Discretionary Access Control (DAC), Mandatory Access Control (MAC), privileges, and audit. It also spent half a day in a consortium with the networking, administration, and security groups addressing areas where the four intersect.

Balloting

The group currently plans to mock-ballot Draft 8. Group consensus was to push the standard forward and get serious about producing a document that we can agree on. The meeting must have been akin to Congress’s recent budget struggle.

The group will address written comments on draft 8 at the January meeting, clean up the draft, and send draft 9 out to balloting members. Only IEEE or Computer Society members can ballot, so if you want your objections to count as official votes, join now. As with all IEEE standards, 75% acceptance will be needed to pass.

During the ballot phase, the group will focus on answering ballot objections, and creating both a language-independence interface and a test suite.

;login: 16:1

Objections, objections

At the October meeting, we discussed P1003.2 commands, cleaned up the DAC mechanism and privileges sections, redefined the audit interface, and filled gaps in MAC. I will discuss each section separately.

P1003.2

We produced two lists from the current P1003.2 draft. One contains commands that need clarification or need to address security: *mailx*, *mv*, and *cd*. Jeremy Epstein, from TRW, will write up our concerns about this list and send them to P1003.2.

The other list is for our own subgroups, which will examine them to decide which are relevant to their subgroup. The commands that need security input are: *cd*, *chmod*, *cp*, *find*, *getconf*, *id*, *kill*, *chown*, *chgrp*, *ln*, *lp*, *ls*, *mailx*, *mv*, *nohup*, *rm*, *rmdir*, *stty*, and *test*. In addition a new command is needed for doing *pax* with security; *bpax* was a name suggested. [Editor: No, no. The precedent is set: *pax91*.] At the January meeting, we will move on to the User Portability Extension, P1003.2a.

One issue raised by the collaborators that produced these lists was that each subgroup in P1003.6 has created a get/set function for its particular area. After considering whether the group should consolidate all get/set functions into one or add options to existing commands, the whole group agreed to stay with the original design: a get/set function per area. The justification was that this design allows a flexible interface and implementation, and makes it easy to add functionality without changing existing commands.

Mandatory Access Control (MAC)

The MAC group feels their portion of the document is almost ready for mock ballot but still needs to address multi-level directories, especially *mlmkdir* (create a multi-level directory) and *mlrmdir* (remove a multi-level directory).

The group agreed on the treatment of opaque objects in the standard, and decided the standard should support variable length objects. They looked at the P1003.2 commands and inserted their information in chapter 5 of P1003.6 and decided to make information labels optional.

MAC brought a pair of open issues to the whole group:

1. Should we use options or positional parameters:

setlabel -x label file

or

setlabel label file?

We agreed on options.

2. At the plenary session, we agreed to make all P1003.6 interfaces optional. Thinking this through, MAC asked what wording to use when one area depends on another. In particular the group wants to address what happens in the absence of least privilege.

Everyone agreed that we need consistent wording, and we will look at proposals next meeting.

Discretionary Access Control (DAC)

The DAC group spent all their time cleaning and preparing their rationale for mock ballot. The scheme they've come up with for Access Control Lists (ACLs) is interesting, but a little complicated.

ACLs contain four entries: `USE_OBJ`, `MASK_OBJ`, `GROUP_OBJ`, and `OTHER_OBJ`. Three of them, `USER_OBJ`, `MASK_OBJ`, and `OTHER_OBJ`, correspond directly to the owner, group and other classes respectively. *ls -l* still displays the file type, such as 'd' for directory, followed by three bits apiece of "owner"-class, "group"-class, and "other"-class information. Modifying an ACL entry modifies the corresponding file class permission bits, and vice-versa.

So far, so good. But what's the fourth entry, `GROUP_OBJ`? Well, it's also related to group access (hence, the name), and often, but not always, contains the same value as `MASK_OBJ`. Here's the algorithm for checking read access based on `GID`: If the effective `GID` or any supplementary process `GIDs` match the `GID` of the file then:

1. If `MASK_OBJ` doesn't give read access, read access is denied. The check stops here.

2. If `MASK_OBJ` gives read access, the system checks to see whether the `GROUP_OBJ` grants read access.

- If GROUP_OBJ grants read access and matches the process GID, access is granted.
- If GROUP_OBJ denies read access and the ACL contains other GROUP_OBJ entries that match either the effective GID or any of the GIDs associated with that process, then it checks all of them until it finds one that will grant access.
- If none of them grants access, access is denied.

Audit

For the second time the audit group has agreed to follow the structure of the X/Open audit document. They are planning to merge X/Open with P1003.6 draft 7. They settled on: event types, headers, information per event type, generic structure, and function call interface.

They have not addressed any audit analysis definitions or interfaces for audit analysis.

Privileges

The privilege group went through draft 7 cleaning up descriptions, writing the rationale for the design, and writing examples on how privileges are assigned and inherited through *exec()* (including a drawing).

The major dispute was about the two flags associated with executable file privileges: *allowed* and *forced*. Members from AT&T, IBM, and Secureware claimed security systems did not require *allowed* privileges.

The close interrelationship of the forced and allowed flags create a complex mechanism to determine how privileges are granted. Let me explain. The *forced* flag lets system administrators grant a process certain privileges unconditionally. It provides backward compatibility with setuid programs.

The *allowed* flag, as originally stated, specifies that a new process image shall be permitted a privilege if the parent process image has allowed that privilege to be inherited. When *allowed* is set, *forced* is used to specify whether a new process image shall unconditionally possess that privilege.

The group agreed that although the interrelationship between these flags is counter-intuitive, the *allowed* flag provides a useful way to let sys-

tem administrators constrain the privileges that a process can inherit, and should remain in the standard.

The new algorithm for determining if a process inherits a privilege begins by checking whether the *forced* flag is on, or the *allowed* and *inheritable* flags are both on. If either of these is true, the *permitted* flag will be set (i.e., the process gets the privilege); otherwise, the *permitted* flag will be cleared (i.e., the process doesn't get the privilege).

Networking, Administration, and Security Consortium

The group met for half a day. The main focus was to identify areas where the groups overlap. Five were identified:

1. P1003.6 resources that need to be administered;
2. Additional security mechanisms;
3. P1003.7 (System Administration) objects and their security attributes;
4. P1003.6 areas that can affect networking;
5. Distributing mechanisms.

Summary

P1003.6 current and future plans are:

- January 1991 — Internal mock ballot on Draft 8
- February 1991 — Send Draft 9 to ballot
- End of 1991 — Write the language-independence interface and insert it as an annex to the balloting during recirculation

1991 (during balloting process) — Write test assertions

Register recirculation draft as a CD (Committee Draft in the international area).

Reorganize P1003.6 to fit the international standard's style: DIS 9945-1 (LIS), DIS 9945-1 (C-Bindings), DIS 9945-2 (command area).

;login: 16:1

Report on 1003.11: Transaction Processing

Elliot J. Brebner
<brebner@s5000.RSVL.UNISYS.COM>
reports on the October 15-19, 1990 meeting in
Seattle, Washington:

Who are we?

The POSIX Transaction Processing Profile working group (P1003.11, or Dot Eleven) is doing what it sounds like — standardizing various aspects of transaction processing in a POSIX environment. We're influenced by work going on in X/Open's XTP Working Group, but we're not simply waiting for X/Open's XTP Working Group to produce documents to be blessed. The group is maintaining a needed critical mass, and if X/Open does little to publish by the middle of next year, will be ready to move on alone.

Profiles

A key accomplishment to date is the push, along with the other profile groups, to have the coordination and Profile Coordination Group Meetings under .0 (Bob Gambrel's) leadership. The result should be a more uniform set of POSIX Profiles whose quality is improved via shared effort.

The Ad Hoc Profile Coordinating meetings were well attended and profitable. (As was ours — Dot Eleven had 16 active participants at the Seattle meetings.) We expect to see more Coordinating meetings in January in New Orleans. Meanwhile, we are starting to flesh out our Transaction Processing Profile in the POSIX-AP-Profile style that we agreed to in those meetings. Our working group doled out work assignments to review other P1003.n documents in search of needed functions. We also eagerly await the example profile that Don Terry has promised, the POSIX Platform Environment Profile being done in 1003.1, which the SEC has now approved a PAR for.

Carl Hall's draft input to the P1003.0 Guide, describing the P1003.11 TP Profile, was edited in detail. The revised draft was submitted to Dot Zero following the meeting, and should appear (with proper figures) in the next Guide draft.

APIs

As expected, the subject of new PAR(s) for TP API(s) came up. At the Danvers meeting, we agreed on a key point: that 1) although Dot Eleven should continue to work on distributed transaction processing, transaction processing need not be distributed, and 2) we need to define some standard API for the substantial existing practice. One API, between applications and transaction management services, should allow applications to start transactions, and then end them with either a commit or a roll-back. To do this, we need a new PAR.

A second possible API would specify the control of resource managers by a transaction manager¹, but the group considers this a system interface, not an API. For now, we and Dot Zero have jointly agreed to call this a System Programming Interface (SPI). Work in this area would also require a separate PAR. We put off deciding whether to submit one or both PARs pending a better understanding of exactly what would be in the API/SPI, confident that we're gaining that understanding rapidly as a byproduct of the profile work. Mark Carges' presentation on the X/Open AP-TM primitives provided further understanding of X/Open's work in this area.

If you received copies of the X/Open Preliminary Specification, *Distributed Transaction Processing: The xa Specification* (dated April, 1990) at Salt Lake, you should by now have received a mailing with instructions on how to comment on the document. If you have not, or you're an active Dot Eleven participant and would like a copy, please let me know. I will forward a request for a complimentary copy to X/Open. X/Open also sells copies at a nominal cost. X/Open has set the closing of the External Review period as January 1991, but comments are welcome earlier.

Language Independence

Stephen R. Walli <walli@osmcl1.gm.hac.com>
reports on the October 15-19, 1990 meeting in
Washington state:

1. The subject of the X/Open Preliminary Specification:
The XA Interface.

1. Programming Language Independent Specifications and POSIX

What is LIS?

A Language Independent Specification (LIS) is an ISO requirement for making POSIX an international standard, and the IEEE Technical Committee on Operating Systems, Standards Subcommittee (TCOS-SS) has agreed to provide the POSIX standards in this format¹.

An LIS is a rigorous functional description of operating system interfaces not tied to any programming language syntax or semantics. Each base LIS description is accompanied by a set of language binding descriptions². This approach — a base LIS standard plus multiple language binding standards — has been used in the past for specifying communications and graphics standards. Its advantage is that it lets programming languages bind to the abstract descriptions in a natural way; its disadvantage is that groups specifying a service interface must abstract that service from its historical language context.

Mundane as this sounds, if you aren't involved with the POSIX working groups, you may not know that LIS is emotionally charged. For starters, LIS affects most POSIX-related working groups. Base standards groups (1003.1, 1003.4, 1003.6, 1003.8, 1224) must all provide LIS versions

1. A quick standards-taxonomy lesson:

The TCOS-SS's executive committee (SEC) is responsible for co-ordinating the IEEE POSIX Working Groups' production of draft IEEE standards, and for bringing them both to ANSI nationally, and to ISO, internationally. While ANSI, the American National Standards Institute, creates U.S. standards, ISO and the IEC are responsible for the highest level of open systems standards; ISO committees and working groups guide standards to international acceptance. ISO/IEC Joint Technical Committee 1 (JTC1) is responsible for information processing standards, Subcommittee 22 (SC22) of JTC1 is responsible for programming-language-related work, and Working Group 15 (WG15) has the specific responsibility to define the criteria IEEE POSIX must meet to become an international standard (such as LIS).

Here, we will refer to the IEEE/ANSI effort as POSIX, and to the WG15 effort as ISO POSIX.

2. Don't confuse the language binding with a programming language standard. A standard C binding for POSIX includes things like C function prototypes for the POSIX system calls: this is separate from the ANSI or ISO standards for C itself.

of their documents. Language binding groups (1003.5, 1003.9) must bind to the LISs. All in all, POSIX participants run the extremes from those eager to use LISs to those vehemently opposed to all that it requires of their documents.

Some Areas of Debate

"Thick or thin?" One current, often contentious, debate revolves around language binding philosophy. While some argue that bindings should be "thick" stand-alone documents, which contain both syntax and semantics, others envision bindings as "thin" syntactic descriptions of how the language binds to the LIS with pointers to the LIS for any semantic description. The Ada working group has chosen the thick road; the FORTRAN working group, the thin.

Though thick bindings might seem more usable for developers, they have two potential problems. First, duplicating semantics risks differences of interpretation. Second, document synchronization complicates the inevitable base revisions.

"Scheduling" Scheduling and resource problems create another area of tension. Because working group chairs all want to move their draft documents forward quickly, an LIS requirement strains the already overworked, all-volunteer working groups.

Yet another problem is scheduling the work of interdependent working groups. Real-time, security, and transparent file access (1003.4, 1003.6, and 1003.8) are all extensions to 1003.1 and all part of the same ISO document (9945-1). 1003.5 and 1003.9 are Ada and FORTRAN bindings for 1003.1. Coordinating these interdependent documents — all at different points in their development and language independence — creates an often-noticeable tension.

Status

To aid the POSIX LIS effort, Paul Rabin (sometime 1003.1 snitch) and I were asked to produce a set of methods and guidelines.

The model is based on work done by ISO/SC22 Working Group 11 (WG11), which is responsible for defining common language-independent data types and procedure-calling mechanisms standards. Unfortunately, although the WG11 work is directly related to POSIX LIS, it hasn't yet pro-

duced ISO standards and POSIX needs the standards immediately. This poses another scheduling problem. (Incidentally, WG11 and WG15 realize their work is interrelated and keep each other informed of their progress.)

Before the October POSIX meetings, the Methods document had been refined to draft 2 and used successfully to produce an LIS for section 4 of 1003.1-1990. The Real-time group also used earlier drafts to produce an experimental LIS translation of large part of 1003.4/D9, but will probably not move ahead until the C-based draft finishes balloting. Meanwhile, P1003.9 (FORTRAN 77 Binding to 1003.1) is about to start balloting, and P1003.5 (Ada Binding to 1003.1) is balloting, but both were developed from a C-based standard.

Seattle, October 15-19, 1990

The October POSIX meetings saw much discussion of LIS issues. Official discussions were concentrated in the Birds-of-a-Feather (BOF) sessions and the SEC meetings.

The BOFs

POSIX meetings schedule regular LIS BOFs to discuss language bindings, LIS methods, and other technical issues. October saw two: one Tuesday afternoon and another Thursday afternoon. Paul Rabin chaired both.

The first was dedicated to work up to the present. With a comment or two added for clarification, here were the issues raised:

- As always, a few people questioned why we were doing the work. What purpose does it serve? These concerns were directed to the SEC, which sets LIS policies and schedules; justification for the work is an SEC policy issue.
- Some voiced concern about the Methods document's stability. Before the next P1003 meeting, it will go through one more major revision. Some take this to mean it is currently unusable, but I think this is unhelpful. Working groups must use this document to improve it, and holding back only delays the process. At this point, the basic format will probably not change; future revision will extend the document based on our experiences with this draft.

- "What exactly does ISO want?" was a common question. No one wants to get all the work done only to face an ISO rebuff. Although this valid question has sometimes delayed LIS work, by the end of the two weeks of POSIX meetings both the SEC and ISO had endorsed our direction.
- There is great interest in using LIS methods to help create base assertions for a binding. (Some feel this is the only reason for doing an LIS.) There is no question that it would help if bindings groups could use such work to help prepare the test assertions that the SEC has placed as another addition to the standards effort.

The second BOF was dedicated to LIS work in progress. We reviewed the Methods document, explaining the document format, the scope, the model, and the guidelines to LIS and language-binding writers. Though the document needs further review, there was general agreement with the approach.

The SEC meetings

LIS consumed a fair amount of the October SEC meetings.

Many now believe that LIS work is blocking C-based standards documents from reaching the industry. Recently, the SC22 Advisory Group (SC22 AG) recommended that the POSIX LIS work not delay standardization of C-based documents. A resolution advanced at the first SEC meeting to relax the LIS requirements at the IEEE level was tabled until the second meeting, and an investigative sub-committee was formed.

At the second meeting, the SEC resolved that C-based base standards that go to IEEE ballot after October 18, 1990 but before a 1003.1 LIS goes to ballot, must have an LIS before balloting ends. Documents that go to ballot after a 1003.1 LIS goes to ballot must have an LIS to go to ballot. Finally, once LIS 1003.1 is an ISO Draft International Standard, all documents must enter IEEE ballot as an LIS or as a language binding to an LIS. Before this, the LIS may take the form of an annex. Although WG15 only asked that such annexes be informative (non-binding), the resolution requires a normative (binding) annex to en-

courage detailed review³. The resolution grandfathers in Shell and Tools (1003.2) and Real-time (1003.4), but Real-time already has a separate PAR for the LIS.

There is great pressure on 1003.1 to complete a 1003.1-1990 LIS quickly. Though 1003.1 lacks resources to devote to this, many cite the lack of a 1003.1 LIS as reason to delay starting other LIS work. Addressing this, the 1003.1 working-group chair proposed a second resolution, which also passed. The resolution:

1. endorses the Methods document and its use and the tracking of future revisions,
2. encourages everyone to help 1003.1 complete their LIS, and
3. asks 1003.3 (Test Methods) to help classify 1003.1 test assertions as language-independent or C-specific.

All this set the stage for the ad hoc WG15 LIS meeting.

Orcas Island, October 22-26, 1990

WG15 met the following week. With only about twenty attendees, the meeting was manageably smaller.

Not a standards-drafting organization, WG15 is concerned with such issues as internationalization and co-ordination with other standards. Long-range, the group's goal is to make IEEE and ISO standards identical and cleanly integrated with other ISO standards. "Interoperability between standards" was a frequently heard phrase, and we were glad for the expertise that the Europeans present brought from their current EC integration work.

The first two days of WG15 saw an ad hoc LIS status review meeting, which involved roughly half the WG15 participants. Happily, the WG11 Convener attended and provided useful insight into their current efforts. Once again, Paul Rabin was the chair.

The ad hoc group reviewed the issues and concerns that the IEEE POSIX working groups had raised, and made a list of recommendations. The

3. After the meeting we realized that the SEC's resolution was silent on C-language bindings to the LIS annexes, but the ad hoc WG15 LIS meeting the following week fixed this, deeming it an implicit requirement.

full session later discussed these and had the drafting committee prepare formal resolutions, which were passed on the last day.

Resolutions included these:

- WG15 supports the Methods document's scope and direction. Specifically, this document says that LISs do not require formal methods, and that interoperability between applications written to different language bindings is desirable, but not required. (Rationale and discussion for the Methods document's scope is provided in both ^[1] and ^[2].)
- Future LIS and C bindings need not *exactly* match IEEE 1003.1-1990; requiring this would prevent necessary bug fixes and addenda. (But reviewing the WG15 resolutions^[3], I can't find this written down.)
- Base LIS standards should specify conformance requirements for language bindings. The Methods document still requires an addition here. It is to be hoped that WG11's work in this area will prove helpful.
- WG15 asked the IEEE to provide a schedule for the delivery of the LIS work, including in that request the constraints on the schedule and the expected items for delivery. (A complete discussion of this resolution can be found in ^[4].)
- The motivation for the resolution on LIS scheduling says:

WG15 considers that thin language bindings (which must be read with a service definition) are suitable for standardization, but thick bindings, which incorporate a service definition duplicating and possibly conflicting with the service definition provided by another standard, are not;

This sends a clear message to the thick/thin language bindings debate.

Putting it all together

How does all this come together and where is it going? Here is my opinion — not my employer's, not P1003's, not WG15's.

Everyone official seems to endorse the Methods document's scope and method.

The Methods document is stable and usable. Draft 3 will be upwardly compatible with Draft 2, and will have the same format.

At its next meeting, with everyone interested in the outcome, 1003.3 will investigate the subject of LIS test assertions. A high percentage of test assertions may prove language independent.

The thick Ada binding will continue to come forward as is at the IEEE level, but will need to slim down to become an international standard. If no one is interested in an international standard, no thin binding may be needed. Document synchronization and standards revision issues with thick standalone national language bindings probably won't be felt for some time. The issue of writing test assertions for the binding looms and will spark much debate in the January meetings.

There are document synchronization issues between TCOS and ISO, which affect the coordination of C-based documents, LIS and language bindings, made worse by the time frame differences between the TCOS and ISO standards processes. Scheduling this large, complex project staffed by overworked volunteers is a thorny project management issue.

It would be great if the SEC built and published a proper schedule, itemizing all of the individual work items, (base C-based draft document, LIS annex, C-binding annex, bindings) and dependencies. This would clearly convey to all parties the work to be done, the interdependencies and the rough time frames. Indeed, this is what WG15 has requested.

The SEC has already outlined what needs to be done to enter/exit IEEE ballot. WG15 merely wants to see project milestones, dependencies, and a schedule. This will certainly spark interesting debate in January.

P1003.1 will stay in the pressure cooker. The SEC's encouraging resolution has no teeth, so will help little. Some have mentioned October 1991 as a date to expect completion of a 1003.1 LIS, with a C binding by the following spring. This seems to be a reasonable date based on the current involvement with LIS to date. It would certainly be helpful if people contacted Paul Rabin (at the address below) to help move this work forward more quickly. Although groups cannot complete their LISs without a 1003.1 LIS, they will continue working productively, following the model of Real-time.

For more information

Paul Rabin is managing an LIS mailing list. Messages for distribution to the whole list should be sent to posix-lis@osf.org (or uunet!osf.org!posix-lis). Requests for updates to the list should be sent to posix-lis-request@osf.org.

References

1. At this point, we have a document, *Programming Language Independent Specification Methods* that outlines the LIS model, and gives guidelines and a format for base LIS standards and guidelines for bindings writers. For brevity, I'll refer to it as "the Methods document."
2. Stephen Walli, *The Context for Programming Language Independence for POSIX*, comp.std.unix Volume 21, Number 197, USENET, 11 October, 1990.
3. Final Version WG15 Resolutions, ISO/IEC JTC1/SC22/WG15, Orcas Island, October 1990.
4. Dominic Dunlop, *Report on ISO/IEC JTC1/SC22/WG15 (Orcas Island)*, comp.std.unix Volume 22, Number 24, USENET, 15 November, 1990.

Report on Name Space/Directory Services

Mark Hazzard <markh@rsvl.unisys.com> reports on the October 15-19, 1990 meeting in Seattle, Washington:

Introduction

I'd like to introduce a new POSIX work group: Name-Space and Directory Services (NS/DS). A PAR has been submitted to and approved by the TCOS SEC, so the working group will be official by the New Orleans meeting in January. The group's number will be 1003.17.

You don't have to be clever to detect a duality in our name. We are trying to solve two separate but related issues.

Name Space

POSIX has several name-spaces: the file system, processes, user and group IDs, and others (with more on the way). Consider, for a moment, just the process name-space. Today, when I want to know something about processes, I use *ps*, which typically shows a one- to five-digit

process-id for each process in the left-hand column of its output. I get a unique value for every process I've asked about. This works because I only care about local processes — processes “within” the specific kernel I'm talking to.

Suppose that it becomes commonplace to run concurrent processes on several kernels. It would be useful to ask for the status of all the processes I own with a single command. A natural way to do this would be to extend *ps* to solicit and display status information on global as well as local processes. The name-space for the process ID would need to be extended to identify a process uniquely within the universe (or some reasonable subset of it).

This is the name-space problem in a nutshell. The group discussed it at length but we were not sure how to proceed. We found ourselves ping-ponging between the name space issue and the “other half” of our charter, Directory Services (DS).

Directory Services

We are a networking group, so to us *directory* means something like DNS or X.500, not “a file that contains file entries.” Specifically, we intend to provide an “API to a directory service, including but not limited to X.500 functionality.” After some soul searching, the group has decided to focus on the DS aspect of our charter, given the limited resources we've been able to muster.

Name-space and directory services are related. Directory functions allow users to read, write, list, compare, etc. global objects and their attributes. Objects are defined within a name-space and share basic characteristics: syntax, semantics, authority, and uniqueness (they can exist within only one name-space). It's logical to conclude that the name-space for an object must be defined before it can be put into the directory information base. This has already been done for many OSI objects (CCITT X.520/521)¹.

1. OSI/CCITT have defined the objects they want X.500 to manage, so we don't have to. These are specified in CCITT X.520, Selected Attribute Types, and in CCITT X.521, Selected Object Classes.

Base Documents

One of the first activities the group undertook was to identify candidates (if any) for a basis specification for the DS API. We turned up a pair of candidate specifications that had been cooperatively developed by the X/Open XNET group and the X.400 API Association. Together, the two specifications, XOM (Object Management) and XDS (Directory Services), form a single API to directory services. The group evaluated their functional content against the requirements and agreed to accept XDS and XOM as the basis for the DS API.

You may be wondering why two specifications are required to define one API. A little historical perspective might help. The collaboration mentioned above actually developed a trio of specifications. The one I haven't listed defines interfaces to (you guessed it) X.400. They decided to define a general purpose API (XOM) for managing OSI objects, which would work for both X.500 and X.400 API (and possibly others).

Our group is currently in the process of “POSIXizing” XDS. This means reworking XDS to conform to POSIX style, content, and format requirements. Improving XDS's readability and comprehensibility is another goal. Providing a language-independent binding and test assertions will require a major effort.

“But what about XOM?” you ask. We were curious too. Since we plan to share XOM with at least one other TCOS group (P1224 - X.400), we thought we'd better ask the Distributed Services Steering Committee, which oversees all the networking-related groups. It answered, “Keep XOM a separate specification and cut a new PAR for it. The PAR will be submitted by the P1224, but the work will be shared between NS/DS and P1224.”

Summary

So, that's what NS/DS is working on. We've got a first cut at a table of contents, and have hacked our way through a language-independent binding for at least one function call. We're trying to get a lot of work done between cycles, and we expect the next few meetings to be “roll up the sleeves” sessions as we work our way through the documents.

CALL DOC STRANGE

Peter Collinson
 Hillside Systems
 Canterbury, UK

pc@hillside@co.uk

*Peter Collinson is a freelance consultant specialising in anything that will pay the maintenance of his machines. He has been associated with EUUG since its early days and is proud to be an honorary member. Peter thinks that the word UNIX should be allowed to be a noun but he still cannot work out whether he should use ++*argv or ***+*argv or ++**argv. **getopt** is for Users.*

This colum is run by Colston Sanger, of GID Ltd
 doc.strange@gid.co.uk

Some Canterbury Tales

Well, it has finally happened. I have my new machine installed and running. "Hey, what's all this about", I hear you all shout loudly. "We didn't even know about the old machine, and you're telling us about this new one". OK. OK. Here's a bit of history.

When I decided to give up the daily grind of working for UKC and become a freelance UNIX[†] person, I felt that I needed a machine to keep me occupied when times were boring, to run my accounting package, to act as a mail host, calculator, diary, word processor and whatever else I do on a machine.

I wanted a Sun because it was not System-V-consider-it-standard and it could run X windows. At the time the most cost effective workstation that could be bought was the Sun386i. It was cheaper and faster than any competitor. I decided that I wanted a monochrome Sun386i/250 with 8 megabytes of memory, a cassette drive for backup and 300 megabytes of disc. I later added another 4 megabytes of memory and a standard AT two channel RS232 card.

On the whole the machine was OK. In the heat of last summer, I finally had a disc drive replaced because it suffered from Sun386i disc overheating disease. It was good going from the Sun3/60 that I had finally managed to get hold of at UKC to the 5 MIPS of the Sun386i, compilation was faster and X windows ran at somewhat closer the speed at which it is usable. Of course, you get used to the speed quickly and need to return to a slow machine from time to time to understand how lucky you are.

I have a lot of very bad things to say about Sun's European pricing policy – it seems to consist of importing the US price lists and changing the \$ signs to £'s. To be fair this is not just Sun, the whole hardware industry seems to do this. It is getting crazy, many things are now small enough to be brought over by an individual and imported quite legally. The total cost including the air fare will be much cheaper than the UK list price and in many cases much



cheaper than the discounted UK price. What do we lose by doing this? Warranty, I guess. But most things don't break in the first year and those that do are often flagged as problems on the net. I must stop this old old complaint and move on...

On the other hand, I have a lot of very good things to say about Sun's software. The whole question of portability is covered very well by Sun. You have *port* code to move it into Xenix, SCO UNIX, Ultrix and all the other 'IX-ses' – you simply compile it on a Sun. And it works.

To me this is worth a *lot*. I have a number of public domain programs that I just want to compile and have them work I don't understand how they work, and I don't want to. I just want to compile and run them.

It is worth a lot for development too. In the last year, I wrote, compiled and tested a program suite destined for a System V Bull machine. I did this entirely on my Sun, bashed the code onto the floppy as a tar image using the DOS emulator, moved the file into the target machine using kermit, unpacked and typed make. *Good stuff.*

It became apparent during the last year that Sun were no longer going to support the 386 range of machines. They failed to produce a new operating system release. They announced 486 upgrade and product, and then officially scrapped the idea. However they offered (and still offer) a very good upgrade path for existing Sun386i sites, replacing the machine with a SPARCstation I+. Basically, the site retains the screen, keyboard, mouse, 300Mbyte disc and cassette drive. You obtain a new SPARCstation I+ system unit that lives under the monitor. The system unit comes with 8Mbytes of memory as a standard, and you can order an 100Mbyte internal disc and a floppy. The disc and cassette live in an external unit that is formed by taking the old expansion cabinet from the 386i and adding a new plastic base.

Of course, the upgrade is half price in the USA – around \$4000, translating into around £4000. But for this price, anyone would be crazy to ignore the offer – it's chance to replace a 5 MIPS machine by one that runs at 15 MIPS. It's an opportunity to move back into

† UNIX is a registered trademark of UNIX System Laboratories, Inc., a subsidiary of AT&T, in the U.S. and other countries.

the world of supported systems and on into the bright new SPARC based future that we are all being promised. The only possible reason for continuing with the Sun line is the desire to have an AT bus accessed by a reasonable UNIX system or perhaps the need to run DOS applications in the Sun DOS emulator. There are SPARC alternatives to the latter, one being European – SoftPC from Insignia in the UK[†]. Actually, this particular emulator is better than the original one I had on the Sun386i.

Anyway, the upgrade was ordered and delivered around a month ago (I am writing this in mid-December). There was some worry that the 8Mbyte of extra SIMM memory from my Sun386i would not work in the new machine. The engineer said that the company policy was that 'he could not officially advise me to put it in, nor was he allowed to do so'. However, he said that it should work and sipped coffee while I installed it into the machine – the machine rebooted saying "yes, I have 16 Mbytes now". It does seem that I should be putting 70nS SIMMS (or better) into my SPARC and I have installed 80nS. Are these running hot and will die at some point? Should I really change them for 'proper' SIMMS? I don't know. It would be nice to know the real story about SIMMS, I guess.

The engineer installed the latest SunOS release, version 4.1 and OpenWindows 2. He left at lunchtime and I set to work recovering my system. I had used dump to save images of the old Sun386i file systems. I had checked that when I ran restore on the SPARC, it would not be confused by the reverse byte ordering on the 386i. The good news is yes, the SPARC restore program announces that it needs to swap bytes, but continues running (full marks again, Sun). So I pulled back all my public domain sources and started compiling.

By around 2200, I had put up and tested everything except the X Window system. I was using the OpenWindows system as a base for running things, and being annoyed by the dreaded cmdtool and shelltool. Just which one do you run? Why is cut and paste so difficult?

OpenWindows 2

I spent the next day looking at OpenWindows 2. I suffer from lack of manuals, although there is a lot of on-line documentation and demos. First, the server runs both X and NeWS, and this is a goody. It's great to be able to have PostScript on the screen with fonts that actually map onto the fonts on my laser printer. Second, it does allow you to use the old SunView tools, the new OpenWindows tools and the various things that I have come to use from the research Athena set.

[†] Contact Insignia Solutions Ltd., Victoria House, 28-38 Desborough Street, High Wycombe, Bucks, HP11 2NF; phone: +44 494 459426.

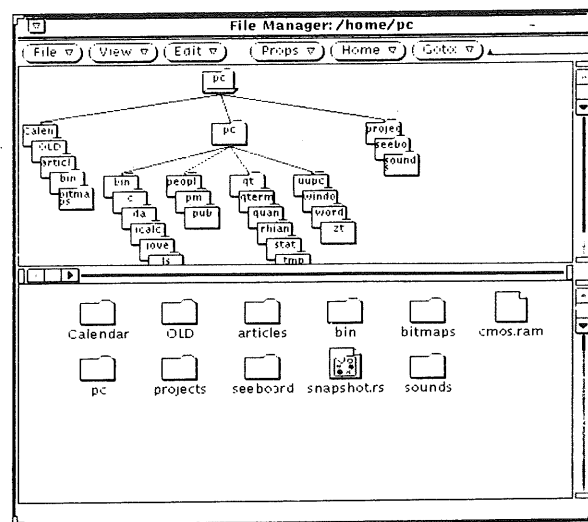


Figure 1: The OpenWindows File Manager

The OpenWindows interface is much more "Mac-like" than I am used to. The look of the file manager application is shown by the half-size screen dump in Figure 1. It shows the Xview 'look' of the window manager olwm, and the rather nice scroll bars supported by the widget set. The grips on the corners of the window allow you to easily resize things. The application is a 'direct manipulation' tool, you can pick files up and drop them into a wastebasket, or onto the printer application. This type of operation also works with mailtool and some of the other applications. Quite nice, but I personally find that I don't do that.

After playing with things for a bit, I decided to try and use the Athenative based applications – I especially wanted to use xterm, xmh and twm. I reloaded the X11R4 sources. I wanted to try and use as much of the existing X11 infrastructure as I possibly could so that I did not have to run two different universes. For one thing, I am now short of disc space. The idea is to use the libraries and include file from OpenWindows 2 and pretend that they were installed from the X11R4 sources.

I planted a lot of symbolic links from /usr/local/lib to /h/openwin/lib where OpenWindows 2 is installed. The main one is a link called X11 that points to the OpenWindows lib directory.

```
#!/bin/sh
cd /usr/local/lib
ln -s /h/openwin/lib X11
# link other library files
for name in X11/lib*
do
    ln -s $name
done
```

This makes use of a useful feature of ln. I then created a copy of the include files in /usr/local/include/X11.

```
#!/bin/sh
cd /usr/local/include
mkdir X11
cd X11
for name in /h/openwin/include/X11
```

```
do
  ln -s $name
done
```

This directory will be modified by the install process later. I planted another symbolic link from `/usr/include/X11` to `/usr/local/include/X11`. That's it. The application libraries and include files for X11 are installed. All this may not be required if you ask for the OpenWindows package to be fully installed at load time, I didn't want its tendrils all over my system in case I needed to take it away.

Now for the Athena libraries. You need to install the Athena widget set from `mit/lib/Xaw` and also the miscellaneous utilities library in `mit/lib/Xmu`. Compile these and do a make install. You will find that real libraries now pop up in `/usr/local/lib` and the new include files in `/usr/local/include/X11`.

When compiling clients using the new libraries, you must tell the compile process to use installed libraries and include files. The easiest way to do this is to use `imake` replacing all the existing junk in `ximake` by:

```
XMIT=/s/X11/mit
$XMIT/imake -DUseInstalled
  -I$XMIT/config -DTOPDIR=$XMIT
  -DCURDIR=.
```

(Should be one line really).

The XMIT string should be set to the place where your X11 sources are stored. Then when compiling clients simply say:

```
ximake
make
```

I compiled `twm`, `xbiff`, `xclock`, `xload`, `xmh` and `xterm` with no problem.

Using Athena Clients Under OpenWindows 2

Athena clients seem to run happily. You will find that `twm` windows will need an extra button. There is a large cultural difference between the clients designed to run under `olwm` and `twm`. Athena clients usually have some way of allowing the user to make them go away – they come equipped with quit buttons or quit menu selections or something. OpenWindows clients expect the window manager to cause their death, and generally use the Quit selection on the drop down menu belonging to their outer frame.

Luckily, `twm` can cope with this. You simply add the line

```
RightTitleButton "target" =f.delete
```

to your `twmrc` file. This gives a cursor shape to use as a button bitmap (`target`) and an action to perform when the button is pressed (`f.delete`). You can add

```
LeftTitleButton
```

should you so desire. You will find that the button appears on all the subwindows of an application. The `f.delete` action is polite, in

the sense that it is only actioned when a window requests it, so pressing the button in pop-up windows simply causes the bell to ring. You can see what this looks like from figure 2, an OpenWindows clock running under `twm`.

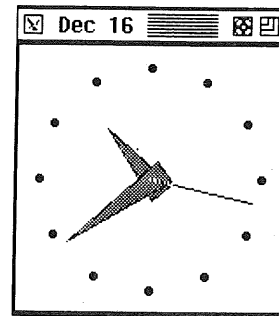


Figure 2: OpenWindows Clock under `twm`

The only other problem is that OpenWindows 2 does not offer the SHAPE extension, even though PostScript can do it so I would guess that it could. So you have to say goodbye to those nice shaped buttons and see through windows. My little boy is disappointed that he cannot fill the screen with zillions of xeyes, it's not the same with frames round them. More seriously, I miss this facility.

I have found that there are difficulties in mixing applications from NeWS, SunView, OpenWindows and Athena on the same screen. Things mostly work, but there are 'rough' edges and bugs in the server pop up. It is hard to put your finger on exactly what is wrong under what circumstances. There are problems with input focus and the text cursor, meaning that sometimes it appears to go away when doing mixed mode working. If you enter an OpenWindows application say `calctool` from an `xterm` – then when you return to the `xterm` your block cursor has gone away as if the focus is in another window. This seems to be dependent on the stacking order of the windows, OpenWindows ones like to be on the top of the stack.

NeWS applications seem reasonably well behaved and interwork fairly well. SunView applications seem reluctant to work with anyone. I suppose it's a wonder that they interwork at all. They will work but their images can get distorted with odd bits of graphics appearing in odd places. Also sometimes they change the cursor shape and you are stuck clicking away until the new shape is altered to something that is correct.

In some ways, I feel that the problems are minor. But yet again, I seem to have gone from an 'academic system' to a 'commercial' one encountering dumb problems on the way. The academic system works and has very few minor niggles. The commercial system works and has a number of large major niggles and a propensity for clients to crash from time to time.

Other Experiments

I tried to take a binary of the standard X11R4 server and run it. This didn't work because Sun appear to have changed the way that fonts are used – presumably to support the infinite font scalability

of the NeWS system. I could not see an easy way to get the Sun font set to interwork with a standard X11R4 server. I had wondered about whether I could slip in different window managers depending on what I am doing. I have given up with this idea for a bit.

I started to use GNU C to compile the various parts. I have installed the whole thing apart from the loader, largely because the GNU ld program cannot cope with shared libraries. On a single user machine, shared libraries are great, saving masses of paging space and speeding up program loading. The GNU compiler simply works and generates much faster code than the standard Sun cc compiler. I have changed all the programs that I can recompile to use gcc to get the speed improvements. I never used it on the Sun386i, due largely to problems with COFF – although I believe that those problems have been fixed now.

A Story from the Net

Colston asked me for a joke. Well I love this next story, it appeared in alt.folklore.computers and was supplied by Brian Randell from the Computing Laboratory, University of Newcastle upon Tyne, UK.

"I can vouch for the following story, which happened in (I would guess) about 1960 at the English Electric site at Whetstone, near Leicester, England, whilst I was employed there as an applications programmer (but was actually devoting all my time to compilers - or "automatic programming" as we then called it).

"English Electric Whetstone housed two major departments, the Mechanical Engineering Laboratory and the Atomic Power Division. Their first digital computer was a DEUCE - effectively a slightly re-engineered version of the original Pilot ACE, developed at the National Physical Laboratory by a team that was originally

headed by Alan Turing. It was a physically quite large machine, built from valves (vacuum tubes) and using mercury delay lines for high speed storage (about four hundred 32-bit words) and a magnetic drum (8k words, I believe). It was air-cooled, with a large fan under the floor pulling in air from outside, which was then blown over the electronics and allowed to escape into the computer room.

"Many stories can be told about the DEUCE, but the most memorable incident at Whetstone was the following. The computer was run overnight by a small operating staff, who recorded their activities in a log book. I and my colleagues were in the habit of checking this log book each morning, and at one time noticed that over a period of a few days there were a gradually growing number of reports of the machine failing, and of a nasty smell – but none of us connected these facts, or succeeded in tracking their cause. Then one day the underfloor fan started becoming somewhat noisier, the smell increased dramatically, and soon afterwards the machine failed abruptly and spectacularly, with red lights all over the power distribution board.

"What had happened was there had been a break in a sewer pipe – a pipe being fed by all the toilets in the large multi-story building whose ground floor housed the computer room. The sewage gradually backed up, and then overflowed into the hole in the ground housing the fan, and then into the fan itself, so as to be distributed evenly and efficiently – for a while at least – around the whole computer!

"It took days to dry out and disinfect the machine – but it was returned to service, though the maintenance engineer never lived down the incident."

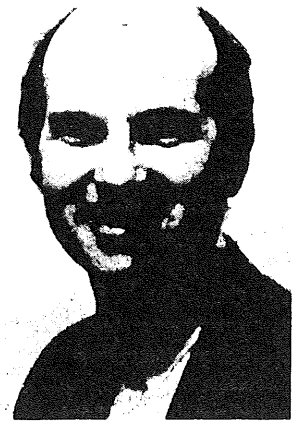
Thanks Brian for permission to reprint.

STANDARDS COLUMN

Dominic Dunlop
 The Standard Answer Ltd.
 Cholsey, UK

domo@tsa.co.uk

Dominic Dunlop has been hanging around the European UNIX scene for years now, industriously researching into why things won't work and assiduously spreading the bad news among those who will listen. Involved with POSIX during its growth from a single, short document to a multiplicity of legalistic tomes, he is currently paid to report to EurOpen (formerly EUUG) and USENIX members on the progress of the ISO POSIX working group.



For the past couple of years, these columns have discussed events and developments in the POSIX-related activities of the International Organization for Standardization (ISO). This time, I'm going to look at a lower — but arguably equally important — level in the standards development process: the Institute of Electrical and Electronic Engineers' Computer Society Technical Committee on Operating Systems Standards Subcommittee. Let's just call it IEEE-CS TCOS SS, or, better still, TCOS.

Last October, EurOpen agreed to provide funding for an institutional representative who would attend the quarterly meetings of TCOS, and provide a means of routing input from European users of open systems into the bewilderingly large variety of POSIX standards being developed by working groups under TCOS. I am that representative, and, since I'm spending your money, I'd better tell you what is going on, why it's important, and how you can help me out.

POSIX Development — Top Down or Bottom Up?

I've referred to the IEEE in my reports on ISO matters, since it is the IEEE which actually develops the POSIX standards. The IEEE routes its documents to ISO via ANSI, the American National Standards Institute. Translating this into ISO-speak, ISO has designated ANSI, its U.S. member body, as the development agency for the POSIX standards. ANSI, in turn, has delegated the work to the IEEE, an accredited body which it considers competent to create operating system standards through a consensus process which allows all interested parties to comment.

This makes the process of standards development look as though it proceeds from the top down: somebody associated with ISO decides that the time is right for a POSIX standard, identifies a means of getting the job done, and controls the process in an orderly, structured manner.

Life is not like that. No matter how much those who work at the ISO level would like to believe that they are, and always have been, in the driving seat, the movement towards POSIX started from the bottom and drifted up. It started in the early nineteen-eighties with /usr/group, a U.S.-based organization of suppliers and

commercial users of open systems, now known as UniForum. This group created The 1984 /usr/group Standard, a minimal definition of an operating system interface corresponding broadly to the unprivileged services offered by AT&T's UNIX System III, together with selections from the Kernighan & Ritchie C language library. Slim but seminal, this document was passed into the IEEE (specifically, to the newly-formed TCOS) to provide the foundation of the POSIX standards. It also gave important input to ANSI in the creation of a standard for the C language.

Despite the fact that neither the IEEE nor ANSI puts any nationality requirement on the individuals (in the case of the IEEE) or the organizations (for ANSI) participating in the creation of their standards, both POSIX and C initially developed in the U.S. with little international input. The costs of travel and of assigning English-speaking technical experts to the task was (and is) one disincentive; another is the feeling, particularly in Europe, that standards activity should begin at home, rather than in the U.S.

By 1987, the international demand for standards for POSIX and C was obvious, and it was natural that ISO should get involved. To be pedantic — and the standards world is nothing if not pedantic — it was natural that Joint Technical Committee 1 (JTC1) of ISO and the International Electrotechnical Commission (IEC) should get involved. (JTC1 had been formed in the mid-eighties to end wrangles between ISO and the International Electrotechnical Commission over the right to create standards for information technology.) It was also natural that the project for the international standardization of the C language should be handled by JTC1's Subcommittee (SC) 22, which is concerned with programming languages. SC22 Working Group (WG) 14 was duly set up to do the job.

It was less natural for POSIX to be assigned to WG15, another new group under SC22. An operating system interface, after all, is hardly a programming language. Nevertheless, after an attempt to set up a new SC to handle system interfaces had failed for political reasons, SC22 picked up the work¹. Both WG14 and WG15 appointed ANSI as the development agency for their respective standards, leaving us with today's situation.

At this point, I shall have to stop discussing C standardisation, as it is not a field in which I am active². But I can tell you more than

you probably want to know about the activities of IEEE TCOS, which is at the work-face of POSIX development.

POSIX in the IEEE

When TCOS was set up in 1985, it had just one IEEE standards creation project under its control — project 1003, known as P1003. (Other well-known IEEE standards projects are 754 for floating point formats, and 802 for local-area networks.) P1003 quickly split into two sub-projects: P1003.1 for the operating system interface, and P1003.2 for the shell and tools. (Recently, these have come to be known as POSIX.1 and POSIX.2.) A working group was associated with each. The working groups were named after the projects: 1003.1 and 1003.2.

This splitting has continued, with over 20 projects currently active. Whenever a possible new POSIX-related standards activity is identified, its promoters can draw up a Project Authorization Request (PAR), and submit it to the Sponsor Executive Committee (SEC) of TCOS¹. If approved (sponsored in IEEE terminology), and subsequently rubber-stamped by the IEEE Computer Society's Standards Activities Board (SAB), a new project is created. Most become sub-projects of the original 1003 project; some initiate new projects, such as P1201 on windowing environments.

If the subject of a new activity is closely associated with the interests of an existing working group, it is assigned to that group; if it is not, a new working group is set up. This means that there are fewer working groups than projects. As an example, the 1003.0 working group is concerned solely with the 1003.0 guide to the POSIX environment, but the 1003.1 working group now handles 1003.1, the operating system interface; 1003.16, C language bindings to operating system services; and 1003.18, a profile for a time-sharing POSIX-based system.

Once a working group has been formed, its job is to draft standards, making sure that they meet the needs of both suppliers and users of information technology. This is done through a somewhat baroque balloting process:

- Associated with each working group is a balloting group. The balloting group is typically formed shortly before the circulation of the first complete draft of the first standard developed by the working group.
- Balloting groups are drawn from the membership of a balloting pool. The pool has three types of member: individual members of the IEEE who have specifically applied to join the pool²; institutional representatives (IRs) accepted by the IEEE-CS SAB (see below); and national heads of delegation to the ISO POSIX

1. SC21, which is responsible for the higher layers of OSI, for SQL and for office document architectures and the like, might have been a candidate, but, after a false start with OSCRL (see my last column), was not interested.
2. Although I can tell you that ISO 9899, the C standard, went to the printers late in 1990, but, at the time of writing, has yet to emerge. It is functionally identical to the U.S. standard, ANSI X3.159-1989.
1. PARs can also be used to request changes to the goals and terms of reference of existing projects.

working group.

- All members of the balloting pool are sent notice of the formation of each new balloting group. Those who respond become members of the group, subject to considerations of maintaining a balance between user and supplier representatives.
- Once a balloting group has been formed, it persists indefinitely with a static membership. Only if there are problems in getting the required 75% response to ballots is the membership of a group reviewed.
- It is almost never possible to join a balloting group after it has formed.
- Individuals or organisations outside the balloting group can make objections to, or comments on, the content of draft standards, just as can balloting group members. All objections from whatever source must be handled through a formal resolution process. However, only members of the balloting group can vote for or against the acceptance of a draft (or indeed, completed) standard.
- A draft is considered approved if it is accepted by 75% or more of those who vote either for it or against it³.

Simple, huh? And I haven't even mentioned the appeals procedure!

Membership of a balloting group is a considerable responsibility: following notice of a ballot, IEEE rules give just 30 days to review a document which may run to almost a thousand pages, and to return any comments or objections to the ballot coordinator. And unless over 75% of the membership of the ballot group responds, the result is held to be invalid. When one considers that a document is likely to go through a dozen drafts before it becomes an approved standard, it is clear that balloters have to work hard (even if not all of the drafts are balloted). Recirculation ballots, initiated when changes are made to a draft in response to an initial ballot, increase the work-load further.

In order to make the task a little easier, TCOS has adopted a procedure called a mock ballot to handle the early drafts of a document. These are similar to mock examinations: the procedures are identical to the real thing, but it doesn't matter so much if it is flunked. In particular, no alarm bells start ringing in the IEEE's offices if a 75% response is not achieved.

What has all this to do with EurOpen?

EurOpen feels that it is important that the views of its membership are represented in two forums. Firstly, on the SEC, which decides on the authorization of POSIX-related projects and controls their development and coordination; and secondly, in the balloting pool from which those who vote on the content and acceptance of standards are drawn.

2. The requirement for IEEE membership appears recently to have been dropped, although the rule book has yet to be amended.
3. If more than 30% of those who return their ballots abstain, things get more complicated. Let's not go into that.

The first objective has already been met: I am happy to be able to tell you that the SEC has unanimously accepted EurOpen's request for me to become its institutional representative¹. I join existing IRs from a number of user groups and industry bodies: OSSWG (a group developing a real-time kernel for embedded systems), SHARE (the IBM user group), UniForum, UNIX International, USENIX and X/Open². (UniForum and USENIX were particularly helpful in the preparation of EurOpen's application.)

Gaining IR status in the balloting pool takes longer, as EurOpen's request must be discussed by the SAB, but I hope to be able to report in the Spring Newsletter that it has been approved.

-
1. Actually, the acceptance was "by acclamation", which is even better.
 2. The Free Software Foundation (FSF) is likely to join the list later this year.

Luckily, this delay gives me a little breathing space to make a request. I need help from volunteers. If you feel competent to help EurOpen's newly-formed Standards Activities Management Group (SAMG) in formulating responses to IEEE POSIX ballots, please contact me at the mail address at the head of this article³. In particular, could experts on secure operating systems please get in touch, as the working group concerned with this aspect of POSIX, 1003.6, is in the process of forming a balloting group.

I hope to see you at the standards birds-of-a-feather session at EurOpen's spring conference in Tromsø, where members of the SAMG will be reporting on the latest developments in the Europe, the U.S.A. and the world at large.

-
3. The other members of the SAMG are Johan Helsingius (julf@penet.fi) and Henk Hesselink (henk@ace.nl).



Management Committee

Minutes of the meeting 14th February, 1991

Present: Andrew Gollan, Chris Maltby, Pat Duffy, Frank Crawford, Peter Barnes. New committee member Peter Karr was welcomed to the meeting. Also attending were AUUGN editor David Purdue, and ACMS principal Wael Foda. Meeting commenced at 10:40.

1 Apologies

Stephen Prince, Michael Tuke.

2 Minutes of last Meeting (21st November, 1990)

- 2.1 Frank Crawford omitted from attendees, when he was only late.
- 2.2 Re 5 The committee voted its thanks to Mike Lawrence (Webster Computer) for forwarding AUUG mail (such as ;login).
- 2.3 Re 4.1 AG suggests that "miser." is an unacceptable abbreviation of "misere".
- 2.4 Re 10.5 Band for the Conference dinner is to be the Conway Hiccup (*sic*) Orchestra.
- 2.5 Re 7.1: only printing costs would be reduced.
- 2.6 Re 7.4: "would" should read "will". The MC approved this technique.
- 2.7 Re 7.2: ISSN
- 2.8 Re 9.4: delete this typo
- 2.9 Add 15.6: The Opus Group has booked the back page of AUUGN for six issues.
Moved (FC/SM) That the minutes as amended be accepted. Carried.

3 Business arising from the Minutes

- 3.1 Re 3.1, discussed in President's report.
- 3.2 Re 7.3 no issue yet.
- 3.3 Re 9.3, no report from Glenn Huxtable. FC reports that Perth, Melbourne and Hobart are going ahead.
- 3.4 Re 12 no change.
- 3.5 Re 13 discussed under 13.
- 3.6 Re 15.1 CM explained that AARNet was levying a charge for third party traffic, to be instituted as a charge for namespace (MX record), with graded costs of \$1,000 and \$4,000 a year for small and large users respectively. Proposal was that AUUG would purchase namespace for \$4000, and be given the right to register names, unlimited in the first year.
- 3.7 Re 15.2 the bulk of the material would be transferred to Softway, a small amount to ACMS for immediate stock, and some copies to libraries.



4 President's Report

Pat Duffy reported:

4.1 CM, PD and PB had all attended UniForum '91.

There was interest whether Usenix has an affiliate program.

New action: AG

4.2 It was noted that we must gain 75% approval in ballot for affiliation.

4.3 We need to review and improve membership benefits.

4.4 UI Australia is now reaching the end of its first year.

Moved (PB/AG) That the President's report be accepted. Carried.

5 Secretary's Report

Peter Barnes reported:

5.1 A membership report was not possible, as the database was in transit. There had been no updates to the information frozen in December. He had received many phone and e-mail membership enquiries, as there had been a long time between newsletters, and we do not currently issue a receipt for membership dues. According to WF, the new disk with the membership information arrived today (14/2).

5.2 There was no correspondence.

5.3 There had been two User Group related meetings at UniForum. The first had been called at short notice, and an invitation had not reached AUUG; this meeting resulted in the formation of a loose alliance of UNIX user groups, the impetus coming from X/Open and the EC. Our name has been added to the list of interested groups.

The second meeting had been the Affiliates meeting the day after the conference. This meeting consisted primarily of status reports from affiliate groups from all round the world. John Hosvath also announced UniForum's intention to move towards becoming an umbrella body, with UniForum U.S. as another affiliate. A President's advisory committee was established, consisting of interested international affiliate representatives. Both PD and PB had to leave to catch planes before the meeting closed.

5.4 The UniForum conference structure was quite different from our current structure, and seemed to favour quantity rather than quality. With many parallel sessions and relatively lightly vetted material, it was difficult to predict interesting or valuable sessions. However, the morning plenaries and afternoon streams seemed quite successful.

Moved (PD/CM) That the Secretary's report be accepted. Carried.

6 Treasurer's Report

Michael Tuke was absent.

11 Secretariat

12.1 The membership database disk could not be read by WF's version of the database software. It was agreed that this should be fixed as quickly as possible, or the database re-entered manually.



13 Other Business (1)

13.1 PK offered to publish a membership form and information in /aur (now Open Systems Review)

8 AUUG '90

WF reported:

8.1 MT has some more bills. We are still owed approximately \$1500. It was agreed we should request payment once more, and failing that, publish the names of the debtors on AUUGN.

WF tabled the current budget.

The meeting continued over lunch.

10 AUUG '91

10.1 WF tabled the projected budget.

10.2 Open Systems Review will be the official publication for the conference.

10.3 There was a discussion about promotion and advertising, and whether the conference and exhibition advertising should be coupled. It was agreed that they should have a consistent "look and feel".

10.4 AG has compiled most of the CFP, but needs a theme. "What is an Open System, anyway" suggested. It was agreed we would proceed with that idea. The CFP would also include mention of Work-in-progress and Birds-of-a-Feather sessions.

10.5 AG tabled a draft timetable.

10.6 It was agreed that we should present 2 one day and six half day tutorials.

10.7 It was noted that we had no press time, and that we should provide a press room for interviews, announcements and so forth. It was important that this should be properly structured and controlled.

10.8 It was agreed we should invite exhibitors to set up hospitality suites (as at Usenix and UniForum).

10.9 AG reported that he wanted to provide dual tracks if possible, perhaps by inviting the same talk twice (from different perspectives). An appropriate gift or memento should be offered as an incentive. We should also offer BOF space for vendor specific presentations, probably in parallel.

10.10 There was discussion about the dearth of software firms exhibiting. Possible solutions included a single software booth, or software BOFS.

13 Other Business (2)

13.1 We will present a President's page in OSR.

13.2 PD tabled a proposal for a membership survey, costed at \$4250, including telephone followup to 100 members and 50 non-members. After discussion it was suggested that the questionnaire might be published in OSR.

13.3 PD tabled a proposal that Symmetry Design (Ellen Gubbins and Joe Watkins) be retained to act as Press agents for AUUG. Fees would be \$1,500 a month.

Moved (AG/PK) That AUUG employ Symmetry Design on a trial basis from March to September inclusive. Carried.

PD left the meeting, and CM assumed the chair.



12 Networking (1)

12.1 Geoff Huston gave a history of AARNet. He explained that AARNet wished to recoup the costs of mail associates, and had decided to licence MX records, with costs on a sliding scale depending on usage.

The lowest cost (for the lowest volume users) was \$1000, but AARNet were prepared to sell records "in bulk" to interest groups like AUUG, who could then take care of the secretarial details, and pass on the savings to members. He proposed that in the first year AUUG would pay AARNet \$4000 one-off, and in subsequent years would take a proportion of AUUG's income from such licences.

AARNet would take responsibility for monitoring usage, and would inform AUUG if any low-use licences had to be upgraded. The cutover point (at present) was \$1000 in OTC or equivalent charges.

Moved (PB/AG) That AUUG should pay AARNet \$4000 for ACSnet MX registration for member sites. Carried.

Geoff Huston was thanked for his time, and left the meeting.

Moved (AG/FC) That AUUG set fees at \$250 for members, \$600 for non-members, where the member must own all the machines in the licenced domain. Carried.

CM volunteered to coordinate the scheme.

New action: CM

7 AUUGN Editor's Report

David Purdue reported:

7.1 V11n4 was at the printer

7.2 Postscript versions of minutes of meetings were still to come.

7.3 Alain Williams from EurOpen Newsletter had offered to place ads for us in their publication.

7.4 Suggestion that we might produce an Institutional Members' issue, with a half page to a page per member.

New action: DP

7.5 Inquiries about back-issues to be handled by Wael.

Moved (PK/FC) That the AUUGN Editor's report be accepted. Carried.

9 Summer Meetings

9.1 No report.

12 Networking (2)

12.1 FC reported slow response to ACSnet survey. Only 11 responses so far.

13 Other Business (3)

13.1 AG reminded the committee that there were two AUUG computers at Softway.

Moved (PK/AG) That AUUG ship the Fujitsu UNIX machine to the Secretary. Carried.



13.2 Suggestion that we provide membership cards. PB to investigate.

New action: PB

13.3 AG suggested that we provide a facility for automatic membership renewal using automatic credit card debit.

13.4 Glenn Huxtable or PD to write a letter thanking Amanda Moore for organising Melbourne Summer '91, and a letter to be published in AUUG'N.

14 Next Meeting

April 26th, PB to advise.

New action: PB

The meeting closed at 16:20



Management Committee Minutes of the meeting 26th April, 1991

Present: Andrew Gollan, Pat Duffy, Frank Crawford, Peter Barnes, Scott Merrilees. Also attending were ACMS principal Wael Foda, Symmetry Design's Ellen Gubbin, and MHSnet's Piers Lauder. Meeting commenced at 10:45. Peter Karr joined the meeting later.

1 Apologies

Chris Maltby, David Purdue, Stephen Prince, Michael Tuke.

2 Minutes of last Meeting (14th February, 1991)

2.1 13.2 "publisher" should be "published".

Moved (PK/FC) That the minutes as amended be accepted. Carried.

3 Business arising from the Minutes

3.1 Re 13.1: Open Systems Review back page looks good, although the writing/publishing delay means that material in there has to be carefully timed.

There will be no charge for our insert in OSR, just print costs. Circulation is about 10,000 direct, and about 4,000 on newsstands. Completed applications should come to the secretariat for processing, and the questions retained for review.

There will be an article by Geoff Huston in July OSR, PK suggests we add a short companion piece.

3.2 Glenn Huxtable to be asked to write a letter thanking Summer organisers (see previous minutes 13.4)

New action: Glenn Huxtable

3.3 Re 10.8 Wael will do an exhibitor briefing.

3.4 Re 10.10 CMP will possibly set up a joint booth and coordinate it if feasible.

3.5 Re 5.1 Wael has the membership database, is working with FC to finish preening it.

4 President's Report

Pat Duffy reported:

4.1 The column in OSR was going well.

4.2 She has been liaising with Ellen Gubbin from Symmetry Design.

4.3 She has been attending Programme Committee meetings

4.4 She has had two meetings with the A.C.S. One was face-to-face with Julian Day, chair of the N.S.W. branch, who wanted to know if we were interested in affiliation. The second was a phone call from Karl Reed (Director of the Technical Board) suggesting that we set up a joint committee.

New action: AG to investigate 2.5% training qualification.

New action: SM to investigate DECUS 2.5% training qualification.

New action: PD to prepare material on UniForum affiliation to be sent with ballots.

Moved (FC/PB) That the President's report be accepted. Carried.



5 Secretary's Report

Peter Barnes reported:

5.1 He would (regretfully) not be standing for re-election; personal and work commitments were too heavy.

5.2 Membership currently stood at: 230 Members, 105 Institutional, 5 Student, 2 AUUGN subscriptions, 1 Life Member, 14 Complimentary members. There are about 200 unfinancial members.

Moved (FC/SM) That unfinancial members not have their membership backdated (although founding members will not lose their status). Carried.

5.3 Correspondence: Letter from X/Open inviting us to join Xtra '91, a user-driven requirements gathering process.
New action: PB to get more details from John Totman.

5.4 PB apologised for inactivity since last meeting.

5.5 There has been a delay in call for nominations.

New action: PD to write letters about unfinancial members and renewals.

New action: PB to generate nomination form.

Moved (AG/FC) That the Secretary's report be accepted. Carried.

Wael Foda reported from the Secretariat:

5.6 Wael needs to be kept informed about events.

New action: AG to set up account at Softway for WF and PK.

5.7 We need to act on storage problems.

New action: PB to remind SP.

6 Treasurer's Report

Michael Tuke reported (by fax):

6.1 AUUG '90 is closed.

6.2 We made a profit on Melbourne and Perth Summer meetings.

New action: SM to promote video of Melbourne meeting.

6.3 Account signatories are not up to date. Perhaps update after elections.

6.4 Our financial year ends in May.

New action: PD to fax response to report, and request MT to set up CBA CM account..

Moved (PK/FC) That the Treasurer's report be accepted. Carried.

7 AUUGN Editor's Report

There is no editor's report. The next AUUGN is promised "in one to two weeks".

8 AUUG '91

AG reported:



8.1 There had been a rush before the old deadline — we now have enough to run a single stream. In total, about 15 “pure” submissions.

There have been many company expressions of interest (IBM, Sun ICL, Unisys, DEC, MIPS, HP, Sequent, Pyramid). We are still short on tutorials.

We have 1 guest speaker confirmed (Rob Pike) and one probable (Rob Asente). Evi Nemeth was also to be approached.

There is a request from Tandem for credit on the front cover. It was decided to refuse this.

New action: WF to advise.

There was a suggestion that we might make audio and/or video records of both tutorials and conference for resale, or instead of/in addition to Proceedings.

9 Networking

9.1 CM has sent out the announcement, there have been some forms returned.

*New action: AG to ask CM to mail *.oz*

10 Other Business

10.1 Ellen Gubbin for Symmetry Design presented a PR strategy for AUUG including support for the Conference. She proposed distributing a poster to Government Departments.

New action: PB to get Qld Depts.

For the conference, S.D. will do text slides, but not complex graphics; speakers will have to provide those themselves.

New action: PB provide Ellen with A.C.N.

New action: PB to register Australian Open Systems User Group.

Amendments to the plan suggested were:

1. No banner
2. 1,000 internal signage
3. 10,000 marketing in addition to Wael.
4. 7,500 speaker support.

Moved (AG/SM) That we accept the PR plan as amended. Carried.

10.2 Constitutional changes: we do not appear to have a mechanism for removal of delinquent office bearers. The current requirement for affiliation is too strong.

New action: PB to check articles of incorporation.

New action: PB to note nominations for Life Membership.

10.3 Conference Fees. These will be set as: \$160 for half day tutorial, \$240 for full day, \$395 for members conference, \$545 for non-members, \$225 for day attendance.

Moved (PK/FC) That conference fees be set as detailed. Carried *nem con.*

11 Next Meeting

June 24th.

New action: PB

Meeting closed at 16:30.

Management Committee

Minutes of the meeting 28th June, 1991

Present: Pat Duffy, Stephen Prince, Michael Tuke, Chris Maltby, Frank Crawford, Andrew Gollan. Meeting commenced at 10:30am. Scott Merrilees and Peter Karr joined the meeting later. Also present were ACMS principal Wael Foda and Ellen Gubbin of Symmetry Design. Rolf Jester, Secretary-elect, attended by invitation of Pat Duffy.

1. Apologies

Peter Barnes.

2. Minutes of last Meeting (26 April 1991)

Moved (AG/FC) that the minutes be accepted. Carried.

3. Business arising from the Minutes

- 3.1 Re 3.2: Stephen Prince will ask Glenn Huxtable to write a letter thanking the Summer organisers.

Action: SP

4. President's Report

Pat Duffy reported:

- 4.1 She now sends a letter welcoming each new member and thanking each renewing member.

4.2 She has contacted the organisers of the "Australian Users group for Open Systems". Their aims are not those of a Users Group, but more like those of the "Houston 30" - i.e. to influence the standards setting bodies. There seems to be little conflict with AUUG, and it has been agreed to exchange information and attend one another's meetings. The consensus of the current meeting, nevertheless, was that we should in future examine the potential for asking such a group to join forces with AUUG.

4.3 Recent publicity has led to a number of membership enquiries.

Moved (AG/CM) that the President's report be accepted. Carried.

5. Secretary's Report

5.1 Peter Barnes being absent, there is no Secretary's Report.

Wael Foda reported for the Secretariat:

5.2 The membership database is now up-to-date except that the additional newsletter recipients of Institutional members need to be added. ACMS will send out a form with the renewals letter, seeking to verify the database information.

5.3 Membership as at 28 June 1991.

Category	Financial	Unfinancial	Total
Institutional	146	61	207
Members	219	140	359
Students	6	4	10
Life Members	1		1
Subscriptions	4	14	18

TOTAL	376	219	595
=====			

6. Treasurer's Report

Michael Tuke reported:

6.1 Bank balance as at 3 June 1991 was \$123,865, plus \$29,497 in a term deposit with the Commonwealth Bank and \$6,000 with Chase.

6.2 There is a need to change the cheque signatories. [The forms were completed during the meeting.]

- 6.3 Michael Tuke will pay the speaker's costs (airfare etc.) for James Pinakis, last year's Student Speaker, as part of Conference Costs.

Action: MT

Moved (FC/SP) that the Treasurer's Report be accepted. Carried.

7. AUUGN Editor's Report

- 7.1 Jagoda Crawford being absent, there is no AUUGN Editor's report. Frank Crawford will ask David Purdue to put out the next issue (1991 #1) as a matter of urgency, and for Jagoda Crawford to issue the next one (1991 #2) as soon as possible thereafter.

The next issue will contain Pat Duffy's report on what has (or hasn't) been happening. The one after that will contain details of the changes to membership renewal (see below). There should be no further Call For Papers.

Action: FC

- 7.2 It was agreed that at the next meeting, there would be a review of the AUUGN target audience, objectives, format, contents and production. As a principle, it was generally agreed that the audience for which AUUGN is intended is technical, those who require more detailed information than can be found in the general IS media. A key question to be addressed is how we can effectively meet the information needs of the commercial user. It is possible that design and production could be contracted to specialist organisations.

All Management Committee members will prepare ideas for this discussion by next meeting.

Action: All

8. Returning Officer's report.

None.

9. Membership renewal and fees.

Moved (MT/CM) that membership fees and renewals be set as detailed below. Carried.

- 9.1 Membership renewals will in future fall due on either January 1 or July 1 depending on which date is nearest to the member's current renewal date. There will be no pro-rata fees, and no charge for the period from the member's current renewal date to the new renewal date - i.e. up to three months free.
- 9.2 Members who are now unfinancial will be billed as at July 1.
- 9.3 Membership fees will remain unchanged.
- 9.4 Pat Duffy will write for the 1991 #2 AUUGN issue explaining the changes.

Action: PD

- 9.5 Wael Foda will draft a letter for members explaining the changes, and will implement the new procedure.

Action: WF

- 9.6 Peter Karr reports that the July issue of Open Systems Review will include an AUUG application form and a survey, the results of which will be provided by CMP to AUUG at a cost of around \$500-700.

10. Network

- 10.1 Chris Maltby reports that we have signed up nearly 40 AARNET subscriptions, earning us a profit of at least \$6,000.

- 10.2 Peter Karr will submit the AARNET article for /osr to Chris Maltby for review and editing.

Action: PK
and: CM

- 10.3 We need a simple information sheet - "How to get on the network."

Action: CM
and: FC

- 10.4 It was agreed that we will sponsor an AARNET connection for the Sun User Group in Melbourne.

Action: SP

- 10.5 Stephen Prince will update the AUUG Executive electronic mailing list.

Action: SP

11. Publicity

- 11.1 Ellen Gubbin reports that the press release regarding AUUG'91 speakers received coverage in one publication. The next releases are about the AUUG'91 Program and XTRA'91.

- 11.2 Ellen Gubbin will quote on a Membership Brochure, Membership Card and Institutional Member Certificate to be issued at AUUG'91.

Action: EG

11.3 It was agreed that we would consider a 6-monthly publication in addition to AUUGN. The new publication would possibly have these characteristics:

- Oriented more toward commercial users and IS Managers;
- Act in a similar fashion to an annual report;
- Be used as an aid to recruitment;
- Be offered with subscription renewals;
- Contents could include: industry trend/comment articles, AUUG financials, resource directory, standards information, market statistics, summary of reports like the DMR study.
- Qty: 2000.

Ellen Gubbin will quote on this publication.

Action: EG

11.4 The fees for Symmetry will remain unchanged until the end of the agreed six month period.

12. AUUG'91

12.1 Andrew Gollan reports that the program is full, with about 40 speakers accepted, and dual streams (technical/commercial) for the 2-4pm sessions.

12.2 Sun and IBM have not yet submitted names for their corporate speakers. Andrew Gollan will request names immediately or offer the slot to someone else.

Action: AG

12.3 Andrew Gollan will send out letters to speakers accepted and rejected, and send out speakers kits.

Action: AG

12.4 The morning and afternoon coffee breaks need to be extended to allow delegates to walk to the Exhibition area.

Action: AG

12.5 The AGM is Thursday September 26 at 6:00pm. There will be a Committee Meeting on Tuesday September 24 at 2:00pm in the Exhibition area in one of the conference rooms upstairs.

12.6 Ellen Gubbin will submit ideas for a \$5 delegate give-away and a suitable gift for speakers, including something appropriate for invited overseas speakers.

Action: EG

13. Review of Membership Benefits

13.1 Current benefits:

- a. AUUGN
- b. Discount for attendance at conferences.
- c. Discounts from certain Publishers (e.g. Prentice Hall) and other businesses.
- d. Reciprocal rights with USENIX and UniForum - attend events at member price.
- e. Reduced AARNET charge for Institutional members.
- f. Input into the X/Open XTRA process for Institutional members.
- g. Chapter activities (three Chapters).

13.2 Ideas for future benefits:

- a. Joint AUUG/UniForum membership. To be formalised and clarified.
- b. Possible association with other organisations, e.g. EurOpen.
- c. Discount (20%) for non-vendor members on the new CMP UNIX weekly newsletter may be offered by Peter Karr.
- d. Membership card to allow members to take advantage of discounts offered by businesses such as bookshops.
- e. Certificate for Institutional members.
- f. AUUG library of AUUGN, Conference Proceedings, and possibly other relevant literature to be preserved at ACMS or at Amdahl for use on site, not for loan out.
- g. Directory of UNIX/Open Systems resources - e.g. publications, contacts for relevant organisations, where to get more information. This could be a small publication maintained on paper and electronically and issued to all interested parties on request. Pat Duffy will collate initial ideas for this for review and adding to by all at the next meeting.

Action: PD
and: All

14. Other Business

- 14.1 Robert Ellis will register new business name and lodge the Constitutional changes with Corporate Affairs. Stephen Prince will follow up.

Action: SP

- 14.2 Rolf Jester will write a review of Pamela Gray's book "Open Systems - a Business Strategy for the 90s" for AUUGN and /osr.

Action: RJ

15. Next Meeting

Monday, 5 August.

Action: RJ

Meeting closed at 3:30pm

AUUG Membership Categories

Once again a reminder for all ‘members’ of AUUG to check that you are, in fact, a member, and that you still will be for the next two months.

There are 4 membership types, plus a newsletter subscription, any of which might be just right for you.

The membership categories are:

- Institutional Member
- Ordinary Member
- Student Member
- Honorary Life Member

Institutional memberships are primarily intended for university departments, companies, etc. This is a voting membership (one vote), which receives two copies of the newsletter. Institutional members can also delegate 2 representatives to attend AUUG meetings at members rates. AUUG is also keeping track of the licence status of institutional members. If, at some future date, we are able to offer a software tape distribution service, this would be available only to institutional members, whose relevant licences can be verified.

If your institution is not an institutional member, isn't it about time it became one?

Ordinary memberships are for individuals. This is also a voting membership (one vote), which receives a single copy of the newsletter. A primary difference from Institutional Membership is that the benefits of Ordinary Membership apply to the named member only. That is, only the member can obtain discounts an attendance at AUUG meetings, etc. Sending a representative isn't permitted.

Are you an AUUG member?

Student Memberships are for full time students at recognised academic institutions. This is a non voting membership which receives a single copy of the newsletter. Otherwise the benefits are as for Ordinary Members.

Honorary Life Membership is not a membership you can apply for, you must be elected to it. What's more, you must have been a member for at least 5 years before being elected.

It's also possible to subscribe to the newsletter without being an AUUG member. This saves you nothing financially, that is, the subscription price is greater than the membership dues. However, it might be appropriate for libraries, etc, which simply want copies of AUUGN to help fill their shelves, and have no actual interest in the contents, or the association.

Subscriptions are also available to members who have a need for more copies of AUUGN than their membership provides.

To find out if you are currently really an AUUG member, examine the mailing label of this AUUGN. In the lower right corner you will find information about your current membership status. The first letter is your membership type code, N for regular members, S for students, and I for institutions. Then follows your membership expiration date, in the format exp=MM/YY. The remaining information is for internal use.

Check that your membership isn't about to expire (or worse, hasn't expired already). Ask your colleagues if they received this issue of AUUGN, tell them that if not, it probably means that their membership has lapsed, or perhaps, they were never a member at all! Feel free to copy the membership forms, give one to everyone that you know.

If you want to join AUUG, or renew your membership, you will find forms in this issue of AUUGN. Send the appropriate form (with remittance) to the address indicated on it, and your membership will (re-)commence.

As a service to members, AUUG has arranged to accept payments via credit card. You can use your Bankcard (within Australia only), or your Visa or Mastercard by simply completing the authorisation on the application form.

AUUG Incorporated

Application for Newsletter Subscription

Australian UNIX* systems Users' Group.

*UNIX is a registered trademark of UNIX System Laboratories, Incorporated

Non members who wish to apply for a subscription to the Australian UNIX systems User Group Newsletter, or members who desire additional subscriptions, should complete this form and return it to:

AUUG Membership Secretary
 PO Box 366
 Kensington NSW 2033
 Australia

- Please don't send purchase orders — perhaps your purchasing department will consider this form to be an invoice.
- Foreign applicants please send a bank draft drawn on an Australian bank, or credit card authorisation, and remember to select either surface or air mail.
- Use multiple copies of this form if copies of AUUGN are to be dispatched to differing addresses.

This form is valid only until 31st May, 1992

Please *enter / renew* my subscription for the Australian UNIX systems User Group Newsletter, as follows:

Name: Phone: (bh)
 Address: (ah)

 Net Address:

 Write "Unchanged" if address has
 not altered and this is a renewal.

For each copy requested, I enclose:

- Subscription to AUUGN \$ 90.00
- International Surface Mail \$ 20.00
- International Air Mail \$ 60.00

Copies requested (to above address) _____

Total remitted AUD\$ _____

(cheque, money order, credit card)

Tick this box if you wish your name & address withheld from mailing lists made available to vendors.

Please charge \$_____ to my Bankcard Visa Mastercard.

Account number: _____ . Expiry date: ____/____.

Name on card: _____ Signed: _____

Office use only:

Chq: bank _____ bsb _____ - a/c _____ # _____

Date: ____/____/____ \$ _____ CC type ____ V# _____

Who: _____ Subscr# _____

AUUG Incorporated

Application for Institutional Membership

Australian UNIX* systems Users' Group.

*UNIX is a registered trademark of UNIX System Laboratories, Incorporated

To apply for institutional membership of the AUUG, complete this form, and return it with payment in Australian Dollars, or credit card authorisation, to:

AUUG Membership Secretary
 PO Box 366
 Kensington NSW 2033
 Australia

• Foreign applicants please send a bank draft drawn on an Australian bank, or credit card authorisation, and remember to select either surface or air mail.

This form is valid only until 31st May, 1992

..... does hereby apply for

- New/Renewal* Institutional Membership of AUUG \$325.00
- International Surface Mail \$ 40.00
- International Air Mail \$120.00

Total remitted

AUD\$ _____

(cheque, money order, credit card)

* Delete one.

I/We agree that this membership will be subject to the rules and by-laws of the AUUG as in force from time to time, and that this membership will run for 12 consecutive months commencing on the first day of the month following that during which this application is processed.

I/We understand that I/we will receive two copies of the AUUG newsletter, and may send two representatives to AUUG sponsored events at member rates, though I/we will have only one vote in AUUG elections, and other ballots as required.

Date: ___ / ___ / ___

Signed: _____

Title: _____

Tick this box if you wish your name & address withheld from mailing lists made available to vendors.

For our mailing database - please type or print clearly:

Administrative contact, and formal representative:

Name:

Phone: (bh)

Address:

..... (ah)

.....

Net Address:

.....

Write "Unchanged" if details have not

.....

altered and this is a renewal.

Please charge \$ _____ to my/our Bankcard Visa Mastercard.

Account number: _____

Expiry date: ___ / ___ .

Name on card: _____

Signed: _____

Office use only:

Please complete the other side.

Chq: bank _____ bsb _____ - a/c _____ # _____

Date: ___ / ___ \$

CC type ___ V# _____

Who: _____

Member# _____

Please send newsletters to the following addresses:

Name: Phone: (bh)
Address: (ah)
.....
..... Net Address:
.....
.....

Name: Phone: (bh)
Address: (ah)
.....
..... Net Address:
.....
.....

Write "unchanged" if this is a renewal, and details are not to be altered.

Please indicate which Unix licences you hold, and include copies of the title and signature pages of each, if these have not been sent previously.

Note: Recent licences usually revoke earlier ones, please indicate only licences which are current, and indicate any which have been revoked since your last membership form was submitted.

Note: Most binary licensees will have a System III or System V (of one variant or another) binary licence, even if the system supplied by your vendor is based upon V7 or 4BSD. There is no such thing as a BSD binary licence, and V7 binary licences were very rare, and expensive.

- | | |
|--|--|
| <input type="checkbox"/> System V.3 source | <input type="checkbox"/> System V.3 binary |
| <input type="checkbox"/> System V.2 source | <input type="checkbox"/> System V.2 binary |
| <input type="checkbox"/> System V source | <input type="checkbox"/> System V binary |
| <input type="checkbox"/> System III source | <input type="checkbox"/> System III binary |
| <input type="checkbox"/> 4.2 or 4.3 BSD source | |
| <input type="checkbox"/> 4.1 BSD source | |
| <input type="checkbox"/> V7 source | |
| <input type="checkbox"/> Other (<i>Indicate which</i>) | |

AUUG Incorporated

Application for Ordinary, or Student, Membership

Australian UNIX* systems Users' Group.

*UNIX is a registered trademark of UNIX System Laboratories, Incorporated

To apply for membership of the AUUG, complete this form, and return it with payment in Australian Dollars, or credit card authorisation, to:

AUUG Membership Secretary
P O Box 366
Kensington NSW 2033
Australia

- Please don't send purchase orders — perhaps your purchasing department will consider this form to be an invoice.
- Foreign applicants please send a bank draft drawn on an Australian bank, or credit card authorisation, and remember to select either surface or air mail.

This form is valid only until 31st May, 1992

I, do hereby apply for

- Renewal/New* Membership of the AUUG \$78.00
- Renewal/New* Student Membership \$45.00 (note certification on other side)
- International Surface Mail \$20.00
- International Air Mail \$60.00 (note local zone rate available)

Total remitted AUD\$ _____
(cheque, money order, credit card)

* Delete one.

I agree that this membership will be subject to the rules and by-laws of the AUUG as in force from time to time, and that this membership will run for 12 consecutive months commencing on the first day of the month following that during which this application is processed.

Date: ___ / ___ / ___ Signed: _____

Tick this box if you wish your name & address withheld from mailing lists made available to vendors.

For our mailing database - please type or print clearly:

Name: Phone: (bh)
 Address: (ah)

 Net Address:

 Write "Unchanged" if details have not altered and this is a renewal.

Please charge \$_____ to my Bankcard Visa Mastercard.
 Account number: _____ Expiry date: ___/___.
 Name on card: _____ Signed: _____

Office use only:
 Chq: bank _____ bsb _____ - a/c _____ # _____
 Date: ___ / ___ / ___ \$ CC type ___ V# _____
 Who: _____ Member# _____

Student Member Certification *(to be completed by a member of the academic staff)*

I, certify that
..... *(name)*
is a full time student at *(institution)*
and is expected to graduate approximately ____/____/____.

Title: _____

Signature: _____

AUUG

Notification of Change of Address Australian UNIX* systems Users' Group.

*UNIX is a registered trademark of UNIX System Laboratories, Incorporated

If you have changed your mailing address, please complete this form, and return it to:

AUUG Membership Secretary
PO Box 366
Kensington NSW 2033
Australia

Please allow at least 4 weeks for the change of address to take effect.

Old address (or attach a mailing label)

Name: Phone: (bh)

Address: (ah)

..... Net Address:

.....

.....

.....

New address (leave unaltered details blank)

Name: Phone: (bh)

Address: (ah)

..... Net Address:

.....

.....

.....

Office use only:

Date: ___/___/___

Who: _____

Memb# _____

