

AUUGN

AUUG Inc. Newsletter

Volume 15, Number 4

August 1994

The AUUG Incorporated Newsletter

Volume 15 Number 4

August 1994

CONTENTS

AUUG General Information		3
Editorial		5
AUUG President's Page		6
AUUG Corporate Sponsors - A New Level of Participation		7
AUUG Institutional Members		8
Announcements		
Affordable Internet Access for Individuals and Small Business		11
Call for Articles for the Australian		12
New AUUGN Editor Wanted		13
GURU's 1994 Open Systems Conference & Exhibition		14
USENIX Conferences		16
AUUG Local Chapters		
AUUG Regional Contacts		17
Queensland Chapter Update	<i>Mark White</i>	18
Canberra Chapter		19
NSW Chapter		19
AUUG Inc. - Victorian Chapter		20
Victorian Chapter Activities	<i>Enno Davids</i>	21
From the Western Front	<i>Janet Jackson</i>	22
WAUG - Meeting Reviews	<i>Adrian Booth, Janet Jackson</i>	23
SAGE		
Please Unsubscribe Me from this List..		24
SAGE-AU Code of Ethical Conduct	<i>Kate Lance</i>	25
AARNet Mail Affiliation		36
Open System Publications		40
Review of the AIR 2.0 TCP Software Package for Windows	<i>Chris Maltby</i>	41
Book Reviews		43
Prentice Hall Book Order Form		49
WoodsLane Information		50
Addison-Wesley Information		51

!AUUGN - from AUUGN Volume 1, Number 1	
A letter from Alistair Kilgour	52
Technical Reports	
C++ - is it really a better C? <i>Michael Henning</i>	54
Managing Mailing Lists with Majordomo <i>Frank Crawford</i>	57
Considerations and Issues when Implementing a Document Management System <i>Phil Barton</i>	63
From login: - Volume 19, Number 3	
An Update on UNIX-Related Standards Activities	75
From login: - Volume 19, Number 4	
An Update on UNIX-Related Standards Activities	79
Unix Tricks & Traps <i>Graham Jenkins</i>	83
Articles from the Australian Newspaper	85
Summary of Management Committee Minutes - 18th July 1994	95
AUUG Membership Categories	98
AUUG Forms	99
ADVERTISEMENTS	
Softway	13
ZIRCON Systems	41
Prentice Hall Book Order Form	49
WoodsLane Information	50
Addison-Wesley Information	51

Copyright © 1994 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 X/Open in the United States and other countries.

AUUG General Information

Memberships and Subscriptions

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

Membership and General Correspondence

All 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
Freephone: 1-800 625 655
Email: auug@munnari.oz.au

AUUG Business Manager

Catrina Dwyer,
P.O. Box 366,
Kensington, N.S.W. 2033.
AUSTRALIA

Phone: (02) 959 3656
Fax: (02) 957 6706
Email: catrina@sw.oz.au

AUUG Executive

President **Phil McCrea**
pmc@atom.ansto.gov.au

Watch this space ...

Vice-President **Glenn Huxtable**
glenn@cs.uwa.edu.au
University of Western Australia
Computer Science Department
Nedlands WA 6009

Secretary **Peter Wishart**
peter.wishart@csis.dit.csiro.au
CSIRO Div. of Information Technology
GPO Box 664
Canberra ACT 2601

Treasurer **Frank Crawford**
frank@atom.ansto.gov.au
ANSTO
Private Mail Bag 1
Menai NSW 2234

Committee Members **Stephen Boucher**
stephen@mtiame.mtia.oz.au
MTIA
509 St. Kilda Rd.
Melbourne VIC 3004

Lucy Chubb
lucyc@softway.sw.oz.au
Softway Pty. Ltd.
P.O. Box 305
Strawberry Hills NSW 2021

Chris Maltby
chris@softway.sw.oz.au
Softway Pty. Ltd.
P.O. Box 305
Strawberry Hills NSW 2021

Michael Paddon
mwp@munnari.oz.au
Kodak
173 Elizabeth St
Coburg, Vic 3058

Rick Stevenson
rick@stallion.oz.au
Stallion Technologies Pty. Ltd.
56 Sylvan Rd.
Toowong, QLD 4066

AUUG General Information

Next AUUG Meeting

The AUUG'94 Conference and Exhibition will be held from the 7th to 9th September, 1994, at the World Congress Centre, Melbourne.

Advertising

Advertisements to be included in AUUGN are welcome. They should conform to the standards of other contributions (see page 5). Advertising rates are \$120 for a quarter page, \$180 for half a page, \$300 for the first A4 page, \$250 for a second page, \$500 for the inside cover 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 business manager 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:

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

Conference Proceedings

A limited number of the Conference Proceedings for AUUG'92 and AUUG'93 are still available, at \$50 for members and \$60 for non-members. Contact the AUUG secretariat.

Acknowledgement

This newsletter was produced with the kind assistance of and on equipment provided by the Australian Nuclear Science and Technology Organisation. A copy of FrameMaker for use in the production of the newsletter has been provided by Platform Technologies .

Disclaimer

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

* Platform Technologies are no longer distributors of FrameMaker, Information Technology Consultants, in Paddington NSW are now distributing FrameMaker.

AUUG Newsletter

Editorial

Welcome to AUUGN Volume 15 Number 4. This should reach you just as you are packing for AUUG94. Unfortunately, I cannot make it to the conference, so if you attend, why not submit a report on the conference so that those unable to attend can read about it.

In addition to the regular features, such as the Chapter reports and the AUUG related announcements, we have details on a number of conferences, papers from the various Summer Conferences, and some reprints from the proceedings of SAGE-AU'94.

We have also received a large number of book reviews, some of which are published here, others will be in the next issue. Note that we now review Addison-Wesley books and are presently organising discounts with them.

Finally, I have reprinted a number of the AUUG columns from the Australian. These are as they were submitted, so you can see what the author originally wrote.

On a personal note, I am at a stage where my workload is such that I have to give up editing AUUGN. This is not intended to be immediately, but I am looking for someone to take over. If you are interested please see the advert in this issue.

Hope you enjoy the newsletter and the conference, and if you have anything of interest, be it technical or otherwise, please consider submitting it for publication.

Jagoda Crawford

AUUGN Correspondence

All correspondence regarding the AUUGN should be addressed to:-

AUUGN Editor,
P.O. Box 366,
Kensington, N.S.W. 2033.
AUSTRALIA

Phone: +61 2 717 3885
Fax: +61 2 717 9273
Email: auugn@munnari.oz.au

AUUGN Book Reviews

The AUUGN book review editor is Frank Crawford. Anyone interested in reviewing books or with book reviews to submit for publishing in AUUGN please contact Frank. His address can be found on page two of this issue. Remember, that any books you review, you keep.

Contributions

The Newsletter is published approximately every two months. The deadlines for contributions for the next issues of AUUGN are:

Volume 15 No 5 Friday 23th September
Volume 15 No 6 Friday 25th November

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 margins, and 30 mm left at the bottom so that the AUUGN footers can be pasted on to the page. Small page numbers printed in the footer area would help.

AUUG President's Page

Can the Internet provide Commercial services?

You can't pick up a newspaper these days without reading about the Internet. To those of us who have been long time UNIX users, it is a source of mild amusement to observe that the rest of the computing world has suddenly discovered the standard UNIX networking facility which we have taken for granted.

Heavens above, even Gareth Powell has discovered the Internet! And what's more, someone has pointed him in the direction of alt.flame.gareth-powell! (For those who are not regular readers of the Sydney Morning Herald, Gareth is the editor of the Monday Computer Section, who doubles up as a travel correspondent...)

It's interesting to watch what is happening at present, as AARNet contemplates its future plans, and rumours start to appear about commercial competitors to AARNet.

Let's consider the situation of 'selling' the Internet commercially. The first question asked is "What does it offer my organisation?". Try responding to that to someone who is keyboard-phobic, or (even worse) to a self-proclaimed expert who has recently mastered MS Word, and can now type a memo!

But it is an absolutely valid question. Use of the Internet won't really grow in the non-academic sector till the services that run on it are packaged appropriately. Current Internet users tend to be 'osmotic', and generally have a feel for what a network can provide. Perhaps the two largest groups who have discovered the Internet recently are librarians and journalists: librarians because of the wealth of databases and search facilities available, and journalists because of the accessibility of news articles and associated discussions on news groups.

In addition, general use of the Internet won't take off till PC users feel comfortable using it. This is where several local Internet 'retailers' are starting to be successful - a modem, some PC software, a few dollars a month, and hey presto you are net connected!

Let's look at the 'wholesale' side of things, which is currently monopolized by AARNet. The AARNet board comprises mainly Vice Chancellors from various Australian Universities, whose interest, understandably, is in providing electronic connectivity for their constituents - *i.e.* Universities and other (*e.g.* CSIRO) researchers.

AARNet is considering sweeping changes to how it charges users, and is favouring a move away from fixed bandwidth towards 'user pays'. This has not been lost on the ASTEC committee reporting on Research Data Networks, which has noted that such a charging system will have a detrimental effect on academic research. I agree with this: the current charging system serves academics well, although it does however encourage frivolous use of bandwidth, which is a problem.

The commercial world, however, is accustomed to a 'user pays' approach, and the resistance being expressed by academics to user pays will not be present in a business environment to the same extent.

Until recently, the AARNet Board has not actively encouraged business use of AARNet, although it has turned a blind eye to those organisations who have been using AARNet for business activity. The recent tacit approval of the AARNet board for use of the net by non-academic organisations, does not provide sufficient incentive for commercial organisations to start to use the Internet for more mission-critical (*sic!*) applications: industry will always be suspicious of something that is controlled by academics...

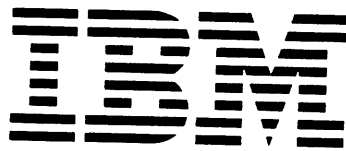
The climate therefore is right for an organisation, or organisations, to set up an alternative backbone to AARNet for commercial use, and to court large commercial organisations to start using it. It will have to be price-competitive with AARNet, and services will need to be packaged appropriately.

I am aware of several organisations contemplating setting up Internet backbones in Australia, and the interesting challenge at this point in time is "will it be profitable?". I believe it will - in the long term.

P. McCrea

AUUG Inc. Corporate Sponsors

AUUG Inc. is pleased to acknowledge the generous support given by the following corporate sponsors:



AUUG Institutional Members as at 12/08/1994

A. Goninan & Co. Limited
A.N.U.
AAII
Aberfoyle Resource Limited
ACAY Network Computing Pty.Ltd.
Actrol Parts
Adept Software
Advanced Software Engineering
Alcatel Australia
Amalgamated Television Services
Amdahl Australia Pty Ltd
Amdahl Pacific Services
Andersen Consulting
ANI Manufacturing Group
Animal Logic Research Pty. Ltd.
ANSTO
Anti-Cancer Council of Victoria
ANZ McCaughan
Atlas Computer Systems
Attorney-General's Department
AUSOM Inc.
Australian Archives
Australian Bureau of Statistics
Australian Centre for Remote Sensing (ACRES)
Australian Defence Industries Ltd.
Australian Electoral Commission
Australian Film Television and Radio School
Australian Information Processing Centre Pty. Ltd.
Australian Museum
Australian National Audit Office
Australian Submarine Corporation
Australian Taxation Office
Australian Technology Resources (ACT) Pty. Ltd.
Australian Technology Resources Pty. Ltd.
Australian Tourist Commission
Australian Wool Research & Promotion
 Organisation
B & D Australia
Bay Technologies Pty Ltd
BHA Computer Pty. Limited
BHP Information Technology
BHP Petroleum
BHP Research - Melbourne Laboratories
BHP Research - Newcastle Laboratories
Bond University
Burdett, Buckeridge & Young Ltd.
Bureau of Meteorology
Butterworths
Bytecraft Pty. Ltd.
C.I.S.R.A.
Cadcom Solutions Pty. Ltd.
Cape Grim B.A.P.S
Capricorn Coal Management Pty. Ltd.
CelsiusTech Australia
Chief Secretary's Department
CITEC
Clarity International
Classified Computers Pty. Ltd.
Clegg Driscoll Consultants Pty. Ltd.
Co-Cam Computer Group
Coal & Allied Operations
Cognos Pty. Ltd.
Com Net Solutions
Com Tech Communications
Commercial Dynamics
Communica Software Consultants
Composite Buyers Ltd.
Computechnics Pty. Ltd.
Computer De Tokyo Corporation
Computer Law Corporation
Computer Sciences of Australia Pty. Ltd.
Computer Software Packages
Computer Systems (Australia) Pty. Ltd.
Comsys International Pty. Ltd.
Concord Repatriation General Hospital
Copper Refineries Pty. Ltd.
Corinthian Engineering Pty. Ltd.
Corporate Workgroup Resources
CSIRO Division of Information Technology
CSIRO Division of Manufacturing Technology
CSIRO Division of Wool Technology
Curtin University of Technology
Customised Software Solutions Centre
Cyberdyne Systems Corporation Pty. Ltd.
Cyberscience Corporation Pty. Ltd.
Cybersource Pty. Ltd.
Data General Australia
Datacraft Limited
Datacraft Technologies
Dawn Technologies
DB Bain Group Services Pty. Ltd.
Deakin University
Defence Housing Authority
Defence Service Homes
Department of Business & Employment
Department of Defence
Department of Defence (TC Section)
Department of Education QLD
Department of Environment & Natural Resources
Department of Family Services & Aboriginal &
 Islander Affairs
Department of Planning & Development
Department of State Services
Department of the Treasury
Dept. of Industrial Relations, Employment,
 Training & Further Education
DEVETIR
Dialix
Digital Equipment Corp. (Australia) Pty. Ltd.
DSTO, Lab 73"
EASAMS (Australia) Limited
Edith Cowan University
Electricity Trust of South Australia
Electro Optics Pty. Ltd.
Engineering Computer Services Pty. Ltd.
Equity Systems Pty. Limited
ERIN Unit, Australian National Parks &
 Wildlife Service
ESRI Australia Pty. Ltd.
Executive Computing
FGH Decision Support Systems Pty. Ltd.
Financial Network Services
Fire Fighting Enterprises
First State Computing

AUUG Institutional Members as at 12/08/1994

Flinders University
Fremantle Port Authority
Fujitsu Australia Ltd.
G.James Australia Pty. Ltd.
GEC Alsthom Information Technology
GEC Marconi Systems Ltd.
Geelong & District Water Board
Genasys II Pty. Ltd.
General Automation Pty. Ltd.
GIO Australia
Golden Casket Office
Golden Circle Australia
Great Barrier Reef Marine Park Authority
Gribbles Pathology
Haltek Pty. Ltd.
Hamersley Iron
Heath Insurance
Hermes Precisa Australia Pty. Ltd.
Hollywood Private Hospital
Honeywell Ltd.
Hong Kong Jockey Club Systems (Australia)
Pty. Ltd.
I.P.S Radio & Space Services
IBA Healthcare Pty. Ltd.
IBM Australia Ltd.
Iconix Pty. Ltd.
Ideas International Pty. Ltd.
Independent Systems Integrators
Information Technology Consultants
Information Technology Services Branch,
Dept of Public Administration
Informed Technology
Insession Labs Pty. Ltd.
Insurance & Superannuation Commission
Integration Design Pty. Ltd.
International Imaging Systems
Internode Systems Pty. Ltd.
ISR Group Ltd.
James Cook University of North Queensland
JTEC Pty. Ltd.
Keays Software
Knowledge Engineering Pty. Ltd.
Laboratory Systems Pty. Ltd.
Labtam Australia Pty. Ltd.
Land Information Centre
Land Titles Office
Leeds & Northrup Australia Pty. Limited
Legent Australia Pty. Ltd.
Logica Pty. Ltd.
Lotus Development
Lyons Computer Pty. Ltd.
Macquarie University
Main Roads Western Australia
Matcom Technologies
Mayne Nickless Courier Systems
Mayne Nickless Information Technology Services
Medical Benefits Funds of Australia Ltd.
Memtec Limited
Mentor Technologies Pty. Ltd.
Mercedes-Benz (Australia)
Metal Trades Industry Association
Mincom Pty. Ltd.
Minenco Pty. Ltd.
Mitsubishi Motors Australia Ltd.
Mitsui Computer Limited
Moldflow Pty. Ltd.
Motorola Computer Systems
Motorola Communications Australia
MPA International Pty. Ltd.
MUA Pty. Ltd.
MUA Pty. Ltd.
MUA Pty. Ltd.
Multibase Pty. Ltd.
Multiline BBS
Multiuser Solutions Pty. Ltd.
National Library of Australia
National Resource Information Centre
NCR Australia
NEC Australia Pty. Ltd.
Northern Territory Library Service
Northern Territory University
Novell Pty. Ltd.
NSW Agriculture
NSW Teachers Federation Health Society
Object Design Pty. Ltd.
Object Oriented Pty. Ltd.
Object Technology International Pty. Ltd.
Ochre Development
Office of State Revenue
Office of the Director of Public Prosecutions
ONA
Open Software Associates Ltd.
Open Technology Pty Ltd
Opentec Pty Ltd
OPSM
OSIX Pty. Ltd.
OzWare Developments Pty. Ltd.
Pacific Semiconductor Pty. Ltd.
Pacific Star Communications
Petrosys Pty. Ltd.
Philips PTS
Port of Melbourne Authority
Powerhouse Museum
PPIT Pty. Ltd.
Process Software Solutions Pty. Ltd.
Prospect Electricity
pTizan Computer Services Pty. Ltd.
Public Works Department, Information Services
Pulse Club Computers Pty. Ltd.
Qantek
QLD Department of Transport,
Information Tech. Serv. Branch
QLD Electricity Commission
Quality Bakers Pty. Ltd.
Quality By Design Pty. Ltd.
Redland Shire Council
Rehabilitation Tasmania
Renison Golfields Consolidated Ltd.
Rinbina Pty. Ltd.
Royal Melbourne Institute of Technology
SCEGGS Redlands Ltd
Sculptor 4GL+SQL
SEQEB Business Systems
Siemens Nixdorf Information Systems Pty. Ltd.

AUUG Institutional Members as at 12/08/1994

Smallworld Systems (Aust.) Pty. Ltd.
Smorgon ARC
Snowy Mountains Authority
SoftGen Pacific Pty. Ltd.
Software Plus (Australia) Pty. Ltd.
Softway Pty. Ltd.
South Australian Co-operative Bulk Handling
St. Catherine's School
St. Gregory's Armenian School
St. John God Hospital
St. Vincent's Private Hospital
Standards Australia
State Revenue Office
State Super (SSIMC)
Steelmark Eagle & Globe
Sterling Software
Storage Technology of Australia
Strategic Information Technologies Pty. Ltd.
Sunburst Regency Foods Pty. Ltd.
Sydney Electricity
Sydney Ports Authority
System Builder Development Pty. Ltd.
Systems and Management Pty Ltd
Systems Development Telecom Australia
TAB of Queensland
TAFE NSW, Information Systems Division
Tandem Computers
Tattersall Sweep Consultation
Technical Software Services
TechNIX Consulting Group International
Telecom (Applied Research & Development)
Telecom Australia
Telecom Australia Corporate Customer
Telecom Network Engineering Computer
Support Services
Telecom Payphone Services
The Far North QLD Electricity Board
The Fulcrum Consulting Group
The Knowledge Group Pty Ltd
The Preston Group
The Roads & Traffic Authority
The Southport School
The University of Western Australia
Thiess Contractors Pty. Ltd.
Thomas Cook Ltd.
TNT Australia Information Technology
Toshiba International Corporation Pty. Ltd.
Tower Technology Pty. Ltd.
Tradelink Plumbing Supplies Centres
Transport Accident Commission
Triad Software Pty. Ltd.
TurboSoft Pty. Ltd.
TUSC Computer Systems Pty. Ltd.
UCCQ
Unidata Australia
Uninet Consulting Pty. Ltd.
Unisys Australia Limited
University of Adelaide
University of Melbourne
University of New England, Dept. of Maths,
Stats & Computer Science
University of New South Wales
University of Queensland
University of South Australia
University of Sydney
University of Tasmania
University of Technology, Sydney
Unixpac Pty. Ltd.
Vanguard Computer Services Pty. Ltd.
Vanoco Pty. Ltd.
Vicomp
Victoria University of Technology
VME Systems Pty. Ltd.
Walter & Eliza Hall Institute
Wang Australia Pty. Ltd.
Water Board
Western Mining Corporation
Woodside Offshore Petroleum
Work Health Authority
Workstations Plus
XEDOC Software Development Pty. Ltd.
Zircon Systems Pty. Ltd.
Zurich Australian Insurance

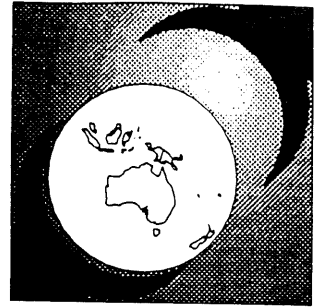
PPIT Pty. Ltd.

Computing Consultants ACN 059 051 320

Trading Office: 12 Moodie Street, Melton South, Vic., 3338.

Postal address: PO Box 879, Melton, Vic., 3337.

Ph: +61 3 747 9823, Fax: +61 3 747 8152



To: The members of AUUG

From: Paul Pavlinovich
Managing Director

Date: 21 July, 1994

Re: ACCSERV - Affordable Internet Access for individuals and small business

You may be aware of our new service AccServ - Affordable Internet Access. We are offering access to Internet e-mail, USENET News, and File Transfer.

We are pleased to announce a discount program for our fellow AUUG members of 10% off our annual subscription fees. This means that you can have commercial access to Internet e-mail and USENET News for as little as \$90 per annum (normally \$100).

For more details please contact us via one of the following methods:

Telephone (03) 747 9823

Facsimile (03) 747 8152

e-mail manager@accserv.ppit.com.au

Regards

Handwritten signature of Paul Pavlinovich

Paul Pavlinovich
Managing Director

Call for Articles for the Australian

The Australian newspaper runs an AUUG column every Tuesday, in its computer section. The aim of these articles is to inform the public and raise the profile of open systems within this country. Having one's views published in a respected national paper also carries kudos and recognition for authors.

AUUG would like to ensure that all members of the open system community have access to this voice, and we are actively seeking a diverse spectrum of people and opinions.

If you are interested in being part of this process, please provide me with the following information:

- * your name
- * contact details
- * a copy of your article

Your article should be between 600 and 800 words in length, and can address any issue that may be of interest within the open systems community. If you can't decide on an appropriate topic, please provide me with some professional details and I'll try and give you some ideas tailored to your expertise. Some typical subjects are listed at the end of this article.

If you have access to email, this is the preferred form of submission. Please format your article as a plain text file, with lines no longer than 79 characters, and with a blank line separating paragraphs. If you don't have email, please provide a hardcopy in a similar format (there's not much point doing any fancy typesetting).

All submissions are accepted on the understanding that they may or may not be used and that the material may be edited. AUUG will only submit your work to the Australian newspaper, although unless you advise us otherwise we will reserve the right to add your articles to a public FTP archive at some time in the future, and reprint them in AUUG's newsletter, AUUGN. The copyright on the material remains yours, your act of submitting material only gives us licence for the abovementioned purposes.

In practice, I submit your work to the Australian unedited and leave the decision of what to print up to them (I'm not in the business of being a thought police!). Usually a period of 2 to 4 weeks will then pass before you'll see your article in print; I maintain a pipeline of material to buffer me against the inevitable fluctuations in supply.

Please email or phone me if you have any further queries.

Lucy Chubb
lucyc@sw.oz.au
(02) 698 2322

NOTE: Lucy has taken over this role from Michael Paddon.

Some topical areas to get you started :-

- | | |
|--|---|
| Standards: POSIX, X/Open, System V. | Managing multiple network protocols. |
| The sudden demise of COSE;
just another consortium? | Living with the Internet. |
| The history of Unix. | Unix on PC's. |
| The future of Unix. | Linux - the people's Unix? |
| If NT is so popular,
how come no one is using it? | Would you run Unix at home? |
| Competition for the desktop:
Unix, Windows/NT and OS/2. | The graphics revolution. |
| Managing security. | Virtual reality. |
| Computer viruses. | CASE tools. |
| System administration. | Is Unix really that hard to use? |
| Network administration. | Now that Unix has grown up,
where have the hackers gone? |
| Networking technologies. | The costs of open systems. |
| Distributed computing. | Analyse a market trend. |
| What happened to OSI? | Run a straw poll on a topical subject. |
| | New technologies. |

New AUUGN Editor Wanted

After over three years of editing AUUGN, I find that due to work commitments I am forced to give up this rewarding task. As such AUUG is looking for a new editor.

The tasks include: the gathering of information from the AUUG Committee, Local Chapters, the Business Manager, the Secretariat and members, and liaising with the printer (currently in Sydney), advertisers and Australia Post.

The final product is a newsletter published bi-monthly of about 100 pages, which contains information both relevant to the membership and articles of a reasonable technical nature.

Most communication is via electronic mail, with a large number of the articles also being submitted this way.

You need to have an interest in newsletter publications, access to e_mail, knowledge of and access to various document processing packages on UNIX and PCs, and access to good quality printing facilities.

It is my intention to hand over to the new editor in early 1995, with a reasonable hand over time after that.

If you are interested in taking over the editing of AUUGN please contact me either via e_mail or phone me on (02) 717 3885.

Jagoda Crawford
auugn@munnari.oz.au

Softway
Excellence in System Software

Moving to OPEN Systems?

Softway is Australia's largest open systems software house.

We understand the needs of the open systems marketplace, and have extensive expertise in the following areas:

- Client/Server architectures
- TCP/IP based networks
- Security auditing
- Internet connections, firewalls
- Network integration
- Benchmarking and performance tuning
- Software Quality Assurance
- UNIX Training
- Contract software development

79 Myrtle Street
Chippendale
NSW 2008

Phone: (02) 698 2322
Fax: (02) 699 9174
Internet: enquiries@sw.oz.au

GURU's 1994 Open Systems Conference & Exhibition

ROSE '94

First Announcement and Call for Papers

The second Conference & Exhibition specifically for Open Systems in Romania, ROSE '94, will be held on November 3-4, 1994, in Bucharest. The Event organiser is GURU - Romanian UNIX User Group, a member group of EurOpen - The European Forum for Open Systems.

The Conference's aim is to promote the knowledge and use of the information techniques based on Open Systems, by favouring the sharing of experiences and information between specialists with similar interests and the direct contact between users and suppliers of Open Systems products.

Two tracks are intended at the Conference: a technical track, in which specialists from Romania and abroad are invited to present papers on all topics related to Open Systems technology, and a business-oriented track, in which leading suppliers of software and hardware will present their strategy towards the Open Systems market; their products will be on show at the Exhibition organised together with the Conference.

The Event seeks to consolidate the success of the first ever Conference & Exhibition specifically for Open Systems in Romania, ROSE featured speakers from 8 countries and over 20 sponsor companies.

The theme of the ROSE '94 Conference is:

"Open Systems, Technology for an Open World".

Submission of Papers

Topics for the Conference will cover the spectrum of recent research, development and experience using Open Systems technology. Papers are solicited on all aspects related to Open Systems, including:

- Architectures
- Operating systems
- Networking and Communications
- Software development tools and Applications
- Free software

Submissions should be in the form of extended abstracts or full papers (5-10 pages in length). The Conference language is English.

Please submit the papers, by post or e-mail, to Alexandru Rotaru, the programme chair, at:

GURU
P.O. BOX 63-42
Bucharest
Romania

or

arot@guru.ro

Accepted papers will be published in the Conference proceedings, which will be distributed to all the attendees. Presentations will usually be scheduled for 30 minutes.

Important Dates

September 3, 1994 Deadline for submission of extended abstracts or full papers

September 17, 1994 Notification of acceptance to authors

October 1, 1994 Deadline for receipt of the final papers

Attendance Fees

The Conference attendance fee for individual participants is \$ 80.

Lodging and travel fees are separate from fees for the Conference & Exhibition.

Payments will be made in hard currency or lei at the current exchange rate. Payments will be ordered to the GURU bank accounts:

- a. the amounts in lei in account no.: 4072996156281, BRD-SMB Romanian Bank for Development, Bucharest branch;
- b. the amounts in hard-currency in account no.: 1520796156281, BRD-SMB Romanian Bank for Development, Bucharest branch.

Please specify "PARTICIPATION FEE FOR ROSE '94" on the pay order.

For further information, including participation conditions for companies, please contact:

Alexandru Rotaru (GURU's chairman)
GURU
P.O. BOX 63-42
Bucharest
Romania
Phone: (+40-1) 633 05 17 (home)
Fax: (+40-1) 688 47 60
E-mail: arot@guru.ro

Adrian Ivanov (vice-chairman)
TC
109 Str. Republicii
3400 Cluj-Napoca
Romania
Phone: (+40-64) 11 60 60
Fax: (+40-64) 11 12 36
E-mail: aiva@cluj.iiruc.ro

Radu Togui (secretary)
GURU
P.O. BOX 63-42
Bucharest
Romania
Phone: (+40-1) 613 21 98
E-mail: radu@guru.ro

Daniel Dumitriu (board member)
P.O. BOX 61-33
Bucharest
Romania
E-mail: daniel@ddits10.guru.ro

Irina Athanasiu
P.O. Box 63-42
Bucharest
Romania

Phone: (+40-1) 621 59 63
Fax: (+40-1) 688 47 60
E-mail: irina@pub.ro or
irina@gfree.guru.ro

USENIX CONFERENCES

8TH USENIX SYSTEMS ADMINISTRATION CONFERENCE (LISA VIII)

Co-sponsored with SAGE, the System Administrators Guild

September 19-23, 1994

Town & Country Hotel, San Diego, California

Program Chair: Dinah McNutt, Zilker Internet Park, Inc.

Two-day tutorial program with twelve half-day tutorials. Three days of technical sessions include refereed papers, invited talks, panels, workshops. Vendor Display on Sept. 21 and Sept. 22.

Contact: USENIX, the UNIX and Advanced Computing Systems Technical and Professional Association, (714) 588-8649, fax: (714) 588-9706, e-mail: conference@usenix.org.

USENIX SYMPOSIUM ON VERY HIGH LEVEL LANGUAGES (VHLL)

October 26-28, 1994

El Dorado Hotel, Santa Fe, New Mexico

Program Chair: Tom Christiansen, Consultant

The three days of the Symposium will feature presentation of refereed papers, tutorial-style invited talks on related topics, hour-long overviews by invited speakers of some of the more popular VHLLs in use today, such as TCL, Perl, Icon, and REXX, and panel discussions. The Symposium will spotlight Very High Level Languages and their usefulness in leveraging certain kinds of tasks and introduce participants to concepts and approaches they haven't yet examined.

Contact: USENIX, the UNIX and Advanced Computing Systems Technical and Professional Association, (714) 588-8649, fax: (714) 588-9706, e-mail: conference@usenix.org.

USENIX SYMPOSIUM ON OPERATING SYSTEMS DESIGN AND IMPLEMENTATION (OSDI)

Co-sponsored by ACM SIGOPS and IEEE TCOS

November 14-18, 1994

Mariott Hotel, Monterey, California

Program Chair: Jay Lepreau, University of Utah

Submit one copy of an extended abstract by June 21, 1994, to the Program Chair via surface mail (preferred) to Jay Lepreau, Department of Computer Science, 3190 M.E.B., University of Utah, Salt Lake City, UT 84112 or e-mail (Postscript or ASCII) to osdi-papers@usenix.org.

Contact: USENIX, the UNIX and Advanced Computing Systems Technical and Professional Association, (714) 588-8649, fax: (714) 588-9706, e-mail: conference@usenix.org.

USENIX WINTER 1995 TECHNICAL CONFERENCE

January 16-20, 1995

Marriott Hotel, New Orleans, Louisiana

Program Chair: Peter Honeyman, CITI, University of Michigan

The only "broad-topic" USENIX Conference in 1995! Submit one copy of an extended abstract or full paper by July 18, 1994 via e-mail to winter95papers@usenix.org or surface mail to Winter 1995 USENIX, CITI, University of Michigan, 519 W. William, Ann Arbor, MI 48103-4943

Contact: USENIX, the UNIX and Advanced Computing Systems Technical and Professional Association, (714) 588-8649, fax: (714) 588-9706, e-mail: conference@usenix.org.

AUUG Regional Contacts 1994 - 1995

Location	Contact	Tel/Fax
Adelaide	Michael Wagner mhw@syserv.com.au	Tel: (08) 212-2800 Fax: (08) 231-0321
Brisbane	David Hughes bambi@Bond.edu.au	Tel: (075) 95-1450 Fax: (075) 95-1456
Canberra	John Barlow john.barlow@anu.edu.au	Tel: (06) 249-2930 Fax: (06) 249-0747
Darwin	Phil Maker pjm@cs.ntu.edu.au	Tel: (089) 466-666 Fax: (089) 270-612
Hobart	Steven Bittinger steven.bittinger@its.utas.edu.au	Tel: (002) 207-406 Fax: (002) 207-488
Melbourne	David Taylor davet@vaxc.cc.monash.edu.au	Tel: (03) 857-5660
Perth	Glenn Huxtable glenn@cs.uwa.edu.au	Tel: (09) 380-2878 Fax: (09) 380-1089
Sydney	Julian Dryden julian@dwt.csiro.au	Tel: (02) 809-9345 Fax: (02) 809-9476

AUUG QUEENSLAND CHAPTER UPDATE

QAUUG July Meeting Report

A moderately large contingent turned out at the Regatta Hotel on Tuesday July 26th for the monthly meeting of the Queensland Chapter of AUUG, affectionately known as QAUUG (that's "kworg", for all you pronunciation freaks ;-). Guest speaker for the evening was the distinguished Product Strategy Manager for Stallion Technologies, and QAUUG Executive Committee member Rick Stevenson, who gave an overview of both the history and state-of-the-art in LAN, MAN and WAN technologies. Rick has a long history in the UNIX community, and proved to be a knowledgeable and entertaining speaker.

Following Rick's presentation, Michi Henning spoke about the committee's efforts to assist members in gaining some form of internet access. Michi has been negotiating with a number of the commercial network service providers to gain appropriate services and discounts for QAUUG members, and a final decision is expected by the next meeting.

Also announced was the composition of the "new" QAUUG Executive Committee. Strangely enough, this looked a lot like the "old" QAUUG committee, but with the notable exception of Ian Turner, who had elected to stand down for a year or so. Ian's contributions to the establishment of the Queensland chapter are immeasurable, particularly his work in the organisation of the Summer Conference, which was a tremendous success.

The committee for 1994/95 is as follows:

Chairman: Tim Butterfield, C&T Computing
Treasurer: Michi Henning, CiTR <michi@citr.uq.oz.au>
Secretary: David Hughes, Bond University <bambi@bond.edu.au>
Committee: Sean Appleby, Sean Appleby Consulting
Rick Stevenson, Stallion Technologies <rick@stallion.oz.au>
Mark White, Pacific Star Communications <mwhite@pacstar.com.au>
Stuart Remphrey
Greg Birnie, Leeds Northrup <greg@lna.oz.au>
Paul Reithmuller, Sequent <par@sequent.com>
Susan Graham, MPA

QAUUG Meetings are held on the last Tuesday of each month, at the Regatta Hotel, Coronation Drive, Toowong. All Open Systems users, managers and developers are welcome to attend.

Mark White
Pacific Star Communications
<mwhite@pacstar.com.au>

AUUG Canberra Chapter

The following is the schedule for the monthly meetings of the AUUG Canberra chapter for the remainder of the year. All meetings are held at the Open Solutions Center in Barry Drive, TURNER ACT.

Time and Date: 7:30pm 13/9/94

Topic: Linux

Presenter: Linus Torvalds

Time and Date: 7:30pm 11/10/94

Annual General Meeting

Time and Date: 7:30pm 8/11/94

Topic: MacIntosh Application Environment

Time and Date: 5:30pm ??/12/94

Christmas drinks at ANU (details to be confirmed)

Peter Davie
Meeting Co-ordinator
AUUG Canberra Chapter
(06) 243 4835

AUUG NSW Chapter

1994 Meeting Schedule

September		++ No Meeting AUUG Winter Conference ++
October	11	Tuesday
November	8	Tuesday
December	13	Tuesday

Venue will be the Novotel Hotel Darling Harbour, 19:00 for 19:30.

We are (always) on the lookout for topics and speakers, if you have something to talk about, drop us a line. For more information feel free to contact:

Julian Dryden
julian@dwt.csiro.au
(02) 809 9345 (bh)
(02) 809 9476 (fax)

or

Brenda Parsons
bdp@sydney.dialix.oz.au
(018) 647 259
(02) 808 2797 (fax)

AUUG Inc. - Victorian Chapter

(formerly SESSPOOLE)

AUUG-Vic is the official Victorian chapter of AUUG Inc. It was the first Chapter of the AUUG to be formed, then known as SESSPOOLE, and its members have been involved in the staging of the Victorian AUUG Summer Technical Meetings every year since 1990. AUUG-Vic currently meets approximately every six weeks to hold alternate social and technical meetings. It is open to all members of AUUG Inc., and visitors who are interested in promoting further knowledge and understanding of UNIX and Open Systems within Victoria.

The purpose of the social meetings is to discuss UNIX and open systems, drinking wines and ales (or fruit juices if alcohol is not their thing), and generally relaxing and socialising over dinner. Whilst the technical meetings provide one or two "stand-up" talks relating to technical or commercial issues, or works in progress of open systems.

The programme committee invites interested parties wishing to present their work, to submit informal proposals, ideas, or suggestions on any topics relating to Open Systems. We are interested in talks from both the commercial and research communities.

Social meetings are currently held in the Bistro of the *Oakleigh Hotel, 1555 Dandenong Road, Oakleigh*, starting at about 6:30pm. Venues for the technical meetings are varied and are announced prior to the event. The dates for the next few meetings are:

Thu, 18 August '94	Technical
Tue, 27 September '94	Social
Wed, 9 November '94	Technical
Thu, 22 December '94	Social

Hope to see you there!

To find out more about AUUG-Vic and its activities, contact the committee or look for announcements in the newsgroup **aus.org.auug**, or on the mailing list **auugvic@clcs.com.au**. The committee may be reached more directly on **auugvic-exec@clcs.co.au**.

AUUG-Vic committee:

President:	Enno Davids	Metva	(03) 882 2333	enno@metva.technix.oz.au
Secretary:	David Taylor	Monash University	(03) 857 5660	davet@vaxc.cc.monash.edu.au
Treasurer:	Neil Murray	Webster Computer	(03) 561 9999	neil@wcc.oz.au
Programme Chair	Michael Paddon	Kodak	(03) 353 2382	mwp@munnari.oz.au
Committee:	Arnold Pears	La Trobe University	(03) 479 1144	pears@latcs1.lat.oz.au
Committee:	Peter Lazarus	Legent Australia	(03) 286 5200	plazarus@auspacific.legent.com

AUUG-Vic Email addresses:

General Membership	auugvic@clcs.com.au
Committee:	auugvic-exec@clcs.com.au
Mailing list administration:	auugvic-request@clcs.com.au

AUUG-Vic chapter activities

by **Enno Davids**

President AUUG-Vic

enno@metva.technix.oz.au

Well, another AUUGN rolls around and we find ourselves with little in the way of progress to report. Nevertheless, there is some value in recapping what has happened in the recently.

Meeting venue

As I mentioned in my last column, we are still looking for a 'better' venue to hold our social gatherings. This was reinforced by the turnout at the July meeting which could only be described as abysmal (no offence to those few who made the effort to turn up). Some of this could be due to the gremlins which struck down my electronic notification and some the brass monkey weather we get round this time of year, but there is still some reason to think that if the event took place somewhere more central, it would receive more support from everyone. So, once again, if anyone out there wants to drop a line with some suggestions, now would be a good time. As I said in the last column we're looking for somewhere with food and drink, reasonable rates and who can manage a quiet corner for from 5 to 30 people. In the meantime I'm pub crawling my way round the CBD. It's a tough job, but ...

Upcoming meetings

The AUUG-Vic announcement on the facing page (if things are printed in the usual manner) shows the intended schedule for our upcoming meetings. As I write this there are about two week to the technical meeting but the odds are it will have been and gone by the time you get to read this. The rest of the meetings look about right at this stage although we may move the December meeting slightly to accommodate the Christmas interregnum. Take a moment to pencil the dates into your diaries, tie pieces of string onto your lesser digits or whatever mnemonic devices you favor.

Internet access & Discounts

No progress has been made on either of these fronts since my last report. This will be dealt with soon, but for the moment most of the committee has had to beg off with their paying work and private lives taking precedence. In the meantime, we are investigating some of our options in both areas and

should have a plan sometime soon. On the discounts front, there may be some light at the end of the tunnel and we may be able to formalize things by the end of month.

Winter Conference

As you will be well aware by now the Winter Conference is upon us. I won't belabor the conference other than to pass my congratulations on to the program chairs who have put together another attractive group of presenters and topics.

In addition to this, AUUG-Vic is taking the opportunity presented by Linus Torvalds presence at the Winter Conference to attempt to put together a little get together in association with the LINUX Users of Victoria. So far we have the date and time arranged with our co-sponsors, the Friday at 7pm and we have lined up Linus. All that is left is to confirm a venue for the occasion. This is still coming together so if you have an interest in attending, keep an eye out for announcements closer to the big day. Michael Paddon is the man behind this and he too deserves our thanks.

Well, that's about all for the moment. Remember, if you have something to say or a question to ask you can drop us a line at any time either electronically or via OzPost or even just come along and corner one of the committee at one of the social or technical evenings or one of the other AUUG events.

See you there,

Enno.

From the Western Front

AUUG events in Perth have been rather eclipsed lately, at least for those of us in systems administration, by the SAGE-AU'94 conference. (SAGE-AU is the Systems Administrators Guild of Australia.) This was held in Perth on 11-13 July and was a great success, attracting around 100 delegates, about half of whom came from interstate.

If you're a systems administrator, as I know many AUUG members are, and you missed the conference, be sure to get along to next year's -- it is likely to improve the way you do your job. (But on the down side, this year's conference appears to have been a major new disease vector: it seems like half the systems administrators in the country, including myself, are now suffering from severe colds. If any more serious diseases were spread, I hope we never find out.)

WAUG's meetings continue to be well-attended -- about 40 of our 120 members regularly turn up. Mark Baker, our meeting organiser, keeps managing to find interesting speakers. In June it was great to see one of our own members, Steve Gunnell, present a comparison of the performance of a number of Unix boxes. In July, Jim Baker of Hewlett-Packard in the US generated some lively audience discussion with his Systems-Centred Engineering approach to managing a large workstation development project. See the reviews elsewhere in this AUUGN.

I'm going to mention the SAGE-AU conference again, because the closing talk would have been of interest to anyone involved in local AUUG chapters. Steve Simmons, from the USA, spoke on "Care and Feeding of a Local User Group". For many years Steve has run a local Sun and Unix user group, SemiSLUG, in his hometown of Ann Arbor, Michigan, and has given a lot of thought to what makes it keep going successfully although other local groups have fizzled. One factor, he said, is that SemiSLUG meets every month without fail, always at the same time and place. This makes it easy for people to remember. Another good idea that I think AUUG's local chapters might like to try is to welcome new members at the start of each meeting and get them to say a few words about themselves. (Actually in SemiSLUG it is called "Abuse of New Members", but that's just a humorous name -- there are no unpleasant initiations, or at least that's what Steve claimed.)

See you at AUUG'94.

Janet Jackson <jackson@cwr.uwa.edu.au> (WA Chapter Sub-editor), (09) 380 2408
From WAUG, the WA Chapter of AUUG

WAUG Meeting Reviews

June: Unix Kernel Variations

Steve Gunnell, Systems Administrator, Fremantle Port Authority

Steve's talk was one of the most technical WAUG has heard for some time: the results of his investigations into four different SVR4 kernel architectures during a machine purchasing decision.

The benchmarking procedure used was to write and run programs that exercised particular areas of the kernel (I/O, memory paging, and so on), measuring the machine's throughput against CPU load. We didn't get much of an opportunity to examine the programs themselves, as there were quite a few graphs of the results to get through.

I found the performance graphs a little long on data and a little short on information. However, Steve justified the effort that went into them by explaining that the purchasing department where he works stipulates a minimum weight for any purchase recommendation.

Unfortunately Steve couldn't explain the most interesting parts of the graphs: the anomalies (such as knees or spikes). His generic explanation was that various vendors make their own changes to the kernel to improve performance in what they consider "normal" conditions, and that the further conditions vary from normality, the more unpredictable the results (just like DOS! :-).

Steve wrapped up, the audience having only one question: what was the final purchasing decision? It turned out that in the end, when they benchmarked the application itself, its performance was proportional to the MIPS rating of the machine (not surprisingly, as it was a well-known machine-hogging database package that only a hardware vendor could love :-). However, the final purchasing decision was made at higher levels, and was not based on system performance anyway!

More than anything else, Steve's talk demonstrated that the only valid benchmark is running a real application mix, with real data and real users, on the machine configuration you might purchase. And that there are higher-level issues (such as vendor support and internal politics) that can override even these results.

Adrian Booth <abcc@dialix.oz.au>, Arena Technology, (09) 354 4936

July: Systems-Centred Engineering

Jim Baker, Workstation Systems Group Program Manager, Hewlett-Packard

I expected this to be an HP sales pitch, thinly veiled in technical-sounding language, but was pleasantly surprised.

Systems-Centred Engineering was Jim's term for a better approach to managing a large systems development group (in this case, the group developing the HP Snake workstation). The fundamental technique was to get everyone involved in the project (all the developers, managers, technical writers, and so on) to agree on one set of priorities, which would be based on the customer's expectations of the overall system. According to Jim the more usual approach is that the various groups (such as kernel developers or GUI engineers) define their own priorities based on incomplete information. Their views are narrow because management do not take the time to explain to them the big-picture view of the project, and the customer's perspective.

Jim emphasised two things that can be translated into anyone's job: firstly, mutual respect between everyone involved in the work; and secondly, getting everyone together to reach a consensus on priorities and objectives.

Jim's talk generated a lot of audience discussion -- always a sign of a good talk. It was generally agreed that HP's approach would not have worked without the support of the top management.

I also thought it worth taking into account that the customer's expectations are strongly shaped by the vendor's own marketing, and that of competitors.

Jim's talk got everyone thinking, and I believe most of the audience enjoyed it. I hope we have more of this style of talk in future.

Janet Jackson <jackson@cwr.uwa.edu.au>

Please Unsubscribe Me † from this List . . .

by Bryan McDonald
<bigmac@usenix.org>

I have been seriously working on the Internet for 6 years now, and it amazes me sometimes. When I first got heavily involved, I was emerging from the dungeons of the UC Davis MIS shop where I worked as a glorified gopher. I had this vision of thousands or millions of intelligent, learned people all plying their trade on the Net (this was well before someone in Washington coined the term Information Superhighway). I had certain expectations about these people: after all, this network was populated by some of the best minds in the world, gathering to create a new frontier. I guess that image set up expectations too high to meet and too pervasive to die, because I am always astounded and disheartened when another round of idiocy makes its way into my inbox.

For example, someone at a large vendor site recently got hold of a large list of email addresses, created a mailing list named foo-invite, and then sent a message to it, advertising a free, online information service. So, mildly annoying, not only because I already subscribed to said service, but also because I usually prefer to get my junk info in news groups or in my snail-mail. However, it is not as if they were trying to sell me something, it was a reasonable service, and I just removed the message.

No, this is not what prompted me to write this article. What followed was the culprit. Following this invitation, I began wading through a stream of abusive email all aimed at flaming the sender for throwing junk email at them. After all, they already got too much email and their time was too valuable to waste removing it. Of course, the flammers did not have the common courtesy to send it to the author of the

message, instead sending it back to the entire mailing list, which I am sure spanned hundreds, if not thousands, of people across the Internet.

Other examples abound. Shortly before the above incident I got a message on a mailing list covering a certain network device. The email, paraphrased, read, "I hate my [device] and I do not have time to learn majordomo, so please remove me from this list." Never mind the fact that to get on the list one has to use majordomo, an automated mailing list handling system, to subscribe. I especially liked this one, again paraphrased, "It would be nice if there were a listname for requests concerning this mailing list, and nicer still if there were an automated system to handle said requests. Please remove me from this list." The mailing list in question had been managed again by majordomo for at least a year.

These are just a slightly more obvious versions of what I see everyday on the 20 or so mailing lists I currently subscribe to. The fact that the lists I am on are mostly used by power-users and system administrators, supposedly the best of the Internet, makes it all the more depressing. The Internet is composed of so many people of so many different backgrounds and occupations, and the medium is so prone to create misunderstanding when used carelessly, that it is imperative that we who are the early explorers, the frontier blazers, get it right, and practice some common courtesy, common sense, and basic thoughtfulness as we ply our trades on this new frontier.

To: foo-invite-request
Cc: bigmac@erg.sri.com
Subject: Please unsubscribe me from this list

Thanks. . . and sorry for the rashness of others . . .

† This is a reprint from ;login, the USENIX Association Newsletter, Volume 19 Number 4

Crafting a Code of Ethical Conduct †

Kate Lance
Department of Computer Science
The University of Newcastle
Callaghan NSW 2308
clance@cs.newcastle.edu.au

1 Why Did SAGE-AU Want a Code of Ethics?

Most professional associations have some sort of code of ethics, conduct, or responsibility. The form of expression and the reasons for these codes are enormously varied. Some are short, “motherhood-and-apple-pie” statements of good intention, some are more promising of punishment than illustrative of good behaviour, some have strong opinions on their members’ moral activities even outside their professional roles, some contain reams of explanation and examples, and some are more like business documents. They aren’t, on the whole, meant to remain without revision for long periods.

For instance, you might imagine that the Australian Medical Association Code of Ethics is something like the single paragraph of high ideals of the Hippocratic Oath you would recall from old Hollywood movies: in fact it contains sixty items under twelve subheadings, with discussions of responsibility to patients, to clinical research trials, to other doctors; it talks about terms of contracts, advertising, organ transplants, and social obligations—and even contains a definition of brain death.

At the other extreme, the Journalists’ Code of Ethics is ten short points (many of which are blithely ignored, without penalty, by the less scrupulous media). The Institute of Electrical and Electronics Engineers get by with a page and a half of crisp statements, while the Data Processing Management Association has a code with two pages of professional guidelines and over *six* describing disciplinary procedures.

Before the Temporary Working Group on Ethics could define a code appropriate for SAGE-AU, we had to first decide just why we wanted a code, what we wanted to express with it, and whether or not it was really ethics we were considering.

1.1 Possible Reasons to Set Up a Professional Code

- to inspire members to be more ethical in their conduct
- to alert professionals to the moral aspects of their work that they might have overlooked
- to be a disciplinary code to enforce rules in order to protect professional standards
- to offer advice in cases of moral perplexity
- to alert employers and clients to what is proper conduct by a member of the profession
- to enhance the image of the professional in the public eye
- as one of the credentials of upgrading the status of a profession
- to protect the monopoly of the profession (which historically has been the major aim of most codes of ethics)

† This is reprinted from the proceedings of SAGE-AU’94

1.2 Definitions of Ethics

Oxford Dictionary:

1. moral philosophy
2. moral principles; rules of conduct
3. set of these

Webster's Dictionary:

1. the discipline dealing with what is good and bad and with moral duty and obligation
2. a set of moral principles or values
3. a theory or system of moral values
4. the principles of conduct governing an individual or group

1.3 Ethics and Codes

Ladd (1980) argues that in fact, professional codes of ethics have very little to do with the philosophical discipline of Ethics. He says:

- Ethics is an open-ended, critical intellectual activity, whose principles are established by exploring, deliberating and arguing about issues. It's *not* something that can be settled by fiat or authority—which is really law-making or policy-making. So ethical principles can't be established by an organisation: codifying ethics is like trying to codify medicine or architecture.
- Even if you could agree on some ethical principles, to *impose* them on others contradicts the notion of ethics itself, which presumes that people are autonomous moral agents. Ethics must be self-directed, not defined by others.
- Being a professional does not automatically make someone an expert in ethics, even in the ethics of one's own particular profession. There are no experts in ethics—everyone is capable of being “a teacher of virtue”.
- Professionals are not, just because they are professionals, *exempt* from the common obligations and duties binding on other people. These are “micro-ethical issues” regarding personal relationships between individuals, which simply involve the application of ordinary notions of decency, civility, humanity, respect and responsibility.
- While ethics *can* be used to criticise or evaluate a code of conduct or a professional code, it's not the same thing, it's the process you use to validate the code, not the end product.

1.4 What's Special About System Administration?

Rob Kolstad: “*System administration, as a field, is unique. I can think of no other field that shares even a majority of its qualities (and I've asked the question of dozens of other technical and non-technical types). The field is incredibly broad and deals with systems in timeframes from milliseconds through months. It deals with components whose size is measured in bits through components whose aggregate is measured in gigabytes (or even terabytes). It deals with cold, calculating machines and warm, human people. It sometimes deals with life and death; it deals with the background color of the someone's screen. It is a discipline which, when performed best, is virtually unobservable.*”

- System administration has been a profession for much less time than traditional ones like medicine and law—it has fundamental differences from any profession that has ever before existed in human history.
- It involves the management of resources whose utility, power and importance in human affairs is increasing without apparent limit.
- Those resources in themselves demand entirely new ways of dealing socially and legally with issues as wide-ranging as privacy, security, and intellectual property.
- The profession has so recently emerged that most people have very little understanding of its demands, its powers, and even its potential abuses.
- The essential relationship in most professions is between clients and practitioners, such as patients and doctors. But the relationship between users and system administrators involves a third party, the system itself—there simply is no way for service to be provided unless a viable system exists.
- Because of this, the very nature of the work system administrators do demands that they aspire to certain qualities, such as:
 - a commitment to technical integrity—because systems are unforgiving of incompleteness or negligence.
 - a commitment to cooperation and communication—attitudes that are fundamental to the existence and viability of network resources.
 - a recognition of the responsibility owed to the people who trust computer systems to manage their years of research data, their medical records, their tax returns, their love letters, and sometimes their very lives.

1.5 The Reasons We Finally Agreed On

After much discussion based on the above points we agreed upon the following items as the rationale for our code:

- To indicate to employers that we are a professional body that takes a serious view of our responsibilities.
- While not being a job description, to delineate the extent of our powers and responsibilities for people unfamiliar with the scope of our activities.
- To indicate to users, colleagues and employers that we will act in good faith, and, as much as possible, in their best interests.
- To protect ourselves should unethical behaviour be demanded of us.
- To explore what ethics might have to offer *us*, to find out for ourselves where our ethical boundaries and obligations really lie.

2 How Other Groups Have Done It

2.1 Privilege, Confidentiality and Privacy

One of the first things we looked at was the protection given to various professions which deal with privileged, confidential information, to see how that might relate to our profession.

- **Lawyers** are protected under Common Law, in civil and criminal cases, from having to reveal lawyer-client communications to do with advice or existing or anticipated litigation.
- **Journalists** have no Common Law or statute protection in Australia (although in England there is statutory privilege protecting them from having to reveal their sources of information).
- **Clergy** have no Common Law privilege, but there are statutes in some states protecting them from having to reveal the contents of a confession. The relevant professional bodies do not permit disclosure: Catholic Canon law imposes immediate excommunication from the Church, and the penalties imposed by other churches for breaking confessional privacy range from excommunication to disciplinary action.
- **Doctors** have no protection under Common Law, but there are statutes in some states protecting them in civil proceedings from revealing confidential information. Doctors revealing confidential information may be censured or even deregistered by their professional association. The Australian Medical Association permits disclosure only in cases when:
 - the patient gives consent
 - it's undesirable to seek consent on medical grounds
 - the doctor has an overriding duty to society
 - for certain medical research
 - it's required by the legal profession

2.2 Privacy Act (1988) Information Privacy Principles

The Federal government requires all of its departments and agencies to comply with the Information Privacy Principles. They govern the keeping of personal information records: methods of collection, storage and security, access, accuracy, completeness, usage, and disclosure to other people.

A copy of the Information Privacy Principles is available by anonymous ftp from [ftp.mel.dit.csiro.au](ftp://ftp.mel.dit.csiro.au/pub/SAGE-AU/Ethics/IPP) as **pub/SAGE-AU/Ethics/IPP** and all members of SAGE-AU are urged to read them. (The OECD Guidelines to Computer Privacy and Security are also available there.) Even if you are not associated with any Federal government activity, similar principles are soon to be introduced at some State levels, and may eventually apply much more widely. Extracts from several of the IPPs with system administration relevance are listed below:

Principle 4: storage and security of personal information

A record-keeper who has possession or control of a record that contains personal information shall ensure: that the record is protected, by such security safeguards as it is reasonable in the circumstances to take, against loss, against unauthorised access, use, modification or disclosure, and against other misuse.

Principle 9: personal information to be used only for relevant purposes

A record-keeper who has possession or control of a record that contains personal information shall not use the information except for a purpose to which the information is relevant.

Principle 11: limits on disclosure of personal information

A record-keeper who has possession or control of a record that contains personal information shall not disclose the information to a person, body or agency (other than the individual concerned) unless:

- the individual concerned is reasonably likely to have been aware, or made aware under Principle 2, that information of that kind is usually passed to that person, body or agency;

- the individual concerned has consented to the disclosure;
- the record-keeper believes on reasonable grounds that the disclosure is necessary to prevent or lessen a serious and imminent threat to the life and health of the individual concerned or of another person;
- the disclosure is required or authorised by or under law; or
- the disclosure is reasonably necessary for the enforcement of the criminal law or of a law imposing a pecuniary penalty, or for the protection of the public revenue.

2.3 Codes from Other Computing Organisations

Here are some examples of codes from other computing groups. While the following one is short and readable, it doesn't address the specific concerns of *computing* professionals, but only professionals in general.

The Australian Computer Society Code of Ethics

I must act with professional responsibility and integrity in my dealings with clients, employers, employees, students, and the community generally. By this I mean:

PRIORITIES: I must service the Interests of my clients and employers, my employees and students, and the community generally, as matters of no less priority than the interests of myself and my colleagues.

COMPETENCE: I must work competently and diligently for my clients and employers.

HONESTY: I must be honest In my representation of skills, knowledge, services and products.

SOCIAL IMPLICATIONS: I must strive to enhance the quality of life of those affected by my work.

PROFESSIONAL DEVELOPMENT: I must enhance my own professional knowledge and skills and those of my colleagues, employees and students.

COMPUTING PROFESSION: I must enhance the Integrity of the Computer Profession and the respect of its members for each other.

The following code had almost a half a page of explanation and examples for each item. While it is certainly comprehensive, it's also repetitive and too long, and probably would never be read through to the end by many members. (These are just the item headings!)

The Association for Computing Machinery—Code of Professional Conduct

1.1 Contribute to society and human well-being.

1.2 Avoid harm to others.

1.3 Be honest and trustworthy.

1.4 Be fair and take action not to discriminate.

1.5 Honor property rights including copyrights and patents.

1.6 Give proper credit for intellectual property.

1.7 Respect the privacy of others.

1.8 Honor confidentiality.

2.1 Strive to achieve the highest quality in both the process and products of professional work.

2.2 Acquire and maintain professional competence.

2.3 Know and respect existing laws pertaining to professional work.

2.4 Accept and provide appropriate professional review.

2.5 Give comprehensive and thorough evaluations of computer systems and their impacts, including analysis of possible risks.

- 2.6 Honor contracts, agreements, and assigned responsibilities.
- 2.7 Improve public understanding of computing and its consequences.
- 2.8 Access computing and communication resources only when authorized to do so.
- 3.1 Articulate social responsibilities of members of an organizational unit and encourage full acceptance of those responsibilities.
- 3.2 Manage personnel and resources to design and build information systems that enhance the quality, effectiveness and dignity of working life.
- 3.3 Acknowledge and support proper and authorized uses of an organization's computing and communication resources.
- 3.4 Ensure that users and those who will be affected by a computing system have their needs clearly articulated during the assessment and design of requirements; later the system must be validated to meet requirements.
- 3.5 Articulate and support policies that protect the dignity of users and other affected by a computing system.
- 3.6 Create opportunities for members of the organization to learn the principles and limitations of computer systems.

Some Points Covered by Other Codes

As well as the two codes above, we also looked at the codes for the Institute of Electrical and Electronics Engineers, the British Computer Society, the Data Processing Management Association, and the Institute for Certification of Computer Professionals. Most of the points in the above ACM list appeared in the other lists, but they were generally not so comprehensive. Some interesting additions to the above list:

ICCP: ...one is expected to apply the same high standards of behaviour in one's personal life as are demanded in one's professional activities.

IEEE: Advance the integrity and prestige of the profession by practicing in a dignified manner and for adequate compensation.

BCS: Actively seek opportunities for increasing efficiency and effectiveness to the benefit of the user and of the ultimate recipient.

DPMA: Use my skill and knowledge to inform the public in all areas of my expertise.

3 Diversion: An Ideal System

As we were trying to identify the essential points from other codes, taking account of additional issues that seemed important, and trying to apply them to as wide a variety of system work as possible, it became clear that we just didn't know enough about the different jobs that SAGE-AU members were doing—what range of activities they actually deal with, whether or not they had the freedom or the opportunity to live up to the items we were considering, and, for comparison, just what they would consider the ideal system setup.

So a questionnaire went out to the Ethics group, asking people to rate the levels of importance, low to high, allocated in their organisation to aspects of system work, such as backups, applying patches, access to the Internet, conditions for users, etc. We received ratings on 30 past and present jobs, and 10 descriptions of the Ideal System.

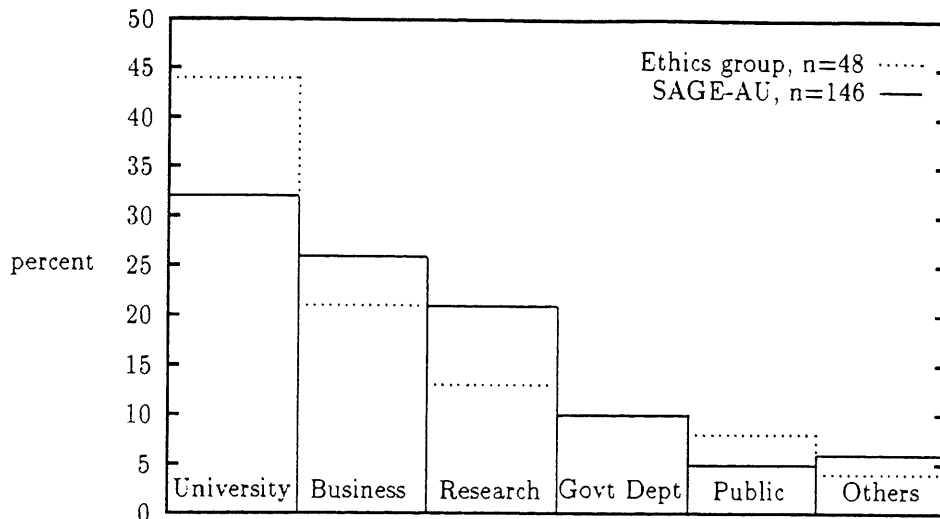


Figure 1: Distribution of organisations in SAGE-AU and the Ethics group

3.1 Did the Ethics Working Group Represent the SAGE-AU Members?

From people's email addresses it was fairly easy to see what type of organisation they posted from— university, business, government research, government department, public access account, and a few which couldn't be identified. I compared the distribution of those in the whole SAGE-AU mailing list with those on the Ethics list (Figure 1).

It was clear that on the Ethics list there were a higher proportion of university sysadmins, and a lower proportion of government research organisation sysadmins, than in SAGE-AU overall. Maybe university sysadmins have to deal more often with ethical dilemmas (from student activity?) than do research organisations.

3.2 Real Systems vs an Ideal One

The questionnaire asked people to state the level of importance the following categories of system work were given in their organisation, from 1 (minimum importance) to 5 (maximum importance); and, given an ideal system, what level they themselves would rank these categories:

- 1) Frequent backups
- 2) Applying patches
- 3) Performance monitoring
- 4) Security monitoring
- 5) Disaster recovery planning
- 6) Privacy of email
- 7) Privacy of accounts
- 8) Access to internet
- 9) Safe user environment
- 10) Comfortable user environment
- 11) Self-education
- 12) User-education
- 13) Updating hardware
- 14) supply of consumables

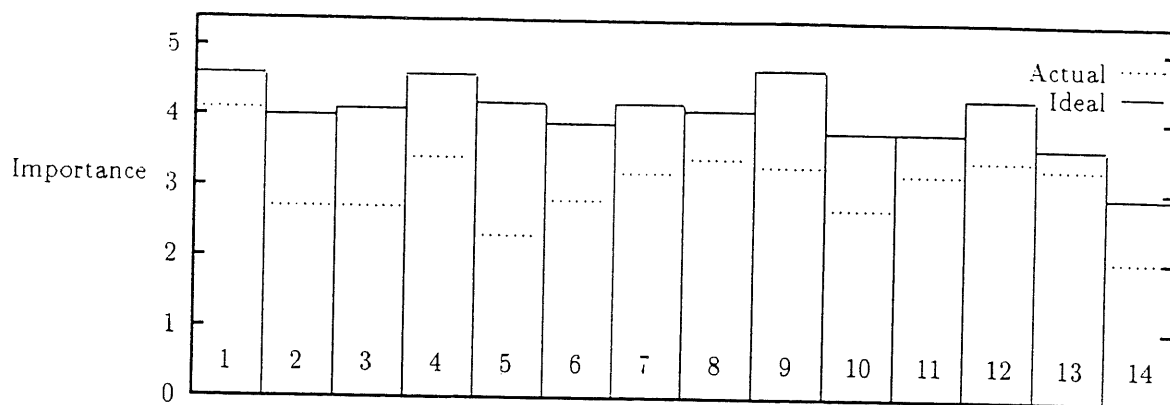


Figure 2: Actual vs ideal systems

The numbers against the categories are the ones plotted on Figure 2, which compares the mean of the replies from different organisations to the mean of the ideal system replies.

The mean of the replies from all the organisations was used, as the total numbers per organisation were too low for separate reliability. However, there were suggestions in the separate results that university sysadmins were more concerned than others about applying patches and security monitoring; while research organisations were less concerned with performance monitoring, but rated email and account privacy very highly.

Both universities and research sites gave access to the Internet an importance more than twice that currently given it by businesses and government departments—but the system administrators of those businesses and government departments gave it just as much importance as the others in their ideal system.

All groups were in agreement that backups are important as an ideal, *and* they get them done as well; however, everyone also think disaster recovery planning is a great idea—but it looks like it's not getting very high priority in real life.

4 What We Ended Up With, And Why

The preamble tries to summarise as many of the points discussed above as possible. The overall aim was to express, to others as well as to ourselves, recognition of our ethical duties as professionals.

When writing the items we found that lists of “job descriptions” were a difficulty: if there was too little detail, the code ran the risk of blandness and irrelevance to real-life situations; but too much detail might make it date too quickly, and be too specific to particular areas of system administration. Detailed lists also run the risk of appearing to be exhaustive, when they may only have been offered as examples.

Another difficulty was making it general enough to apply not just to administrators of Unix systems or those with access to the Net, but to apply to all kinds of installations, with their vast array of different system duties. The problem also arose concerning just what the people involved at all levels with systems should be called—the list of users, clients, employers, employees, colleagues, peers, subordinates, and other administrators, was eventually collapsed just to “users”.

“Informing” users of things they need to know was also a discussed point. At one stage we were just

going to strive to do it, but finally, it seemed better to affirm that we would indeed *inform*, we would make information available, then it was up to others to utilise it.

Similarly for the items that relate to our own continuing education, both technical and social—we affirm our duty to actually do it, not just wish to do it, because in this rapidly-changing field we need to emphasise that further education is essential, it's not really a luxury or a choice.

SAGE-AU Code of Ethical Conduct

In a very short period of time computers have become fundamental to the organisation of societies world-wide; they are now entrenched at every level of human communication from government to the most personal. Computer systems today are not simply constructions of hardware—rather, they are generated out of an intricate interrelationship between administrators, users, employers, other network sites, and the providers of software, hardware, and national and international communication networks.

The demands upon the people who administer these complex systems are wide-ranging. As members of that community of computer managers, and of the System Administrators' Guild of Australia (SAGE-AU), we have compiled a set of principles to clarify some of the ethical obligations and responsibilities undertaken by practitioners of this newly emergent profession.

We intend that this code will emphasise, both to others and to ourselves, that we are professionals who are resolved to uphold our ethical ideals and obligations. We are committed to maintaining the confidentiality and integrity of the computer systems we manage, for the benefit of all of those involved with them.

No single set of rules could apply to the enormous variety of situations and responsibilities that exist: while system administrators must always be guided by their own professional judgement, we hope that consideration of this code will help when difficulties arise.

(In this document, the term "users" refers to all people with authorised access to a computer system, including those such as employers, clients, and system staff.)

As a member of SAGE-AU I will be guided by the following principles:

1. Fair Treatment

I will treat everyone fairly. I will not discriminate against anyone on grounds such as age, disability, gender, sexual preference, religion, race, or national origin.

This was an item in the ACM code which we thought was important. There was some argument about whether or not the list was essential: but it was decided it might be helpful given the diversity of people in places like universities and research organisations.

2. Privacy

I will access private information on computer systems only when it is necessary in the course of my duties. I will maintain the confidentiality of any information to which I may have access. I acknowledge statutory laws governing data privacy such as the Commonwealth Information Privacy Principles.

At one stage we had "information that belongs to others" rather than "private information", which brought up problems with defining who actually owns the files on a computer—the owner as defined by the operating system, or the owner of the hardware? We also tried to avoid emotive terms like "infringing privacy" or "system privileges".

3. Communication

I will keep users informed about computing matters that may affect them—such as conditions of acceptable use, sharing of common resources, maintenance of security, occurrence of system monitoring, limitations of electronic media, and any relevant legal obligations.

Issues here were whether or not you can get people to actually take in the information that you present to them, whether or not most sites actually had defined Conditions of Use, and how much job description was necessary.

4. System Integrity

I will strive to ensure the integrity of the systems for which I have responsibility, using all appropriate means—such as regularly maintaining software and hardware; analysing levels of system performance and activity; and, as far as possible, preventing unauthorised use or access.

Here we argued about mentioning anything to do with current technology (such as disk backups) and again, how much detail should go into this without it becoming a job description.

5. Cooperation

I will cooperate with and support my fellow computing professionals. I acknowledge the community responsibility that is fundamental to the integrity of local, national, and international network resources.

This was one of the hardest to define, because it refers to Net access, which some sites don't have; but in this form, even if people aren't on the Net, they can still acknowledge its importance. An early draft mentioned keeping in touch with "organisations that coordinate security efforts on behalf of network users", but these organisations are also examples of community cooperation.

6. Honesty

I will be honest about my competence and will seek help when necessary. When my professional advice is sought, I will be impartial. I will avoid conflicts of interest; if they do arise I will declare them.

This was the least revised item!

7. Education

I will continue to update and enhance my technical knowledge and management skills by training, study, and the sharing of information and experiences with my fellow professionals.

This had "attend professional conferences" in one draft, but sadly, we decided to be less specific.

8. Social Responsibility

I will continue to enlarge my understanding of the social and legal issues that arise in computing environments, and I will communicate that understanding to others when appropriate. I will strive to ensure that policies and laws about computer systems are consistent with my ethical principles.

Initially there was a list here of issues such as "privacy, confidentiality, academic freedom, copyright, intellectual property, illegal access and computer crime", but again, in the interests of greater applicability, it was put more generally.

9. Workplace Quality

I will strive to achieve and maintain a safe, healthy, productive workplace for all users.

Too-specific lists again: "fundamental health and safety procedures, appropriate ergonomic furniture, and adequate space and lighting" was discarded for greater generality.

5 Future Work

The Ethics working group will convene again in a year's time to consider any revisions to this code that may then be necessary.

One additional point of discussion was, just what do we do if someone in SAGE-AU *doesn't* respect this code when doing system administration. Some other codes had substantial sections on disciplinary procedures. We decided that any such activity would have to be looked at by a different working group, one with more legal expertise than this one, to consider what mechanisms should be in place:

- to give support to sysadmins who might feel their personal ethics are being compromised
- to deal with complaints if others think that a sysadmin is compromising the code of ethical conduct
- to discuss, defend, decide about the complaint
- to enforce a judgement if the complaint is upheld

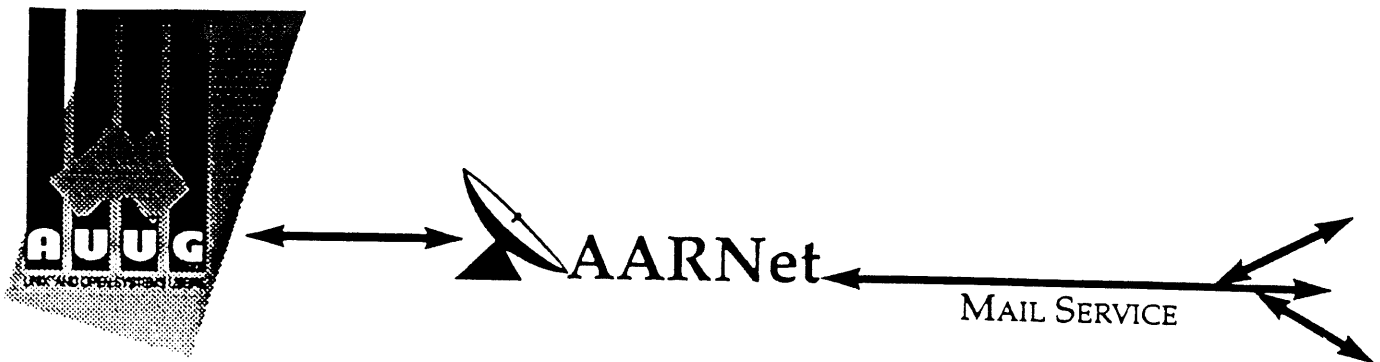
6 Acknowledgements

Almost half of all the people on the Ethics group mailing-list took some part in the discussion over the last six months, but I am particularly grateful for the substantial contributions of Janet Jackson, Chris Deeble, Glenn Huxtable, Douglas Ray, Shane Youl and Hal Miller.

Thanks also to John McPhee, of the Faculty of Law, The University of Newcastle, who assisted us with informal opinions and the on-line copies of the Information Privacy Principles and the OECD Guidelines on Privacy and on Security, available by anonymous ftp from [ftp.mel.dit.csiro.au:pub/SAGE-AU/Ethics](ftp://mel.dit.csiro.au/pub/SAGE-AU/Ethics).

7 References

Ladd, J., 1980, *The Quest for a Code of Professional Ethics: an Intellectual and Moral Confusion*, in *Ethical Issues in the Use of Computers*, Johnson, D. G., and Snapper, J. W., 1985, Wadsworth Publishing Company, Belmont, California.



Dear Site Administrator,

As you may be aware, the arrangements for mailing to addresses outside Australia (and also to AARNet sites) changed in May 1991. Since then, the University of Melbourne are no longer managing the administrative details associated with maintaining this service. The AARNet (Australian Academic and Research Network) management has taken over administering the service, and are requiring all ACSnet and similar sites to register with AARNet and pay a fee for continued access to Internet mail services. AARNet have set this fee as \$1000 per annum for most sites, with larger sites paying more (you know who you are).

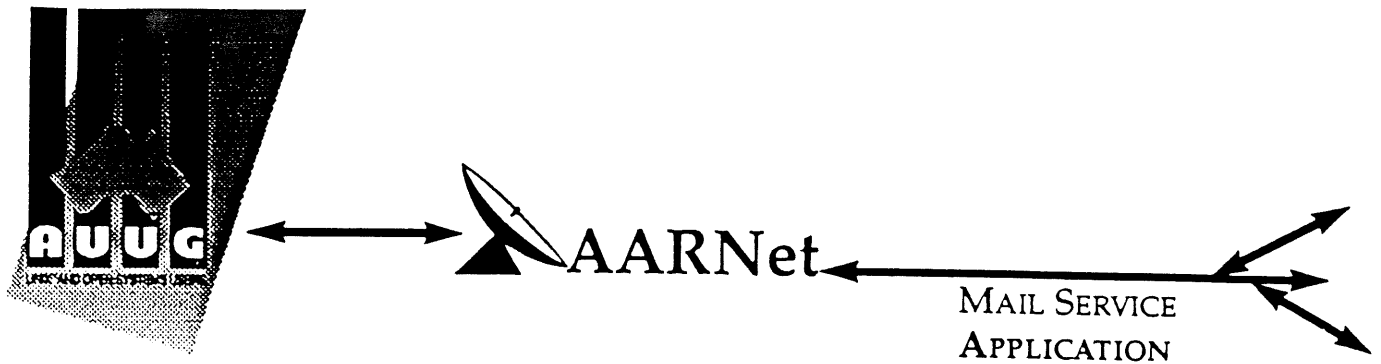
The fee is intended to cover use of AARNet bandwidth for your network traffic. Registration with AARNet, however, provides ONLY the registration of your address in worldwide address tables - your site will be unreachable without this registration. The fee does NOT cover the costs involved in obtaining a connection to AARNet or ACSnet NOR does it include a guarantee that you can be connected or even to help you find a connection point. See Note B for some information about connection services.

AUUG as a service to its members has negotiated with AARNet to achieve a lower price for this basic address registration service. The lower price is based on the reduction in paperwork for the AARNet management authorities. The AUUG/AARNet fee is dependent on the membership status of the owner of the machine(s)/domain involved, and is currently \$250 for members and \$600 for non-members. As such it is a substantial discount on the AARNet fee, but only applies to sites in the AARNet \$1000 category. Larger sites will need to negotiate directly with AARNet.

The address registration is for one AUUG membership year. Membership years start on the 1st January or July, whichever is nearest to receipt of your application. Sites which do not renew their AUUG/AARNet registration annually with their AUUG membership each year will be removed from the Internet tables and will no longer be able to communicate with international and AARNet hosts. Reminders/invoices will be sent along with your membership renewal.

The required initial registration form is attached below. It should be completed and forwarded to AUUG's (postal) mailing address at the bottom of the form or faxed to (02) 332 4066. If you have any queries on the AUUG/AARNet arrangements please direct them to Catrina Dwyer at the AUUG office on (02) 959 3656 (catrina@swift.sw.oz.au) or myself (frank@atom.ansto.gov.au).

Regards,
Frank Crawford
AUUG-AARNET Administrator
AUUG Inc.



On behalf of the organisation listed below I wish to apply to be a Mail Service Affiliate Member of AARNet, and accordingly request that AUUG Incorporated arrange for the Australian Vice-Chancellors' Committee (AVCC) to maintain on my behalf an electronic mail delivery record in the Australian Academic and Research Network (AARNet) to allow my organisation to send and receive electronic mail carried across AARNet.

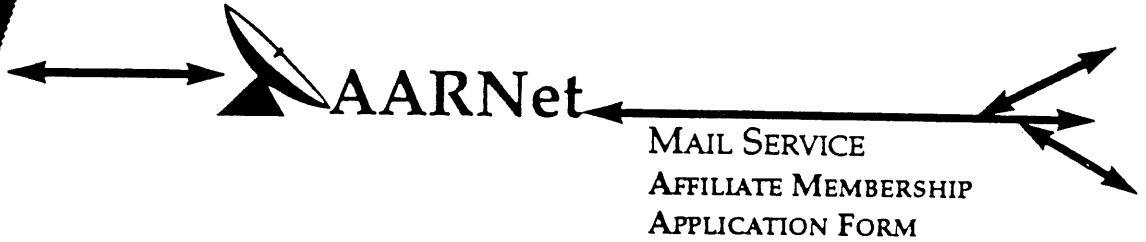
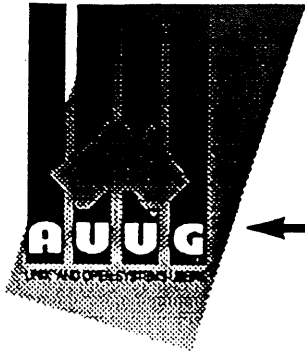
I understand that the AVCC may consult the recorded logs of my organisation's usage of AARNet facilities for 1990, and determine that I am ineligible for registration under the terms of the agreement between AVCC and AUUG Inc. I understand that AUUG Inc will invoice my organisation for this service for the calendar year 1991 and for subsequent years unless it receives my organisation's written advice to terminate the Affiliate Membership of AARNet.

I understand that the AVCC and AUUG Inc maintain the right to vary the Mail Service Affiliate Membership charges from year to year, and maintains the right to cease offering this service to my organisation at the start of any year, at their discretion. I understand that in the event of any variation of the Mail Service Affiliate Membership of AARNet, my organisation will be advised in writing by the AVCC or AUUG Inc to the address below.

I understand that in consideration of the AARNet Mail Service Affiliate Membership charge, AARNet will undertake to maintain a mail directory entry which will direct incoming electronic mail to the AARNet gateway system(s) which I have nominated below. Furthermore I accept that there is no other undertaking made by AARNet in terms of reliability of mail delivery or any other form of undertaking by AARNet or the AVCC in consideration of the payment to AARNet for the maintenance of the mail directory entry on AARNet.

I undertake that my organisation's use of the mail delivery services over AARNet will not be used as a common commercial carrier service between my organisation and other organisations receiving similar services from AARNet, nor will it be used as a commercial carrier service between branches of my organisation. Furthermore my organisation undertakes to use AARNet facilities within the terms and conditions stated in the AARNet Acceptable Use Policy. I accept the right of the AVCC or AUUG Inc to immediately terminate this service at their discretion if these undertakings are abused by my organisation (where the AVCC retains the right to determine what constitutes such abuse).

I understand that a fee is payable with this application: of \$250 if the host/hosts covered are owned by a member of AUUG Incorporated, or \$600 if the host/hosts covered are not owned by an AUUG member. Corporation host owners may only claim the member price if the corporation is an Institutional member of AUUG Inc. My cheque payment of either \$250 or \$600 as appropriate is enclosed with this application.



PLEASE PRINT CLEARLY!

Date: _____

Name of Organisation/Owner: _____

Signed: _____ AUUG Membership No (if known): _____

Name: _____ Position: _____

on behalf of the organisation named above.

Address: _____

Postcode: _____

Administrative Contact: _____ Title: _____

E-Mail: _____ Phone: () _____

Fax: () _____

Technical Contact: _____ Title: _____

E-Mail: _____ Phone: () _____

Fax: () _____

Mail Delivery Information to be entered in AARNet (see Note A next page)

Domain Names Requested: _____

Gateway Addresses: _____

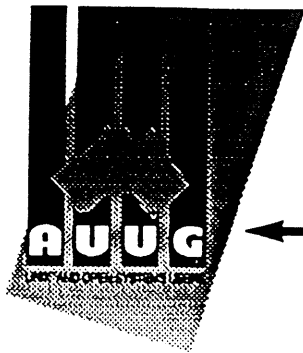
Expected Link Protocol: UUCP SL/IP MHSnet Other: _____

* * * *

Send this page only to:

AUUG Incorporated
PO Box 366
Kensington NSW 2033

Phone: +61 2 361 5994
Fax: +61 2 332 4066



AARNet

MAIL SERVICE
AFFILIATE MEMBERSHIP
APPLICATION FORM cont'd.

Note A. Mail Delivery Information

Two items of information are required: firstly the preferred name of your mail host (or the domain name(s) of a group of hosts) in Internet domain name system format, and secondly the name (or names) or AARNet gateway systems who will accept electronic mail over AARNet (and connected overseas networks) on your behalf and forward it to you. The primary requirement for an AARNet gateway is its ability to recognise your host/domain addresses and perform the necessary mail header rewriting reliably.

Please check with the postmaster at your preferred AARNet gateway host site before citing them as a gateway for AARNet mail delivery. For ACSnet addresses (*.oz.au), the host "munnari.oz.au" (Melbourne University) is a recommended gateway. Other possible sites include "metro.ucc.su.oz.au" (Sydney University), sirius.uca.adelaide.edu.au (University of Adelaide), uniwa.uwa.oz.au (University of WA) and bunyip.cc.uq.oz.au (University of Qld). Note that all gateway addresses must be fully domain qualified.

Example Mail Directory Information request:

Mail addresses required:	acme.oz.au, *.acme.oz.au
Mail Gateways (primary)	gw.somewhere.edu.au
(secondary)	munnari.oz.au
(secondary)	unnet.uu.net

The addressability of your site and the willingness of your nominated gateways to act in that capacity will be determined before registration proceeds. Processing will be made faster if you contact the postmaster at your nominated gateways in advance to inform them of your intentions. Your nominated technical contact will be notified by email when registration is complete.

Note B. Getting Connected

New sites will need to find an existing AARNet or ACSnet site who will accept their site as a connection, and also select a protocol for transferring data over their mutual link. Although the UUCP package is a standard inclusion with UNIX, it is little used in Australia due to its relatively poor performance. Other possible choices for your link protocol include SLIP (TCP/IP) and MHSnet.

Among a number of organisations who provide connection services, Message Handling Systems Pty Ltd have announced a special offer on both their link software and connect time for AUUG members. For more details on this offer, contact Message Handling Systems on (02) 550 4448 or elaine.mhs.oz.au.

Open System Publications

As a service to members, AUUG will source Open System Publications from around the world. This includes various proceeding and other publications from such organisations as

AUUG, UniForum, USENIX, EurOpen, Sinix, *etc.*

For example:

EurOpen Proceedings		USENIX Proceedings	
Dublin	Autumn'83	C++ Conference	Apr'91
Munich	Spring'90	UNIX and Supercomputers Workshop	Sept'88
Trosno	Spring'90	Graphics Workshop IV	Oct'87

AUUG will provide these publications at cost (including freight), but with no handling charge. Delivery times will depend on method of freight which is at the discretion of AUUG and will be based on both freight times and cost.

To take advantage of this offer send, in writing, to the AUUG Secretariat, a list of the publications, making sure that you specify the organisation, an indication of the priority and the delivery address as well as the billing address (if different).

AUUG Inc.
Open System Publication Order
PO Box 366
Kensington, NSW, 2033
AUSTRALIA
(02) 332 4066

Fax:

Review of the AIR 2.0 TCP Software PACKages for Windows

Chris Maltby
chris@softway.sw.oz.au

The first thing unusual about the Air Series of Windows TCP software is that Spry don't actually have a TCP/IP stack of their own. They have bundled a copy of both the Novell and also the Microsoft stacks as commodities, and then conform to the Winsock interface supported by both of those stacks.

The strength of this approach is that you are more likely to be able to support the Air packages alongside your existing PC LAN software, be it ODI or NDIS based. Given that Winsock is now a de-facto standard, this is a good approach; and you can expect to be able to run the Air software on top of any transport that offers Winsock.

I tested both the ODI and the NDIS stacks for myself and discovered why Spry recommend the Novell ODI for preference! The Microsoft stack gave a lot more trouble on my machine (a 33Mhz 386 with 8Mb running Windows 3.1). And since we didn't have a Windows for Workgroups environment to conform to, I was happy to leave the NDIS stuff alone. In any case, this seems to be a Microsoft problem, not Spry's.

When I began testing the software, and when I explained what I was doing to Softway's inquisitive technical staff, they asked me "What does this offer over the PD and shareware TCP/IP utilities?". This is a fair question which has at least two answers.

First, it's an integrated package of utilities which share a reasonable similar look and feel. Second, Spry claim to have put a lot of effort into optimising the interface with the Windoze primitives, based on their internal connections at Redmond, Washington. As well, unlike the freeware, you get a pretty good set of manuals pitched at the first time user and someone to help you get it going.

On the first point, it's fair to say that they have succeeded reasonably well. I had some quibbles about the way that some of the functions were presented, but maybe they were aiming them at people used to the way that Microsoft do things.

The big missing item is a simple FTP client. They have an FTP based file manager tool, which (among other things) knows about the stuff returned by a DIR command on different server systems. Unfortunately, if your system doesn't conform closely to one of these (and one of ours didn't) you get very restricted access to file attribute information. Zircon tell me that the next release has a line based FTP client.

The rest of the basic set are there, Telnet, a POP Mailer, an NNTP news reader, *lpr*, a Gopher and customised Mosaic client; as well as servers for *rnp* and FTP. There is an extension kit with a pretty reasonable X server (X11R4 in this release), which supports both the windows in a window view and also allows the Windoze manager to look after the X clients on the desktop.

Another package includes everything except the X server, including support for client NFS, based on the Beame and Whiteside PC-NFS product. This is claimed to be compatible with Windows for Workgroups, but I wasn't able to test that. The NFS stuff worked well except for an unresolved problem with printing from an ancient version of 123 running in a DOS window. I couldn't work out if this was a problem with the BIOS INT14 redirector or 123. We found a workaround using *lpr*.

On the performance side, the TCP/IP stacks seemed adequate if not exciting. I think the fact that they were commodity items has influenced that result. A proprietary stack (like FTP's) might well be quicker, if harder to integrate with your other networking requirements. Given the woeful performance of most PC LAN cards, coupled with the restrictions of the ISA bus, I wasn't really looking for mind blowing transfer rates. In any case, the Air software is more than adequate.

The interface with Windoze, however, seems to be a strength of the Air tools. The X server in particular worked intuitively with the Windoze desktop manager, and got excellent performance on a Western Digital VESA VGA card based 486DX33 system. Given the difficulty it seems that even Microsoft have getting good graphics performance for their products, Spry seem to know what they are doing.

Finally, on the manuals and support side, there was lots of printed stuff which conformed mostly to what the software did. There were a few minor differences, but nothing which would get in your way. Support may be more questionable. I sent a support query to the published Internet address and didn't get a reply. As I resolved the issue myself, I didn't have to follow it up, but Zircon expressed surprise that I hadn't heard from them. Perhaps it was a Gareth Powell Internet mail problem!

In summary then, I'd be prepared to recommend the Air 2.0 software, though it might be worthwhile waiting for the 2.5(?) release due real soon which has the simple FTP client and an SMTP based mailer. At the AUUG special price it was an excellent deal. At the regular one-off pricing of \$399 for the core utilities (just Telnet, FTP/rcp and lpr), \$499 for the core plus tn3270, Mail, News and Gopher/Mosaic, or \$599 for all that plus NFS, it's competitive with the other PC TCP/IP kits. The X server is separately priced at \$199.

In fact, Softway liked the Air software enough to include it as an optional component of our Internet connection/firewall service package - for those customers who won't give up their Windoze environment just because they're connecting to the Internet.

INTERNET GATEWAY SERVER

FEATURING

- Ready to run UNIX based servers providing secure access to the Internet

OTHER SERVICES AVAILABLE

- Firewalling
- Integration to Existing LANS
- Integration to Existing MAIL



DESKTOP APPLICATIONS

- Suite of Windows Based TCP/IP Software from **SPRY** Inc.

The AIR Series™ 2.0

Highlights of features available from modules

- ◆ Telnet ◆ Network File Manager™ (ftp) ◆ NFS ◆ X Windows Server
- ◆ AIR tn3270 ◆ Line printer Redirector ◆ FTP Server ◆ RCP Server
- ◆ NetWare Virtual Terminal (NVT)

- Dial up SLIP/PPP Access for Personal Users

For further information contact us by email: info@zircon.oz.au
phone: (02) 317 4055 or fax: (02) 669 3241

ZIRCON TECHNOLOGIES

Ground Floor, APEX Building, 925 Botany Road, Mascot NSW 2020

SPRY - AUSTRALIAN MASTER DISTRIBUTOR

Book Reviews

Here is this edition's collection of reviews, and as you can see, it is packed. Thank you to all the reviewers. Aside from those who received books from AUUG, we have one unsolicited review.

The highlight of this review section is obviously Stevens' TCP/IP Illustrated. This is also the first review of an Addison-Wesley book, as part of our new arrangement. We also have reviews of Prentice-Hall and O'Reilly and Associate books, from various reviewers.

All these arrangements means that we will have lots of books for review. The current practice is to post a note to the newsgroup *aus.org.auug* when we have new books available. Unfortunately, this disadvantages members without network connections, or on the end of a low speed link. For people in such a position, either mail, via the AUUG PO Box, or fax me on (02) 717 9273 (note the new number), with your contact details and preferences.

Frank Crawford

TCP/IP Illustrated, Volume 1 The Protocols

by W. Richard Stevens
Addison-Wesley
1994, 570 Pages Hardback
ISBN 0-201-63346-9

Reviewed by
Michi Henning
CiTR
<michi@citr.uq.oz.au>

I had great expectations in this book, and I wasn't disappointed. Stevens has produced yet another classic, written in his usual lucid and precise style. If you want to find out what makes the Internet tick at the protocol level, this is the book to get.

Stevens concentrates on the practical aspects of the Internet protocols, examining in great detail how the protocols work at the packet level. Protocols covered include SLIP, PPP, IP, ARP and RARP, ICMP, UDP, TCP, plus many others. In fact, I am not aware of any Internet protocol that has been left out. Not only are the protocols explained, but related topics and applications are covered as well. For example, Stevens provides excellent explanations of how traceroute and ping use the protocols to do their work, and the chapter on SNMP contains background information about MIBs, as well as the basics of ASN.1 and BER.

Every chapter contains many examples that illustrate the protocols in action. Usually, this includes network packet traces that demonstrate a specific aspect of a protocol. Together with

Stevens' clear explanations and annotations, the examples allow you to see how things work, and bridge the gap between theory and practice.

The overall focus of the book is on the practical, and is not meant for beginners. It is assumed that you have a basic understanding of both network theory and the Internet. For example, you should know what a CRC checksum is for and you should understand the need for domains and routing. However, if your knowledge has become a bit rusty, there are enough background "memory joggers" to help you along.

Stevens never fails to mention the authoritative references for the various protocols, so if you want to dig into all the gory details, you know where to look, and the book is almost worth buying for its excellent bibliography alone. The index is well organised, and makes it easy to use the book as a reference.

If you are looking for a text on theoretical issues, you will be disappointed. However, if you have a basic knowledge of networking in general, and the Internet in particular, this book will allow you to learn about all the nuts and bolts, without getting buried by information theory and standards documents. The emphasis is on the "how does the Internet do it", not on the "why" or "how else could it be done".

In his excellent "UNIX Network Programming", Stevens concentrated on network programming at the API level. In this book, he explains what happens below the API. Read them both, and you have all the necessary ingredients for turning yourself into an Internet guru.

**TCP/IP Illustrated, Volume 1
The Protocols**

by W. Richard Stevens
Addison-Wesley
1994, 576 pages
ISBN 0-201-63346-9

Reviewed by
Danny Yee
Sydney University
<danny@cs.su.oz.au>

Stevens is well known for his books on Unix programming. In the first volume of *TCP/IP Illustrated* he deals with the specification and behaviour of the protocols that make up the TCP/IP suite. He begins at the bottom of the network stack, with the link layer protocols, and works his way upwards, dealing with IP, ARP, ICMP, routing, UDP, IGMP, DNS, TFTP, BOOTP, TCP, SNMP, telnet, FTP, SMTP and NFS (among others). Chapters on tools like ping and traceroute are included, and a `tcpdump` program is used throughout (on a real network) to allow us to actually watch the protocols in action on the wire; we are always kept in touch with what is happening at the link layer.

The focus is very much on how the protocols work in practice rather than on the theory behind them. So the discussion of RIP includes a detailed look at the protocol's behaviour on an example network, but only mentions the counting to infinity problem in passing, and ASN.1 is only given a brief description, since "the details of ASN.1 and BER are only important to implementors of SNMP". If you are primarily interested in the theory behind the algorithms and protocols then this will be frustrating, but if you are interested in the protocols from an practical perspective then it will probably be a welcome simplification.

TCP/IP Illustrated is not an introductory book: the treatment is more systematic than pedagogic and a fair amount of knowledge is assumed. (So, for example, SLIP and PPP are discussed in chapter two along with the other link layer protocols; this would probably be confusing to someone without much networking background.) This approach does make it easy to find things, however, and, together with a thorough index, enhances the volume's value as a reference. There are useful exercises at the end of each

chapter (with solutions at the back), which make it suitable as a textbook for those who already have some acquaintance with networking.

For many years the recommended survey of TCP/IP protocols has been Comer's *Internetworking with TCP/IP*. While I certainly wouldn't suggest that that book has been superseded, since it has a rather different approach, *TCP/IP Illustrated* is definitely serious competition. Particularly attractive features of Stevens' book are its coverage of different Unix versions (BSD4.3, Sun, SVR4, Solaris, BSD4.4 and others), its consideration of what the protocols actually mean in terms of "packets on the wire" and its concentration on issues of practical importance. As mentioned, complete beginners and those interested in theoretical issues will probably prefer other books, but for many people I think *TCP/IP Illustrated* would be the book of choice on TCP/IP.

C++ and C Debugging, Testing and Reliability

by David A. Spuler
Prentice Hall
1994, 338pp + diskette.
ISBN 0-13-308172-9

Reviewed by
Greg Rose
Sterling Software
Greg_Rose@sydney.sterling.com

Subtitle: The prevention, detection and correction of program errors

This book by a Lecturer in Computer Science is a useful reference for any C++ product development team, despite failing to live up to the promise of its subtitle. It delves deeply into some issues of memory allocation and deallocation in C++ which are clearly hard to get right, and gives useful boilerplate code for avoiding the problems.

The publication is in three parts: Part 1 of the book is devoted to "Techniques and Tools"; Part 2 is a "Catalog of Common Errors"; lastly there is a diskette containing some snippets of code and libraries useful in developing robust programs.

For my money, the real meat in the book is in the later chapters of Part 2. It is here that the author's encyclopedic knowledge of the

languages is demonstrated, and where many man hours of programming could be saved. There are chapters on C++ class errors, ANSI library related errors, pointers and memory allocation, preprocessor macro errors, to name just a few of them. Appendix A is about "Symptom-based error diagnosis", and could be an extremely useful reference.

Unfortunately, Appendix A is also the only part of the book which even addresses the primary emphasis of its title, namely the *act* of debugging. While the book details an amazing number of things which can go wrong, and seemingly endless iteration on the subject of how to add printouts to the code, just how to find an offending insect in a program is left to this appendix. Indeed, while Chapter 2 is entitled "Debugging techniques", the bulk of this chapter is about the pros and cons of various preprocessor #defines for doing debugging printout (and the bugs that they each may introduce...).

Testing techniques also get a fairly superficial treatment.

Where Part 1 shines is in the pervasive flavour of reliability. Constant attention is paid to the future and long term maintainability of programs, and whether the program will fail gracefully in the field. These are important real world issues.

I had a few specific complaints about the book's readability. In parts it is extremely repetitious. Extremely repetitious. It repeats things endlessly. The same points (slap) ... I'm sorry. The text is not pitched at any particular level of experience in the reader; while this is unavoidable to some extent, I don't think it is handled well. Sometimes the simplest errors are explained in terms recognisable only to members of the ANSI X3J11 standards committee, and at other places quite subtle C++ errors are exhibited with minimal explanation. Lastly, my blood boils on Dennis Ritchie's behalf when I read statements like "It is about time compiler writers added array bounds checking." I beg your pardon, which languages are we talking about here? Finally, repeated reference is made to the code on the diskette, but none of it is ever reprinted in the book. I read books on aeroplanes, in the loo, in the bath, in comfy chairs, in fact if I was in front of a computer I probably wouldn't be reading the book. So it is annoying not to be able to see the referenced code.

I'm sorry that this review, upon rereading, sounds so negative. I put this down to the fact that I am desperate for a book which really does address the hard issues of finding the bugs. This is a valuable publication, particularly the C++ aspects, and should be a mandatory reference library member if you write C or C++ code and subsequently try to sell it.

Learning the UNIX Operating System 3rd Edition

by Grace Todino, John Strang and Jerry Peek
O'Reilly & Associates, Inc.
1993, 92 Pages
ISBN 1-56592-060-0

Reviewed by
Greg Black
Greg Black & Associates
<gjb@gba.oz.au>

Most books reviewed in *AUUGN* are of interest to the magazine's readers for their own purposes, but an introduction to Unix for new users is worthy of attention because of its potential as a book that *AUUGN* readers might recommend to **other** people. As such, it is – or should be – an important book in a world where Unix is becoming steadily more entrenched as one of the operating systems of choice for so many different users.

Before going any further, I should state a bias: I have been a fan of O'Reilly books for quite a few years. More than a dozen of them now grace my shelves and the only one of those that disappointed me was very much improved in its second edition. I expected to like (and to be able to recommend) *Learning the UNIX Operating System* as well.

One important task for the technical writer is to identify the book's audience and to pitch the content and style so that it will be useful to that audience. The back cover of this book says:

If you are new to UNIX, this concise introduction will tell you just what you need to get started, and no more. Why wade through a 600 page book when you can begin working productively in a matter of minutes?

Perhaps the truth is that a user can begin working productively pretty quickly with that

600 page book if it is well planned and written, and some of them are.

If you are a sysadmin, you might be interested in this book if you could give a new user a login, a password and the book – if that would then enable the user to go away and start using the system productively. You might also be interested if you were somebody who had just got an account on a public access Unix machine, or even if you had just acquired *linux* or *FreeBSD*. And you might be tempted by this book if you were a personal computer user who was wondering what it would be like to start learning about Unix. All those people would be disappointed. Todino et al. attempt to get people started in less than 100 pages and they fail in their mission.

Let's take a look at the more significant reasons for that failure. Readers with plenty of Unix experience will appreciate that the apparent triviality of some of what follows needs to be seen in context – this is a book for **beginners**, so care and consistency are of real importance.

In this kind of book, examples are critical to the reader's understanding, so accurate and clear examples are paramount. The Preface says that examples will be "set off from the main text in typewriter-style characters" and that those parts that you as a user "would type if you were trying the example are shown in bold characters." This is pretty standard stuff and I think it works well. But they don't do this – nothing appears in bold, even in the illustrative example that immediately follows those words. In trivial examples, this failing is not an insuperable obstacle to understanding; but the examples of the use of *mail* and some of the login sequences would be almost impossible for a new user to understand without the bold type.

A real irritation stems from very poor proofing of the text, something that is hard indeed to excuse in a third edition. Again, some of the typos would not interfere with even a new user's understanding (e.g., *gmod* for *good*, or *asterick* for *asterisk*) – although both of these would have been caught by almost any spell-checker worth the name. However, suggesting that the user type *f* (rather than *fg*) to bring a job into the foreground is likely to lead to confusion; and confounding the names of *lpr* and *lprm* in the generally unclear discussion of printing is hardly likely to inspire confidence.

Another critical factor is a pedagogical issue. Modern education practice recognizes that people learn much more successfully when they are treated as intelligent and capable. But this book avoids explanations at all costs. So we are told that passwords do "not appear on the screen as you type" – but not why. We are told that we "should *not* end a session by just turning off [the] terminal!" Even though an understanding of the reason for this rule would take only a sentence and would make it more likely to be remembered, no explanation is offered.

In many places, recognition is given to the fact that there is really a lot more to be said about a particular topic that cannot be covered in this small introductory book. Unfortunately, this seems to be done mainly in order to help sell other (expensive) O'Reilly books. In some cases, there is also a suggestion to "see your online documentation" – but the half page devoted to *man* is not anywhere near enough to get a new user over the initial hurdles with that arcane (even if well-loved) component of the Unix environment.

Even when sound advice is given, it is often not followed. For example, the reader is told that it's a good idea to organise files into directories, yet the only figures that portray files and directories show **all** the user's files in the HOME directory.

There are many small errors of fact. Some of these will never become apparent to somebody who is at the level for this book, but they still should have been avoided. For instance, in the discussion of file ownership, it is stated that "files you create will be marked with the name of your group", even though this only applies to some Unix variants. When explaining the meaning of the output of *ls -l*, it says about group permissions that the "next three characters show permissions for other members of *your* group", although this actually applies to the *file's* group.

A very strange choice was made in the discussion of the *chmod* program. Most introductory books use symbolic modes because they are easier to understand and master; they usually mention absolute modes in passing as an alternative for more skilled users. This book **only** discusses absolute modes, giving a cookbook for a subset of possible permissions and no explanation of what the magic numbers

mean, or of how they are constructed, or even that they are octal. Furthermore, the explanation of the consequences of various permissions for both file and directory access is cloudy at best. The discussion of the *umask* command is similarly obscure.

Although the Preface mentions the existence of System V, BSD and other variants, no distinction between these is made in the text. Therefore, in the discussion of printing (covered in less than three pages), although both the BSD *lpr* facility and the System V *lp* facility are mentioned, nothing is said about how you could determine which to use or why one or the other might (or might not) be available. This discussion is further handicapped by some unfortunate typos.

I'll only mention two more items from my notes - I've got lots more, but this is already a very long review. For some reason, the book likes to use the word *enter* whenever it is illustrating command input. This leads to peculiar effects when a two-column format is used to show what is to be done in one column with the actual command (plus "enter") in the other. One sample (to illustrate copying a distant file to the working directory) shows in the command column:

```
Enter cp /etc/passwd  
myfile
```

Leaving out the "enter" would have let this fit on the one line. But, if they had to leave it in, some explanation of the need to type this command (and others) on one line would have been an appropriate step.

Appendix B is entitled "Reference - Commands and Their Meanings". It lists 17 general Unix commands, four BSD commands and four System V commands. This is hardly a comprehensive list. Surely they could have printed the commonest titles from Section 1 and included a notation about their origin.

In the final analysis, the attempt not to overwhelm a new user has resulted in a book that gives simplistic prescriptions without explanations. The lack of care in production (typos, factual errors, the scrappy index) will win few fans for either O'Reilly or for Unix. An experienced colleague (or one of the alternative and better books) will be much more use to the neophyte Unix user than this book.

Solaris Desktop Integration Guide

by SunSoft
SunSoft Press, Prentice Hall
1993, 204 pages + Diskette
ISBN 0-13-035726-X

Reviewed by
Catherine Allen
Prentice Centre, University of Queensland
<cccathy@brolga.cc.uq.oz.au>

The Desktop Integration Guide is an attractive, useful manual. It gives a good overview of desktop integration concepts, defining and explaining terms and jargon both on the fly and in a glossary at the back of the book. The authors state that they assume prior X11 programming experience, but I found that I could understand the code examples with only a working knowledge of C and C++.

I haven't written (and don't currently write) integrated programs for the Solaris desktop. As a novice, I found this book an excellent introduction to the area. The first chapter is an easily-understood overview of desktop integration and each integration technology is introduced with a broad overview.

Three integration technologies are discussed: Selections (also called Drag and Drop), Classing Engine and ToolTalk Services. TNT, OLIT and XView selections technologies are covered. The Classing Engine is covered in depth in this manual. Selections technologies and ToolTalk services are discussed in more depth in reference guides and manuals, which are listed in the Preface. Integration using DeskSet is also discussed in the context of the three integration technologies.

Experienced programmers looking for a manual to cover the differences between SunOS and Solaris may be disappointed - I didn't once see a mention of SunOS nor a comparison between the two. X11 programmers wanting to become familiar with the tools available in Solaris will find most chapters useful but can safely skip the first chapter and all overviews.

The authors presented concepts well, in clear and concise language. Data structures and functions are discussed in the framework of the concepts presented. A code example follows each structure or function, which I found helped my

understanding. Examples of DeskSet tools (eg File Manager, Binder) were dotted through the book, which give an idea of the look and feel of the screen and of the programming environment.

I was itching to run the sample programs on the accompanying disk but couldn't get access to a machine running Solaris 2.X and OpenWindows 3.1. I'm sorry I'm unable to review the disk for you.

One criticism is that the index is very short, which will make using the Guide as a reference difficult. This is offset by the detailed table of contents, which suggests that this book was written as an introductory guide and not as a reference text.

Overall, I found the book interesting and easy to understand. Terms and jargon are defined from scratch; concepts are discussed in simple language; details of data structures and functions are given; code examples are clear and helpful; and there are figures to illustrate the OpenWindows look and feel. The Desktop Integration Guide is suitable for novices (familiar with C or C++) and for experienced software developers.

SCO Open Desktop/SCO Open Server User's Guide

by The Santa Cruz Operation Inc
SCO Press/Prentice Hall
1994, 289 Pages
ISBN 0-13-106816-4

*Reviewed by
Lindsay Trewin
Bourke Johnston Systems*

This book is a part of SCO's documentation set for the SCO Open Desktop and SCO Open Server environments. In particular it is a simple introductory text on using the SCO Open Desktop graphical environment, the Unix command line and the DOS Services provided with SCO Open Desktop and SCO Open Server.

This guide claims not to assume you know anything about the Unix system, or even about working with computers. The guide does not cover any administration side of SCO Open Desktop and SCO Open Server.

The first six chapters of the book cover the fundamentals of the graphical desktop. This includes an explanation of the desktop, icons and the various different pointers used. The basics of moving, sizing and selecting windows are also covered. Attention is also given to how files and directories are represented and manipulated in this environment. Plenty of graphics are used with examples in these chapters.

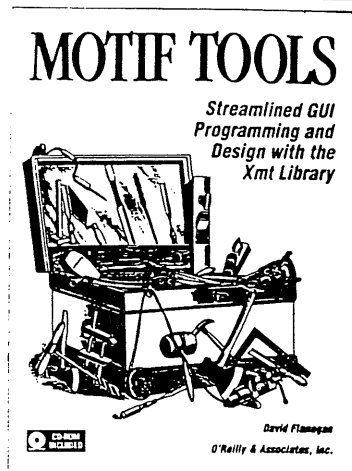
The next seven chapters are basically an introduction to Unix and its command line. Most basic commands are introduced, including grep, tar and vi. Fundamentals such as standard input and output, and pipes are covered.

Chapters 14 to 19, in some detail, describe the DOS and MS Windows compatibility (ie DOS Services) provided under SCO Open Desktop. These chapters assume a reasonable knowledge of DOS (and hence a knowledge of computers!). Chapter 19 is dedicated to installing and running MS Windows (versions 3.0 and 3.1) under the DOS Services. These chapters explain many of the limitations and features of the DOS Services and its MS Windows support.

The remaining 5 chapters cover networking. Reference is made to connections to UNIX systems (via TCP/IP and UUCP) and connections to LAN Manager networks. Connections to non Unix systems through ftp is touched on. Various different ways to do remote logins, command execution, copying of files and printing are covered. The sharing of files over networks is touched on. The usage of the command line mail program is also covered. Most information in these chapters is very brief.

The book is quite clearly written and easy to understand. It is well cross referenced and has many useful notes and tables. Many topics are glossed over to keep the content simple, as you would expect in an introductory guide. Due to the amount of the book dedicated to the DOS Services side of SCO Open Desktop it would be well suited to a DOS user moving to the SCO Unix environment. This book covers the essential information they would need to know to get working quickly, and later explore the Unix environment.

O'REILLY & ASSOCIATES, INC.



NEW BOOK

MOTIF TOOLS

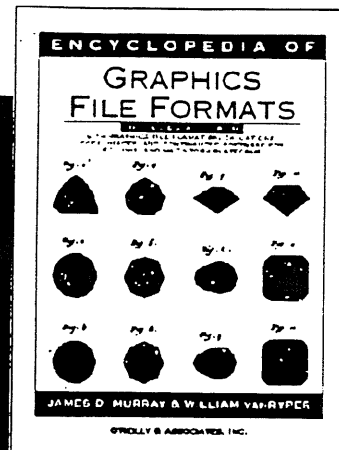
Application Design & Development with Xt, Motif, and the Motif Tools Library.

AUTHOR: DAVID FLANAGAN

Motif Tools and the Xmt programming library that accompanies it on CD-ROM offer resources to empower Motif programmers and dramatically speed up application development with the X Toolkit and Motif. The book guides the reader through the development of a complete application: from the first conception, through the design and implementation, and on to the finishing stylistic touches.

PRICE: \$110.00

ISBN: 1-56592-044-9



NEW BOOK

ENCYCLOPEDIA OF GRAPHIC FILE FORMATS

AUTHORS: JAMES D MURRAY AND WILLIAM VANRYPER

This Encyclopedia includes technical information on nearly 100 file formats, as well as chapters on graphics and file format basics, bitmap and vector files, metafiles, scene description, animation and multimedia formats, and file compression methods. Best of all this book comes with a CD-ROM that collects many hard to find resources.

PRICE: \$120.00

ISBN: 1-56592-058-9

AVAILABLE FROM AUSTRALIA'S LEADING TECHNICAL BOOKSELLERS

- NSW:** Dymocks City Business & Education
George St Sydney 224 0411
Dymocks Professional & Technical
Wynard Sydney 224 0444
Hotline Chatswood Sydney 415 2265
Dymocks North Sydney 956 6771
- QLD:** Dymocks Edward St Brisbane (07) 220 0152
- SA:** The Bookshelf Adelaide (08) 223 3899
Dymocks Adelaide (08) 223 5381
- VIC:** Tech Books Melbourne (03) 663 3951
McGills Melbourne (03) 602 5566
Dymocks Melbourne (03) 663 1344
- WA:** Rellim Booksellers Perth (09) 321 5684
Boffin Bookshop Perth (09) 321 5755
- A.C.T.** Uni Co-Op Bkshp
ANU Canberra (06) 257 2673

Or call WoodsLane for the stockist nearest you.

Distributed by **WoodsLane Pty Ltd**

A.C.N. 003 667 549

Phone: (02) 979 5944 Fax: (02) 9997 3348 EMail: woods@tmx.mhs.oz.au

AUUG BOOK CLUB & PRENTICE HALL AUSTRALIA

20% DISCOUNT TO AUUG MEMBERS

Please send me a copy/copies of the following book

- SOLARIS Desktop Integration Guide**
ISBN: 013035726X, Book & Disk, 1993, RRP \$59.95*
- SCO Open Desktop Users Guide 3.0**
ISBN: 0131068164, Paper, 1994, RRP \$59.95*
- C++ & C Debugging, Testing & Reliability**
ISBN: 0133081729, Book & Disk, 1994, RRP \$49.95*

*Deduct 20% from listed retail price

Name: _____ Organisation: _____

Position: _____

Address: _____

_____ Telephone: _____

Please send my book/s on 30-day approval (tick box)

Enclosed cheque for \$ _____ (Payable to 'Prentice Hall Australia')


Charge to me **OR** Company purchase Order No. _____

Please charge my: Bankcard Visa MasterCard

Credit Card No:

Expiry Date: _____ Signature: _____

Mail or fax completed order form to Prentice Hall Australia, PO Box 151, Brookvale NSW

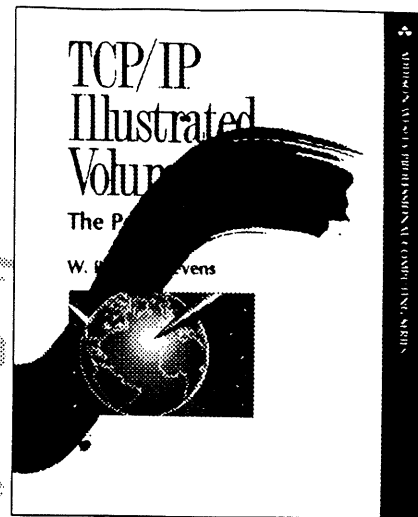
OR  Use our FAST PHONE SERVICE by calling Liz Guthrie.
SYDNEY (02) 939 1333

A.C.N. 000 383 406



Prentice Hall Pty. Ltd.
7 Grosvenor Place, Brookvale NSW 2100.
Tel: (02) 939 1333 Fax: (02) 905 7934

A Paramount Communications Company



20% Discount to AUUG Members

TCP/IP Illustrated Volume 1: The Protocols

W. Richard Stevens

The innovative approach of *TCP/IP Illustrated* helps readers at all levels to truly understand how TCP/IP really works – whether you need to write applications that run over TCP/IP, or you're responsible for managing and maintaining a TCP/IP internet.

TCP/IP Illustrated examines the entire TCP/IP protocol suite using a bottom up approach, starting at the link layer and working up through the application layer. The book also takes a functional approach, explaining features of the protocols in relationship to applications that use them.

TCP/IP Illustrated covers the following topics: Link Layer, IP, ARP, RARP, ICMP, ping Program, traceroute Program, IP Routing, Dynamic Routing Protocols, UDP, Broadcasting and Multicasting, IGMP, The Domain Name System, TFTP, BOOTP, TCP Overview, TCP Connection Establishment and Termination, TCP Interactive Data Flow, TCP Bulk Data Flow, TCP Timeout and Retransmission, TCP Persist Timer, TCP Keepalive Timer, TCP Futures and Performance, SNMP, Telnet, FTP, SMTP, and other TCP/IP Applications.

CODE 63346 (RRP \$50.95) AUUG Price \$40.75

ALSO FROM W. RICHARD STEVENS:

Advanced Programming in the UNIX Environment	Code 56317	(RRP \$51.95)	\$41.55
TCP/IP Illustrated Vol 2: The Implementation	Code 63354	Avail Jan '95	Price TBA

YES: I would like to order: QTY: _____ TITLE: _____

QTY: _____ TITLE: _____ QTY: _____ TITLE: _____

I enclose a cheque for \$ _____ **OR** Please charge my credit card No: _____

BANKCARD VISA MASTERCARD AMERICAN EXPRESS ID No: _____

EXPIRY DATE: / / AUUG MEMBERSHIP NO: _____ PHONE NO: _____ (Wk Hrs)

NAME: _____ SIGNATURE: _____

STREET ADDRESS: _____

STATE: _____ POST CODE: _____

Send your order to ►

Addison-Wesley Publishing Company

Unit A1, 6 Byfield Street, North Ryde, NSW 2113, Australia.
Telephone (02) 878 5411 Customer Service Fax (02) 888 9404



!AUUGN

The following letter appeared in AUUGN Vol 1 Number 4. Compare the information presented at the meeting to AUUG94.

UNIVERSITY OF GLASGOW



Computing Science Department,
THE UNIVERSITY,
GLASGOW, G12 8QQ.

TEL. 041-339 8833
EXT. 473/7458

5th April, 1979.

Dr. I. Johnstone,
Australian Graduate School of Management,
University of New South Wales,
P.O. Box 1,
Kensington,
New South Wales,
AUSTRALIA 2033.

Dear Ian,

I am returning, under separate cover, your tape containing the amendments to your third distribution, which we have distributed to most people in the U.K. who are using your software. Unfortunately we were unable to read the other tape with collected software from the U.S., though we tried on another computer in St. Andrews as well as our own. (As of the last two weeks our own drive is out of action).

We have just returned from the biggest-yet Unix meeting in the U.K., at the University of Kent at Canterbury. There were about 150 people there, and the main speakers were Ken Thomson and Brian Kernighan. The first day was meant to be a publicity exercise for those new to Unix, and the second day for the cognoscenti, but in practice both days were made open to anyone interested, so no user group business was transacted. A full report will be published in the next U.K. newsletter, (which must be soon!), but perhaps a summary of the highlights now might be of interest.

Principal item of hard news is that Version 7 is now available, though distribution may be held up because of delays in getting the manuals printed (they are much larger than Version 6 manuals). You can have a licence either for a normal PDP-11, a VAX or an Interdata 8/32. Goodies included are two C compilers (the system-specific one and the portable), new shell, lint, lex, learn (a CAI package for learning about Unix), and - guess what - Fortran 77. They don't seem quite sure whether to be proud or apologetic about this item! The new Shell is one of the largest programs on the distribution (!) but there was some relief that version 7 kernel need be only around 2K larger than version 6. Most utilities (Fortran 77 being one exception) will run on systems without separate I & D spaces. Indeed Bell already have version 7 running on the new LSI 11/23 (which has the full 11/34 instruction set, with memory management and floating point available on additional boards). Good news also that 'nroff' and 'troff' have been cleaned up and made compatible, and a standard set of useful macros is provided. Security has also been tightened up to some extent, with a new password encrypting system.

Most U.K. users were astonished to hear that one thing which has not changed in Version 7 is the default for "delete character" and "delete line" in the teletype handler - we thought we'd seen the last of ~~it~~ and ~~it~~! What was very clear was that version 7 is a "snapshot" of a still developing system, and indeed neither speaker seemed quite sure of when the snapshot was taken or exactly what it contained. The general feeling among users at the meeting was that the new tools provided with version 7 were too good to resist (though many had doubts about the new Shell). We were, however, relieved by the assurance that there would never be a version 8!

Ken Thomson finished with a presentation which might have been titled (but wasn't) "How I took on the world (at computer chess) and won". Truly a tour de force. Like all good ideas, incredibly simple once explained - what was astonishing was that no-one had done it before. His "chess machine" cost less than \$1000 to build, and was connected to an 11/70 via three DR11 interfaces. It could, he said, just as easily be attached to an LSI-11; the 11/70 acts mainly as a huge database of book openings and endings. A very clever method of organising the opening book enables him to decide if a position is in the book, and if so generate the next move, in under a second. The mark II version, using more modern technology, will be about 10 times faster. Ken was modestly sceptical about predictions that this would put Belle (the name of the program) in the Grand Master class.

Software available at the meeting included the latest release of the Vrije (Andy Tanenbaum) Pascal System. This included the option of either interpreting or compiling the EM1 intermediate code, with very good run-time diagnostics on the interpreted version. If you are interested I can probably arrange for a copy to be sent, even if our mag. tape does not get fixed soon. Also being distributed was release 2 of the Modula system - I guess you should request this direct from York if you want it.

Finally a quotation, attributed to Steve Johnstone, with which Brian Kernighan introduced his excellent sales campaign for Unix on the first day of the conference: "Using TSO is like kicking a dead whale along the beach". Unix rules.

Best wishes,

Yours sincerely,

Alistair C. Kilgour

Alistair C. Kilgour.

C++ - is it really a better C? †

Michael Henning
CiTR
University of Queensland 4072
michi@cit.ug.oz.au

One of the arguments often touted in favour of C++ is the ease with which C programmers can make the transition to an object-oriented paradigm. Since C++ is largely a compatible superset of C, there is a belief that C programmers can gradually and gently migrate to C++, without the need to spend large amounts of money on training or a different development environment. Further, C++ allows existing C source to coexist with newly developed code, and it preserves investments in existing C libraries.

Unfortunately, many projects find that their hopes are shattered once real C++ development gets under way. This article suggests that attempts to use C++ as a "better C" are likely to fail, and that a substantial investment is required to make a productive transition to C++.

In the simplest case, the transition to C++ can be made by not using any C++ features at all. That is, you simply use a C++ development environment to compile and debug C code. Unfortunately, there is a heavy price to pay for this approach. Your new C++ compiler will most likely be slow, produce comparatively poor diagnostics, and generate executables that are both larger and slower than those from your favourite C compiler. The C++ debugger will lack good C++ support and have more bugs than your C debugger. Tools to instrument and debug memory management code may not work correctly with C++. The code coverage tool will produce incorrect reports unless it is C++ aware. In short, current C++ tools lag behind C tools in features, reliability, performance and price. Unless these disadvantages are offset by some other gains, your project will lose, and just using C++ as a replacement for C does not achieve those gains.

What then about using C++ "gently"? For example, you could avoid the "hard" object-oriented techniques such as inheritance and polymorphism, and just take advantage of "nice and easy" language features, such as template functions, references, function and operator overloading, user-defined type conversions, and default parameters. All these can be used without getting too deeply into the object-oriented paradigm. However, whilst these features are certainly worthwhile using, they are notoriously difficult to use correctly. For example, the interactions of user-defined type conversions, overloaded functions and default parameters are subtle - so subtle, in fact, that the C++ language definition (the Annotated Reference Manual, or "ARM") needs to devote several pages of text to defining their exact semantics. Material of this level of complexity can hardly be considered "gentle". More importantly, it is unlikely that the increased costs of using a C++ development environment can be recouped by using the "gentle" features alone.

So it seems unlikely that you will be able to use C++ "gently" without making a loss. Is it possible to use it "properly" and come out ahead? There is no short answer, so here is the long one.

To use C++ productively is to use it in a way which offsets increased development costs by gains during the later stages of the software life cycle. This can only be achieved by leveraging the advanced object-oriented concepts of the language - inheritance, virtual functions, polymorphism and encapsulation. These of course are the "hard" object-oriented concepts, and you cannot expect staff to get on top of them by osmosis. Instead, you need to invest in C++ training, and allow a substantial amount of time for programmers to gain proficiency in the new language. C++ can be learned in a few days or weeks, but this is only a first step. After that, programmers need to learn how to use it effectively, and how to avoid its numerous pitfalls. C++ requires a new way of looking at problems, and

† This paper was originally published in the July/August edition of the ACS (QLD) Branch publication, The Source .

experience shows that a qualified C programmer will require 12 to 18 months to make the necessary paradigm shift. Further, to produce object-oriented code, you will need object-oriented analysis, design, documentation standards, quality assurance procedures, management techniques, and a whole host of other object-oriented things.

So, it seems there is no way to migrate to C++ and still make a profit, unless you go object-oriented all the way, by designing a framework and using objects from day one. The risks of asking a staff of twenty or more to "bite it off all at once" are unacceptable for many projects. However, there are a number of things that can be done to minimise those risks.

- Expect your first C++ project to take twice the time you would have spent building the same software with C. C++ will force a major shift in outlook, design approach and development procedures, no matter how much it looks like C. Budget for the additional overheads caused by this and reserve high expectations for project number two.
- Be sure to pick a comparatively small project when making the switch, and have at least one person with substantial object-oriented and C++ experience on the team. Be prepared to use that person as a teaching resource, rather than as the most productive designer or programmer.
- Expect to spend more time on design. C++ emphasises design more than C, which is beneficial in the long run, but an unusually long time may pass before the first line of code is produced.
- Allow time to iterate over the design, and expect to find design flaws during implementation. C++ design is complex, and it is difficult to spot design flaws before implementation. On the positive side, where a C project would have to start over, a C++ project will happily absorb such late changes, due to its better encapsulation of implementation details.
- Have the most experienced programmers build classes, and everyone else use them. C++ class design and implementation are decidedly non-trivial. Use the most highly skilled people to make sure the fundamental building blocks work correctly.
- Do not attempt to use everything all at once. C++ presents the uninitiated with a bewildering array of features and possibilities. It requires experience and judgment to consider the trade-offs involved. Have the most experienced staff develop boiler-plate solutions for various problems, and have them point out ways of using language features safely.
- Expect your compiler to have bugs, and to implement only a subset of the language. C++ is still evolving, and it has many features that were added only recently. Many compilers do not yet support exception handling, nested classes or templates. The semantics of some language constructs are still poorly defined, and not all compilers implement them identically. Finally, C++ is a large and complex language, which is reflected in the number of bugs you can expect to find in your compiler.
- Consider using a subset of the language. Many of the "gentle" features of C++, such as operator overloading, are excellent candidates for banishment. Be sure to take advantage of the "hard" features, otherwise you will lose most of the object-oriented flavour of the language, and most of the long-term benefits.
- Don't reinvent the wheel. Spend time evaluating general-purpose class libraries, and learn how to use them. There are dozens of commercial and public-domain libraries which offer data types such as generic lists, trees and hash tables. Using such libraries will allow work at a higher level of abstraction and reduce coding effort substantially.

- Don't expect to get reusability for free, as a side effect of developing in C++. Reusability adds a new dimension to software development. Instead of building something that will do for the job at hand, you need to build something that will work for future projects with unknown requirements as well. Reusable code needs to be more general, more robust, more extensively tested, more extensively documented and more extensively maintained than a once-off solution. Don't expect reusable code unless you are prepared to fund it.

After all this advice, you may feel that switching to C++ will be a lot worse than expected. If that is the case, this article has achieved its aim, namely to dispel the myth that C++ is a painless way of easing into object-oriented development. The choice of programming language is only one aspect of taking such a step. Beyond that, you need to build a project framework which supplements the language with object-oriented analysis, design, quality assurance and project management techniques. Much of the costs of switching to object-oriented software lie in supplying this infrastructure, not in the costs of switching to a new language. The apparent similarity of C and C++ has misled many project teams into cutting infrastructure corners, and ultimate failure. By avoiding the mind-set of "C++ is a better C", you will allow your team to succeed where those teams have failed.

Managing Mailing Lists with Majordomo †

Frank Crawford

ANSAMS, Private Mail Bag 1, Menai 2234
(frank@atom.ansto.gov.au)

ABSTRACT

Mailing lists are becoming an increasingly important way of distributing information over the Internet and to related networks, *e.g.* ACSnet. One reason for this is that, unlike network news, it is a very focused medium, with all people on the list having an interest in the topic.

This paper will discuss the requirements for a mailing list, the administration of a simple list and the administration of a complex mailing list. Tools such as Majordomo, which help with complex mailing lists, will be discussed.

1. Introduction

Access to information is the major driving force behind the explosive growth of the Internet^[1]. Many of the people accessing the Internet today are doing so based on reports of the current technology. Unfortunately, much of this technology doesn't scale well.

An important aspect of some of these technologies is that they aren't restricted to the Internet, but rather, can be used over any "message-connect" network. One facility that has been available for many years, in fact it preceded the Internet and is still heavily used on such networks as *BITNET*, is mailing lists. Mailing lists are an extension of simple mail, where mail sent to a special address, is *reflected*, or remailed, to people who have expressed interest in receiving it. Mailing lists are generally organised around a topic of common interest.

On the Internet, *UUCP* and *ACSnet* mailing lists had been superseded by Network News, however, the sheer volume of news is becoming a problem in itself. For example, there are over 2000 newsgroups and over 50Mb of news being generated a day. A site which wishes to keep all newsgroups available online for a week, needs to allocate over one gigabyte of disk space. Further, the volume of news is increasing at a spectacular rate, doubling every year.

Because of these problems many users are moving away from netnews as a means of obtaining their information. Many of these people are moving back to the use of mailing lists. This reduces both the volume and complexity of network load, and also, generally, increases the signal to noise ratio.

2. Setting up of a Mailing List

Mailing lists are easy to set up and even easier to use, however, the exact details depend on the system software used for handling and processing network mail. By far the most common mail system is *sendmail*^[2] which I am concentrating on in this paper.

A mailing list is really only a mail address which reflects all mail sent to it back to all the addresses which have subscribed to the particular list. This leads to a number of different functions of a mailing list that have to be controlled. These tasks include:

- establishment of appropriate aliases,

† This paper was first presented at SAGE-AU'94 and published in the Proceedings of the Conference.

- maintenance of the subscribers addresses,
- correction of mailing problems, and
- other special requirements, *e.g.* digest production.

2.1 Establishing an Alias

All mailing lists need a special mail address to be set up, under *sendmail*, this is done through setting up an *alias*. These aliases are set up generally in a file called either */etc/aliases*, or */usr/lib/aliases*¹. The simplest form of such an alias is:

```
alias: user1,user2,user3,...,userN
```

When mail is sent to this alias, it is then automatically sent out to each of the users (*user1*, *user2*, *user3*, ..., *userN*) specified in the alias. This format is only suitable for very small mailing lists which change very infrequently. Further, changes to such an alias also requires special permission, usually restricted to the system administrator.

A second format for setting up aliases is much more suitable for establishing mailing lists, especially those that are controlled by ordinary users. The format of such aliases is:

```
alias: :include: /path
```

Note that the string **:include:** has to be specified as shown above. In this case, the mail addresses, to be used for this alias, is taken from the file */path*. Further these addresses are only evaluated at the time the alias is invoked. The file containing the mail addresses does not have to be owned by the system administrator, and in fact, is generally owned by the maintainer of the mailing list.

One other advantage of reading from such a file is that it is much easier to automate the process of adding and deleting entries for the mailing list.

2.2 Administering a Mailing List

It is all very well setting up the mechanics for a mailing list, but it is much more important to establish methods for interested people to make use of the list. A number of conventions have been adopted over the years to help this administration.

The first and most important is that there is a special alias related to each mailing list for administrative matters. This alias is constructed by adding the string **-request** to the end of the mailing list name. For example, if the mailing list name is **sage-au** then requests would be directed to **sage-au-request**.

In fact this is not strictly correct for most automated lists. Such lists normally have a special alias for the administration, however, they do return instructions on the correct address to use and the commands to send.

Once a request has been received to join (or *subscribe*) to the mailing list, or to leave (or *unsubscribe*) from the list, then it is the list maintainer's responsibility to update the aliases file appropriately. Until the task is done, the new subscriber will not be able to receive mail directed to the list. Obviously, these two types of request are not the only ones that may be received. Other common ones include:

- requests for information about the list,
- requests for information about other users on the list,
- requests for old articles,

1. The actual location is controlled in the *sendmail* configuration file, **sendmail.cf**.

- requests to only receive the articles as a group rather than individually, and
- requests to change a users address.

For manually administered lists all these tasks take time, often considerably more than the administration of the address list. These are also tasks that can be automated, or at least set up as tasks for remote users to do themselves, for example by making files available for *anonymous FTP*.

2.3 Dealing with Mailing Problems

One of the most time consuming tasks of any mailing list is dealing with mailing problems. This starts from ensuring that the mail address given on subscription is valid. Despite instructions, some subscribers give incorrect or invalid information, some of which is obviously wrong. List maintainers have to at least peruse the supplied addresses and consider if they are reasonable.

The next problem is that even a reasonable address can be wrong, *e.g.* mistyped user name. To take care of these problems the maintainer has to take two different actions. The first is to take care of local mailing problems, and the second is to propagate an address for remote errors to be returned to.

To handle local mailing errors, an alias of the form:

```
owner-list: listowner
```

needs to be created. This alias is used by *sendmail* to return local processing errors. If this is not defined, then any errors would be reported back to the original poster, who would be confused, and may not be able to do anything about it.

To address remote problems there are two possible solutions, both of which may be undertaken at the same time. You can either modify the sender recorded in the mail envelope to be the list maintainer, or add an **Errors-To:** header directed to the list maintainer, for all mail sent to the list. This rewriting of the mail header takes some program intervention between the original posting and the posting to the mailing list.

A simple example of a mailing list which takes care of these error is:

```
sage-au: "|/usr/lib/sendmail -fsage-au-request sage-au-real"
sage-au-real: :include: /usr/local/lib/sage-au
sage-au-request: frank
owner-sage-au: sage-au-request
```

Unfortunately this set up still has a lot of problems, primarily with maintenance. It would take considerable work by the local maintainer, who does need a local account, to ensure that such a setup works².

3. Managing Large Mailing Lists

The next step in the management of a large mailing list is to automate much of the process. There are a number of packages available to undertake this task for *sendmail* based sites. The three most popular packages are:

- Majordomo – a *perl* based system written by Brent Chapman,
- Almanac – a package written in C by the Oregon State University, Extension Service, and
- ListProcessor – a system written by Tasos Kotsikonas, the oldest and most extensive of the publically available systems.

2. There is also a permission problem with this solution, which would affect users mailing on the local machine. However, this problem is beyond the scope of this paper.

The *Majordomo* package is the one used by SAGE-AU for the administration of its many mailing lists. The current version is 1.92, recently introduced to supersede version 1.62³. It is available by anonymous FTP from *FTP.GreatCircle.COM* in the file *pub/majordomo.tar.Z*.

The features of *Majordomo*[3] (in common with the other packages) which make it useful for SAGE-AU mailing lists include:

- automatic subscription and removal of users,
- remote administration of individual lists by administrators,
- addition of relevant mail headers for correct error returns,
- automatic processing of help and list information messages,
- automatic access of list archives, and
- inclusion of a number of utilities for mail digests, archiving, moderation and approval.

3.1 Using *Majordomo*

The core of *Majordomo*⁴ consists of two programs:

- *majordomo* – the mailing list maintenance program, and
- *resend* – a program for the modification and verification of mail prior to being mailed out to the list.

While it isn't appropriate to give full details about the installation of *Majordomo* here, as they can be found in the distributions, some points are worth discussing.

The most important feature of *Majordomo*, is that the *majordomo* program is normally run out of a special alias. This alias is generally called **majordomo**, although it can be anything appropriate. This alias is used for all administrative and maintenance mail for the various mailing lists under the control of *Majordomo*. Also, to overcome various permission problems within *sendmail*, all *Majordomo* aliases are run through a special wrapper program. Thus the alias for *majordomo* would look like:

```
majordomo: "|/path/to/majordomo/wrapper majordomo"  
owner-majordomo: frank  
majordomo-owner: frank
```

As can be seen there are two other special aliases, one, **owner-majordomo** is for *sendmail* error processing, the other, **majordomo-owner** is used by *majordomo* for internal problems and messages.

When *majordomo* is run, it reads a configuration file, which defaults to */etc/majordomo.cf*, for the setting of various general parameters, such as the directory holding the mail lists, *etc*.

For administration of a mailing list *majordomo* accepts commands embedded in the body of mail messages. These commands fall into two different categories: those available to all users, and those applicable to list administration. The commands available to all users are:

- **subscribe** – subscribe to a named list,
- **unsubscribe** – remove or unsubscribe from a named list,
- **which** – find out which list you are on,

3. In fact, version 1.90 was introduced to test out new features planned for the next major release (2.0), however, corrections to a few problems quickly caused the release of version 1.92.

4. In this paper references to *Majordomo* (uppercase 'M') refer to the package, while *majordomo* (lowercase 'm') refer to the program.

- **who** – find out who is on the name list,
- **info** – retrieve the general introductory information for the named list,
- **lists** – show the lists served by this *Majordomo* server,
- **index** – return a list of files associated with the named list,
- **get** – get a file associated with the named list,
- **help** – retrieve a more detailed description of the commands accepted by *majordomo*, and
- **end** – stop processing commands (useful if your message includes a signature line).

The commands available to list maintainers all include a password set on a list basis. These are:

- **approve** – approve a subscribe or unsubscribe command,
- **passwd** – change the password for the list,
- **newinfo** – set the information text associated with the list,
- **config** – retrieve the configuration file associated with the list,
- **newconfig** – validate and update a new configuration file,
- **writeconfig** – write a new configuration file, intended mainly for use after an upgrade of *Majordomo*, since it will add new keywords, and
- **mkdigest** – force the generation of a digest for the named list.

The other side of managing a mailing list with *Majordomo* is the establishment of the separate lists. This consists of the creation of a number of files in the directory specified in the **majordomo** configuration file, and the establishment of a number of aliases.

The files to be created include:

- *list-name* – the file containing the mail address for the list,
- *list-name.passwd* – the default password for the list,
- *list-name.info* – the introductory information for the list, and
- *list-name.config* – the configuration file for the list.

While, the aliases to be created include (for the list *sage-au*)⁵:

```
sage-au: "|/path/to/majordomo/wrapper resend -l sage-au \  
          -h ansto.gov.au sage-au-outgoing"  
sage-au-outgoing: :include: /usr/local/lib/sage-au
```

```
owner-sage-au: frank  
owner-sage-au-outgoing: owner-sage-au
```

```
sage-au-request: "|/path/to/majordomo/wrapper request-answer sage-au"
```

There are a number of features about a mailing list that can be set by the list maintainer. These include:

- the reply address for errors and replies,

5. The *sage-au* alias is split over multiple lines for readability, it should be entered on a single line.

- moderation of all postings – all postings are sent to the moderator for approval before posting,
- setting a closed list – one that needs approval before a subscription or a removal from the list,
- setting an auto list – one that automatically processes all subscription and removal requests,
- setting an open list – one that processes simple requests for subscription and removal, and passes others to the moderator for approval, and
- setting a private list – one that isn't normally listed in *list*, *etc.* requests to *majordomo*.

These options are set in the mailing list configuration file. Note previous versions of *Majordomo* specified these options either as arguments to the program or by the existence of various files.

In addition to these simple mailing list facilities, *Majordomo* supplies programs for *archiving* and the production of *digests*. As with *majordomo* and *resend* many of the parameters can be set in the list configuration file. This configuration file also includes comments, which should be sufficient to enable the list maintainer to update the information appropriately.

Through the use of *majordomo* it is possible to configure most features of a mailing list remotely. In fact, this is one of the major attractions of *Majordomo*, it is possible to manage the mailing list entirely through messages mailed to *majordomo*.

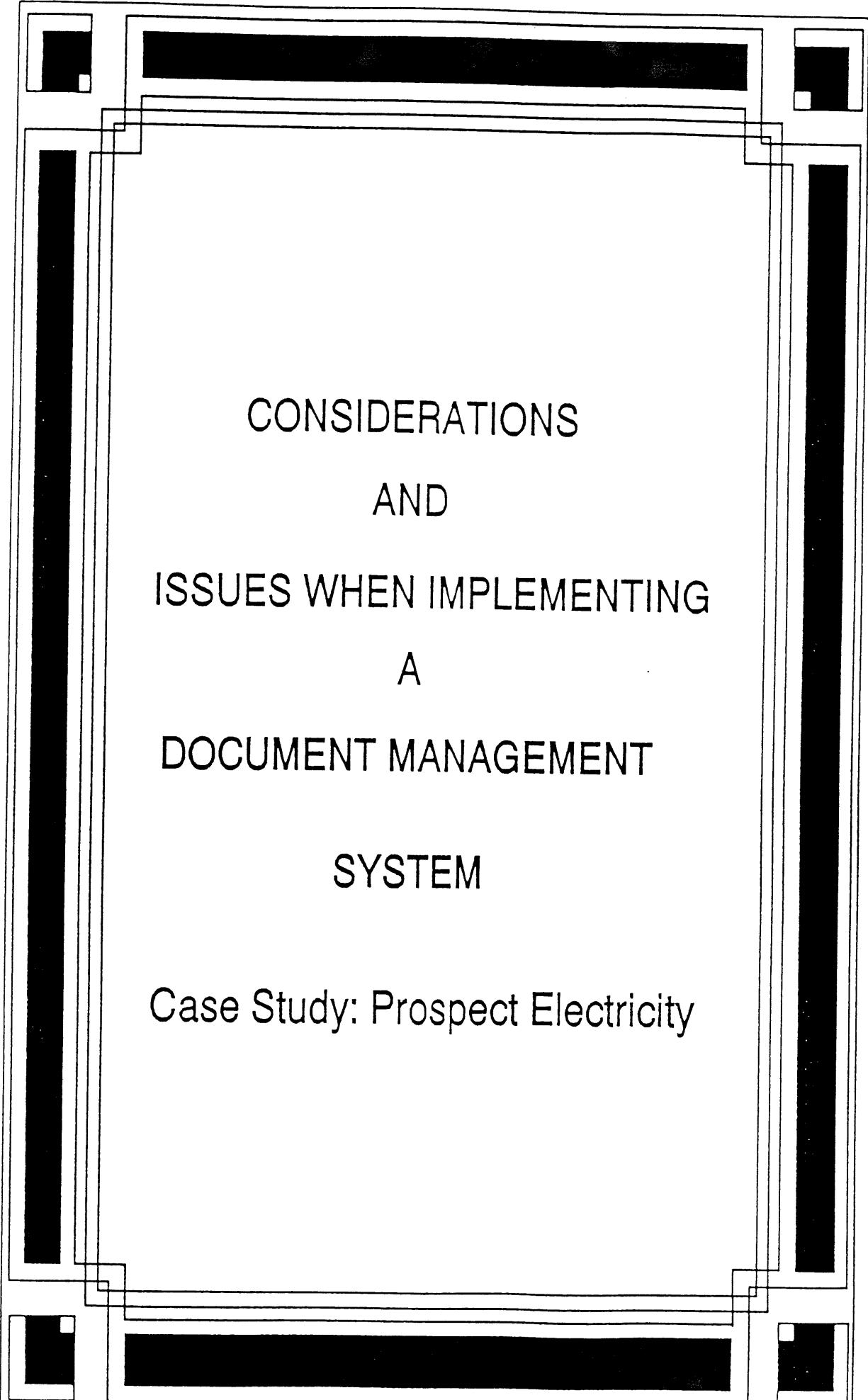
One thing to note, although these configurations can be set by the list maintainers, they have to be initially established by the local system administrator, as they often involve the creation of special aliases.

4. Conclusion

The creation of a mailing list is a simple process, consisting of the creation of appropriate aliases, however, as the list becomes larger or more complex features are requested, the use of a package becomes essential. Further, these packages, usually include features that both add to the functionality, and ease the administrative burden.

5. References

- [1] Ed Krol (1994): *The Whole Internet User's Guide and Catalog, 2nd Edition*, O'Reilly & Associates Inc., ISBN 1-56592-025-2.
- [2] Bryan Costales, *et.al.* (1994): *sendmail*, O'Reilly & Associates Inc., ISBN 1-56592-056-2.
- [3] John Rouillard, Brent Chapman, *et.al.* (1994): *README* accompanying *Majordomo* 1.92 distribution.



CONSIDERATIONS
AND
ISSUES WHEN IMPLEMENTING
A
DOCUMENT MANAGEMENT
SYSTEM

Case Study: Prospect Electricity

INTRODUCTION

DMS, Re-engineering, Workflow, RMS, Text Retrieval and so on..... are more than just acronyms or new concepts in the computing industry. These systems are beginning to make large inroads into the day to day running of businesses trying to get an edge on their competitors, but what are they ?

Most people have heard of them but very few have actually had the chance to experience first hand the advantages that such systems provide to users.

The purpose of this paper is to examine the components of a Document Management Systems and address some of the issues involved in the implementation of a Document Management System (DMS). Prospect Electricity will be used as a case study to share some of the experiences gained in what is, and continues to be, a large and complex project.

STRUCTURE

For easy of reading, the paper is divided into three main areas:

- a. *Document Management Systems.* Their origin and key components.
- b. *Prospect Electricity Case Study.* The hardware and software Prospect has installed and the applications developed to improve the running of their business.
- c. *Issues Addressed.* An examination of a number of the issues encountered and some considerations to be taken into account when implementing a Document Management System in an open systems environment.

DOCUMENT MANAGEMENT SYSTEMS

Before we can understand what a Document Management System is, we must first examine their origin.

Where Did Document Management System Originate From ?

I firmly believe that Document Management Systems evolved from automated Record Management Systems ie. systems designed to control the inwards and outwards movements of physical documents and files. Traditionally these were database systems sitting on mini computers linked to users by dumb terminals dedicated to the task of maintaining control over physical files and documents. As users demand for information became stronger and the sheer volume of information stored on machine readable media by businesses grew, companies began to demand a way to manage, organise and control not only their physical documents and files but also their electronic records, be they electronic mail, word processing or other application records, eg: Auto CAD, all of which foRm part of the corporate information base.

Document Management Systems have evolved to provide an easier way to control corporate records and take full advantage of the efficiencies provided by electronic systems.

Object Of A Document Management System

In general terms, the object of a DOCUMENT MANAGEMENT SYSTEM is to:

- a. Manage information from new or existing textual sources across a wide range of mediums.
- b. Develop a logical organisation of this information into coherent data structures that can be manipulated and queried at personal workstations, on minicomputers and mainframes, or across local or wide area networks.
- c. Provide users with an easy method of navigating electronic information.

Components Of A Document Management System

To many people, a Document Management System can be seen as an add-on to the concept of Record Management Systems, in that Document Management Systems are designed to manage and control other forms of information, not just paper.

Indeed in Prospect's case, a traditional Records Management System has formed the front end to the Document Management System. However, there are many components to a Document Management System and in Prospect's case it comprises of the following

components:

- a. *Records Management System.* A systems developed to maintain management of paper records particularly during the transition from paper to electronic records.
- b. *An Imaging System.* Comprising a scanning subsystem, mass storage (magnetic and optical disk) and a variety of servers connected via a high speed network.
- c. *Workflow Programming.* The automating of a business process through the management of the flow of information as it is processed, shared, manipulated and compiled.
- d. *Optical Character Recognition(OCR).* Software developed to convert paper documents in bit-map form into data that can be manipulated by a word processing package or indexed for text retrieval purposes.
- e. *Fax Gateway.* An electronic gateway developed to send and receive faxes from a network
- f. *Text Retrieval.* Software developed to help users search for certain textual phrases or words in a large scope of documents.

PROSPECT ELECTRICITY - CASE STUDY

Origin

Prospect Electricity was established in 1957 and has its origins in Local Government. It is the second largest electricity distribution authority in NSW providing power to more than 1.3 million in an area of 16,115 square kilometres. With an annual turnover of \$1 billion a year and a staffing level at approximately 2,500 staff, Prospect is seen to be an extremely successful corporation, maintaining a triple A rating.

Current Environment

In 1989/90 Prospect undertook the installation of an electronic business network which would enable all staff to communicate between each other irrespective of their location, and also to connect to all the mainframe systems from which they need information.

WordPerfect Office operating on a Novell Operating System was selected to be installed as the user interface. At a similar time, a strategy to convert some of our existing applications from the Fujitsu mainframe and DecVax stations to an Open systems environment was being developed.

It was also around 1990 that the corporate team made a decision to commence work on a Document Management System.

Several years later, Prospect now has over 1,500 users on the electronic business network with most of the business systems running on various Unix platforms. A Windows interface is now standard and the Document Management System has over 300 users which will grow to the 1,500 users on the network as training resource and network considerations permit.

Systems Overview

At Prospect the purpose of the Document Management Systems is to control and protect the ever growing flow of paper within Prospect. It will improve efficiency, competitiveness and services by reducing the administrative effort required to process and maintain paper documents and drawings. The current hardware configuration of the Document Management System comprises of the following:

- a. A Sun 6/90 which has the specific purpose of running the Records Management Software and database. It is linked to the IBM RS/6000 via an Oracle product called SQL Net, which allows image information to be transferred directly from the image server to its Oracle tables.
- b. An IBM RS/6000 which is the image server running AIX UNIX. Its purpose is to run the image software and database.

- c. A Cygnet optical disk Jukebox connected to the RS/6000. It holds 25, 5-1/4" (write once read many times -WORM) optical disks, each with a storage capacity of approximately 650 megabytes.
- d. 1 x Fax Server which runs the fax software controlling four lines, each being capable of receiving and sending faxes.
- e. 2 x A3/A4 scanners.
- f. 1 x A0 Context scanner from large plans and drawings.
- g. 2 x Hewlett Packard Laserjet 111 printers and associated PC with special high resolution graphic cards.
- h. 2 x A3/A4 image printers and associated PC. These are designed to provide relatively high speed output of business sized A4 documents, plans and drawings up to A3 size. Larger sized documents can be reduced or printed as tiles, although in the future an A0 plotter capability will be provided.
- i. Image workstations. Prospect has standardised on 486 SX PC's with 25MHZ, 4Mb RAM and 120 Meg H/D usually combined with one of the following screens:
 - i 14" SVGA colour monitors for the normal users,
 - ii 17" colour monitors for the developers and users of workflow systems, or,
 - iii 19" Cornerstone Greyscale Monitors for staff displaying images most of the day, such as the registry clerks.

The Software used by the Document Management System is as follows:

- a. *Records Management System*. This is an Oracle Forms 3 based application called "Collector" from Logical Technologies in Melbourne. It runs through a terminal emulator called PC Connect which is DDE Enabled. In general, the control functions that the Records Management System provides are as follows:
 - i Search and Request,
 - ii Item Movement,
 - iii File audit,
 - iv File and Document Registration,
 - v Thesauruses control,
 - vi Archiving/Disposal, and
 - vii Various reporting and maintenance functions.

Some of the details stored by the RMS are as follows:

- viii Title,
 - ix Status (ie Active, Closed, Archived),
 - x Security, (0 to 9)
 - xi Company/Author,
 - xii Keywords,
 - xiii Related Files,
 - xiv Date Created
 - xv Image File Number. The IFN is used via Dynamic Data Exchange (DDE) under a Windows environment to link the RMS and Image systems in order to display related images.
 - xvi Document and File movement details and locations.
 - xvii Document and File types or classifications (EG: Customer)
- b. *Image Software.* The Tower Technology imaging system was selected to perform the following tasks:
- i Scan all inwards and record all outwards correspondence to be registered on the Records Management System,
 - ii Display, retrieve, rotate, zoom and print the scanned and converted images,
 - iii Integrate with a Facsimile gateway conversion utility, to allow images to be registered into the RMS and conversely allows images to be faxed externally, and
 - iv Integrate with an OCR server to allow for the conversion of bit-map image data into text.
- c. *Workflow.* After an extensive tendering process a workflow product called WorkMAN by Reach Technology in America (marketed by Starcom Australia) was selected for the development of Propert's workflow applications. The product runs under Windows and over several networks including Novell and Banyan Vines. This product combined with a number of third party products can communicate (using DDE) with the image system to display images and also communicate with the Oracle database to retrieve and write information to the Records Management System. This product itself is a E-Mail based product rather than a database.
- d. *Text Retrieval.* This is currently the subject of an open tender and a product will not be selected for a number of weeks. However, in general terms, the product selected will have the following capabilities:
- i Provide a native Microsoft Windows user interface,

- ii Index both WordPerfect Documents and Oracle database entries,
- iii Perform as a DDE client, and
- iv Be able to integrate into Prospect's current computing environment.

SYSTEM RESULTS

Some of the advantages the Document Management System has provided to Prospect are as follows:

- a. Simultaneous and fast multi-user access to information.
- b. A truly Corporate information base where staff can interrogate and display incoming, internal and outgoing correspondence.
- c. A reduction in the physical transfer of paper and associated delays.
- d. Automated workflow processes with the reduction in associated processing times.
- e. A significant reduction in the time taken to find and research information.
- f. Staff receiving incoming documents faster through the electronic distribution of faxes. (described in detail in the Issues and Considerations section of this paper)
- g. A reduction in the number of paper files stored and maintained.

ISSUES AND CONSIDERATIONS

Some of the major issues experienced in the implementation of the Document Management System are as follows:

1. *Image Traffic.* A major consideration in the implementation of any imaging system is network traffic, ie the networks capability to handle the large amounts of data associated with image traffic. Taking into consideration that an average A4 size page scanned in at 200 DPI is approximately 40K bytes when using Group 4 compression, this represents a significant amount of data traffic when compared to an average word processing page of 2K bytes. Multiply 40K by the number of pages received, and by the number of users retrieving images at the same time and suddenly your network could be nearing or reaching its capacity, and thus significantly affecting response times. To overcome these potential problems, image traffic was confined in the early stages of the Document Management System to a dedicated LAN, on which resided the majority of the image users. This LAN was bridged to the Prospect backbone server for textual traffic and occasional image traffic. Since that time, the Prospect backbone has been upgraded to FDDI and users throughout Prospect including depots (via 2Mb WAN links) have image access. To assist in the management of image traffic and to optimise response times it is envisaged that in the future, image subsystems will be installed at designated depots. Depot specific files will reside on these subsystems yet depot staff will still have access to Corporate information by the central database.
2. *Backup and Recovery.* At Prospect we have been rather lucky in that we have never had to test our Document Management System recovery plan and we hope that we never will. Compared to other systems at Prospect, the Document Management System backup and recovery plan is rather complex. This is mainly due to the number of components which comprise the DMS all of which are interlinked in some manner. When designing a backup and recovery system consideration had to be given to the RMS, the Imaging system (ie database), the images residing on optical disk, other forms of electronic files and the workflow applications. As it turned out, the approach we adopted in designing the backup and recovery plan was rather simple. We took a common time to run backups for all the individual systems, and on top of that tried to use the inherited features of each environment. For example, since the RMS is Oracle based we inherited the roll forward capability. In addition, our Image system can restore its database from optical disk.
3. *Paper Conversion.* Backfile conversion, date forward conversion, partial conversion or no conversion. Which option is best I cannot really answer, it depends upon your system requirements. At Prospect, an assessment of work involved to convert all existing files was undertaken, and after much deliberation over the complexity and the time and resources required to complete the task, a date forward conversion approach was adopted. As a result, all files since the implementation date (15 September 1992) have been maintained in image form and all files prior to this date will be maintained in their physical form until such a

time as they are "closed".

4. *Data Conversion.* As part of the Document Management System conversion, Prospect was required to complete a data conversion from an old Records Management System to the new one. Given that the old RMS provided for no structured or controlled input, eg. Thesaurus control, the result was somewhat staggering. Over 3,000,000 keywords were created, the majority of which have since been deleted or at least amended to meet the Thesaurus standards. As an example of the impure data that needed to be corrected was that there were more than 300 variations of the keyword Prospect Electricity or Prospect County Council. This, combined with misspelling, abbreviations and despite thorough testing of all contingencies, required many weeks of man hours to sort out.

5. *Integration.* One thing that can sometimes be over-looked when examining Prospect's Document Management System is the significant amount of integration that was required between Prospect's current computing environment and the variety of products selected that comprise the Document Management System. At Prospect, the DMS comprises of products from four different suppliers mainly due to the inability to find the one supplier who could meet all requirements. This issue of integrating all these different software products as well as the new hardware into Prospects' current environment has been a difficult one. Some of the more interesting issues involved are as follows:
 - a. *Communication Protocols.* Initially, when we first introduced the Imaging System and RMS into Prospect, the current communication protocol was FTP version 2.03. Because at the time the only available communication protocol that supported the Windows socket library was Arpa, we had to install Arpa on to the users workstations. The maintenance of the two protocols was a rather large headache and it was not until latter on when FTP released its Windows socket library that we were able to overcome this problem.

 - b. *Monitors.* As mentioned earlier, for those users who had to view a large number of images per day such as registry clerks, we supplied 19 inch grey scale CornerStone monitors. At the time, due primarily to cost considerations, colour monitors were out of the question. Since that time, the prices of the large high resolution colour monitors have reduced significantly. As a consequence, all future purchases of large monitors for those staff with high image usage will be colour.

 - c. *Windows.* Because access to the DMS was via Microsoft Windows, initially we had to install Windows locally on the users workstation. This was required because the Windows on the network was policed in such a way that the groups within Windows were unable to be saved, ie when DMS users would execute or exit from Windows on the network they would receive group file errors. When a user had ten or so groups, twenty of so

warning messages would be displayed which as you can imagine can get very frustrating. The procedure of policing Windows has since changed, to allow users to maintain their own groups and Windows drivers are kept locally on the users workstation.

- d. *Product Integration.* The issues of integrating different products has mainly be addressed by the insistence of a GUI (Windows) front end and mandatory requirements placed on the product suppliers. For example, all products at the users interface support the Windows protocol of DDE, meaning different products are able to share and exchange information.
6. *Fax Gateway.* The DMS provides the facility to convert Group 3 incoming facsimile to Group 4 and vice versa for outwards facsimiles. As the facsimiles are already in data form, the opportunity arose to automate the distribution process and remove the requirement to print and physically distribute the faxes. The automated process that was developed is as follows:
- a. Incoming faxes are received as images and registered into the Records Management System.
 - b. When registering into the RMS, certain "trigger" cause documents entries to be inserted into the workflow queue.
 - c. The workflow server retrieves the entry and identifies it as being a fax. (Using keywords eg: FAX)
 - d. The workflow server then creates a DDE session with WordPerfect Mail for Windows and by issuing a number of DDE commands instructs Word Perfect Mail to send a message to the Action Officer (ie the addressed) with an attached ASCII file comprising of the relevant IFN (Image file number).
 - e. When the Action Officer receives the mail message they are instructed to double click on the attachment icon.
 - f. Windows looks for an association between the attachment and launches the Tower software resulting in the image of the fax being displayed.
 - g. The image id displayed for a period of one minute. At that time the user is prompt to close the image display. If no answer is received to the prompt the image display software is automatically closed.
7. *OCR.* Due to network and data storage demands associated with imaging systems, pressure was initially applied to use the OCR server to convert all incoming documents to data form. Given the volumes and diversity of document types involved however, this was never a realistic option with the equipment installed. Instead, the OCR facility has been provided to all users in such away that they are

the ones who determine if a document is to be converted, eg. for Text Retrieval purposes. The user simply searches for the document using the RMS. Once the document has been located, the user displays the image of the document and selects the OCR and format options from the image window. This in turn sends the request to the OCR server, and the bit-map form is converted into WordPerfect formatted text or ASCII as required.

8. *Expert Assistance.* Due to the complexities involved in the implementation of a DMS, Prospect sought expert assistance very early in the project life cycle. As a result of an open tender evaluation, Prospect selected Opticon - Australia Pty Ltd, to fulfill these needs. This has proven to be a sound decision. The employment of expert help has certainly minimised the risks involved in such a large and complicated task as well as lowering the learning curve of Prospects current staff.
9. *User Acceptance.* With the amount of change in regards to computer systems within Prospect over the last two years, it comes as no shock that the users also had rather a steep learning curve. The registry clerks at Prospect not only had to learn a new computer system but they also had to learn to navigate a new user interface, Windows. In Prospects' case these users were moving from a dumb terminal interface to an interface that was "driven" by a mouse. As users become familiar with Windows, other problems began to appear. For example, for no apparent reason, some Unix processes began running wildly "chewing" large amount of resources on the host computer and gradually resulting in a slow system. After some investigation into the problem, a strange pattern began to appear. Our users had become so familiar with the Windows interface, that after entering a large query, while waiting for the result their impatience got the better of them and they clicked on the RMS window and opened another RMS session. Unbeknown to them, their original query was completing an unstructured search across the entire database.

SUMMARY

In summary, the considerations and issues discussed in this paper may not be applicable to all Document Management Systems, I just hope some of the issues discussed can provide an insight into Document Management Systems and acronyms like DMS, Re-engineering, Workflow, RMS, Text Retrieval have more meaning.

An Update on UNIX-related Standards Activities †

by Nick Stoughton
USENIX Standards Report Editor
<nick@usenix.org>

Report on POSIX.7: System Administration

Martin Kirk <m.kirk@xopen.co.uk> reports on the April 18-22, 1994 meeting in Lake Tahoe, Nevada:

This is the first snitch report on the POSIX Systems Administration Working Group since it was renumbered from P1003.7 to P1387 – you don't want to know why it had to be renumbered, believe me.

As part of this renumbering exercise, the sub-projects were re-numbered as followed:

P1003.7.1 -> P1387.4 Printer Administration
P1003.7.2 -> P1387.2 Software Administration
P1003.7.3 -> P1387.3 User and Group Account Administration

The eagle-eyed will have noticed that P1387.1 is not used in the above list. For reasons of compatibility with ISO document numbering, this number has been reserved for a possible future project to produce an overview of the Systems Administration projects.

Having dealt with the boring bureaucratic stuff, what is actually happening inside P1387?

The Printer Administration standard, which is based on Palladium, is still in ballot resolution. The first ballot period closed back in 1993, and the process of dealing with the ballot objections is currently nearing its end. The result of the first ballot was 51% approval, with the threshold for final approval of the standard being 75%.

If a balloter's objections are all accepted and the appropriate modifications made to the document, the balloter's vote is automatically converted from negative to affirmative. Once all the ballot objections have been addressed, the revised document will be re-circulated and balloters have the opportunity to change their votes based on the modifications that have been made.

If the standard reaches the 75% threshold after this re-circulation, it will be approved. If not, a further ballot resolution cycle will start. This process continues until either the standard is approved or it is obvious that the threshold will never be reached and the proposed standard is withdrawn.

The Software Administration project is about to enter the ballot process. The first ballot is scheduled for May 29 – June 29, and the ballot resolution process should begin at the July POSIX meeting. The ballot draft will be D13, which should have only minor changes from D12 which was produced after the January meeting. The P1387.2 document addresses the issues of software installation – a packaging layout, a set of information about a software package, and a set of utilities for manipulating both the packages and the information.

The following reports are published in this column:

- Report on POSIX 7: System Administration
- Report on Standards for Formal Description Techniques (FDTs) and Programming Languages
- Services for Reliable, Available, and Serviceable Systems (SRASS)

Our Standards Report Editor, Nick Stoughton, welcomes dialogue between this column and you, the readers. Please send email to <nick@usenix.org> .

† This is a reprint from ;login, the USENIX Association Newsletter, Volume 19 Number 3

With the current document about to go into ballot, the working group is now looking at what future work need to be done to build on the initial functionality in the first document. Possible extensions, to be covered in a new document, include:

- Software distribution
 - queueing and queue management
 - fan-out
- Extensions to current admin
 - software update
 - version management (same version for multiple architectures)
 - roll back
 - apply, commit, reject
 - synchronous changes (1000 machines with scheduled atomic commit)
- Management of PC software
- Licensing
- Network install of base operating system
- Interoperability
- Distribution library management
- Queries and report generation
- Management of multiple catalogues, roots, etc.
 - potential inclusion of host object
- Software administration with a distributed computing infrastructure
- User software configuration
- Installation of object libraries
- Internationalisation issues
- Compression
- Management of software collections
- Diskless/server support
- Attributes/options (distribution, checkpointing, ancestors)
- Security levels

Clearly the above list represents a much wider range of topics than could reasonably be included in a single standard. The group will identify what it will tackle next over the next couple of meetings, and a new project proposal is likely towards the beginning of 1995.

The User and Group Account Administration project intends to hold a mock ballot before the July POSIX meeting. This process is intended to provide an informal opportunity to test the acceptability of the overall approach before the formal ballot takes place. If you are interested in taking part in the mock ballot, contact Louis Imershein <louisi@sco.com>. The formal ballot is currently scheduled for December 1 to January 1. (Ballots will only be accepted if packaged in Christmas gift-wrap!)

The scope of this standard is the creation, deletion, and modification of user and group accounts within a POSIX.1 conformant, distributed, heterogeneous environment. The interfaces defined in the document are based on SVID3, SCO and the public-domain "shadow" package.

The other activity of note at the April meeting was a Birds-of-a-Feather session that solicited suggestions for future P1387 projects. Once the Software Administration document goes into ballot, the group will have free bandwidth which would allow one or more new projects to be initiated.

Suggestions made at the BoF included the following:

- Device Driver APIs
- Process and Thread Status APIs
- Thread Checkpointing APIs
- Virtual Memory Support APIs
- Shared Code APIs
- Resource Management APIs
- APIs to support the functionality of ps and df
- Quota Management
- Common Core Services (standard "traditional" interfaces)
 - File system related issues
 - commands (mount, df, find, du, fsck, etc.)
 - file formats
 - system calls
 - NFS type issues
 - Authorization related issues
 - commands (su, login, chown, chgrp, etc.)
 - file formats (passwd file, group file)
 - User environment
 - Boot issues
 - commands (init, halt, shutdown, etc.)
 - rc scripts
 - Process issues
 - commands (ps, killall, etc)
 - Miscellaneous issues
 - UNIX accounting (includes sar)
 - terminfo / printcap
 - cron/at
 - sendmail
 - syslog
- Standard Directory Structure

The BoF provided input into the process of determining what P1387 might tackle next. Any new project proposals are likely to appear at the beginning of 1995.

Report on Standards for Formal Description Techniques (FDTs) and Programming Languages

John Hill <jhill@bb.unisys.com> reports on Standards for Formal Description Techniques (FDTs) and Programming Languages:

This article focuses on US standards activity in the area of Formal Description Technologies (FDTs), especially those most appropriate for programming languages. It is not a tutorial on those FDTs. The FDTs themselves are far too technically complex to describe meaningfully in a single article. Instead, this tells you what FDTs are, in a somewhat abstract, non-technical way, and what is taking place to develop standards for FDTs.

What is an FDT?

An FDT is a meta language for expressing processes or inter-entity relationships. Proper use of FDTs assures, to a great degree, that the process exhibits certain desirable, mathematical properties including, for example, symmetry and completeness.

That's a mouthful. At some advanced stage of personal frustration, we have all resorted to reading a manual. Probably our first exposure to an FDT came not from computer software, but from Sears. The illustrated parts breakdown, and assembly instruction for your power drill, garbage disposal, or lawn mower are each expressed using FDTs, albeit simple minded FDTs. The breakdown diagram and instructions each have some meaningful properties. They identify exhaustively:

- a. the elements involved in the process or entity being described
- b. the relationships among those elements.

We, as users, trust the diagrams and lists to such an extent as to be furious with inaccuracies – did you ever successfully assemble a child's toy by following the instructions?

In the realm of computer software, your first exposure to FDTs probably came with learning of your first programming language. Do you recall the bewilderment you felt when you saw a railroad diagram? Well, the formal name for that is Backus-Naur Format, BNF. BNF is an FDT developed specifically to explain programming language syntax clearly. It shows syntactical flow, options, requirements and inter-syntactical-element relationships. BNF provides a non-ambiguous meta language for describing processes, most frequently computer programming languages.

I have a personal view, albeit likely full of technical errors, on the value of the use of FDTs for specifying programming

language syntax. It works for me. You may find my perspective sufficient to invoke your interest in participating in development of corresponding standards.

Allow me the indulgence of telling you.

The application arenas in which computers are being used continue to pervade human life at an ever increasing rate. Human physical and emotional safety as well as intellectual progress are being increasingly subjected to invasion by digital machines controlled by software programs. The programming languages themselves must be provable, in a mathematical sense (i.e., subjected to predicate calculus) in order to ensure that the foundation upon which the application is built, is itself complete, secure and robust.

In essence, an application programmer unwittingly makes assumptions of the verifiability of the programming language being used. A programmer typically recognizes the assumption that the source code is reliably translated into the machine's object code. The programmer has probably given little thought to whether the language of choice possesses properties such as completeness and symmetry.

Standardization of FDTs for Programming Languages

There are two subcommittees of JTC1, the international organization responsible for developing standards for information technology, working on FDTs. JTC1/SC21, for OSI upper layers, database systems and open distributed processing, is one. Their standards activities for FDTs include ASN.1, ESTELLE, and LOTOS. You may recognize them as most widely used for networks and communications. I will not address these FDTs. If you need additional information on them, contact members of X3T2 or X3T5.

The other subcommittee of JTC1 that is working on FDTs is JTC1/SC22. (You may recall an earlier article in which I described SC22 to considerable detail.) As you might imagine, formal verification of programming languages is a keystone of the bridge linking the application (coded in the programming language) to the functional and technical specifications of the application.

While Working Group 15 (WG15) of SC22 is working on POSIX, WG19 of SC22 is working on standards for two FDTs: VDM-SL (Vienna Definition Method – Specification Language) and Z. Derek Andrews, Leicester University, England, is the Convener. He, together with his colleagues in the UK, has extensive academic and practical experience with both VDM-SL and Z. They meet once per year for a week.

WG19 has been in business for about three years, starting with VDM-SL. The WG19 project for Z is about one year along. In terms of progress, the Committee Document (CD)

ballot period for VDM-SL just closed. It is reasonable to expect a second CD ballot for it. Z is still in the working draft stage of development. SC22:WG19 meets once per year with participants delegated by their national standards body.

There is a US group, X3J19, focusing all its efforts on VDM-SL and Z. X3J19 is the US Technical Advisory Group (TAG) to SC22/WG19. Currently Bill Harvey, Robert Morris College, provides the leadership for X3J19. X3J19 meets four times per year for two days per meeting.

Summary

Development of standards for FDTs, especially those for programming languages, continues apace. The work of the relevant committees is vital to the success of programming languages, and perhaps overall computer software quality leading to society's safety. You may wish to get involved.

Report on SRASS – Services for Reliable, Available, and Serviceable Systems

Arun Chandra <achandra@vnet.ibm.com> reports on the Jan 10- 14, 1994 meeting in Irvine, CA.:

Are you interested in Fault Tolerance, High Availability, Reliability, Serviceability, Maintainability? If so consider joining the "Fault Management Study Group" at the next POSIX meeting. By the way this probably is the last time that our study group will be called the "Fault Management Study Group." The group approved the new name to be "Services for Reliable, Available, and Serviceable Systems Study Group." If you see any reference to any of the above two groups it's us.

October was the first meeting of this group, following Birds-of-a-Feather (BOF) sessions at the two previous meetings. The status of the group is a "Study Group" preparing a "Project Authorization Request" (PAR). The PAR will go up for review at the April meeting. If approved we will become an official POSIX working group. Healthy participation at the next meeting would indicate that Fault Management is something organizations are interested in sending people to work on. This is one of the basic criteria for PAR approval especially in these hard times.

[Editor's note: the April meeting has now in fact passed, and the PAR was deferred for three months in order to fully understand the scope of the work to be undertaken. The appeal for support is now even more critical than when Arun first submitted this report.]

A number of existing documents are being studied as base documents. To obtain a list of the documents or the documents themselves please contact the chair of the group – Helmut Roth. Also, the detailed minutes of the October '93

or January '94 meetings can also be obtained from Helmut. The group started its January meeting with presentations on the state of the art in Fault Management. Dr. F. Cristian from University of California at San Diego gave two talks on the subject. Other presenters were from IBM, Johns Hopkins-University, Unisys, and JPL. The group once again worked on the list of submitted requirements to identify services that can be standardized. The Fault Management process model developed at the October meeting was once again worked upon. This process model allowed the identification of the APIs involved. After intense discussion the group identified four key APIs. These are:

- 1) Detection of abnormal conditions during system operation,
- 2) Logging and notification of abnormal conditions,
- 3) Classification and analysis of abnormal conditions for fault diagnosis, and
- 4) Corrective actions for system reconfiguration and recovery. These APIs are the group's immediate focus.

This study group spent an intensive week, looking at a wide range of topics in the fault management arena. Writing the PAR was another accomplishment. The group is optimistic that the PAR will be approved in April.

If you are interested in more information on the group why not contact the group Chair Helmut Roth, <hroth@relay.nswc.navy.mil>, or the group Vice Chair Arun Chandra, <achandra@vnet.ibm.com>.

An Update on Standards Relevant to USENIX Members †

Nick Stoughton
USENIX Standards Report Editor
<nick@usenix.org>

This column used to be entitled "An Update on UNIX Related Standards Activities." However, UNIX is now a licensable trademark of X/Open. In order to be able to use it, a system must support the interfaces described in their SPEC 1170 document, otherwise known as XPG 4 release 2. I don't have all the 1170 interfaces, so I guess I shall have to rename the column "An Update on Standards Activities Relevant to USENIX Members." USENIX, of course, is not a trademark of X/Open. Thanks to Peter Salus for reminding me of the original occasion that this footnote was used.

Is SPEC 1170 the answer to a maiden's prayer? How will it affect the future of Open Systems and UNIX as we now it? 1170 has one major difference from its predecessors. It represents the union of all the current major implementations of UNIX and not the intersection. Earlier versions of the X/Open Portability Guide (XPG), while not standards as such, provided a list of interfaces that worked the same on all the X/Open branded systems. POSIX took that concept one step further, as well as making a real international standard. POSIX, though obviously derived from UNIX, has allowed non-UNIX people into the game. But to whom is X/Open accountable – the system vendors? Or the end users? Or their shareholders?

The strength of POSIX is that it is accountable to vendors, users, general interest, and academic groups. They all have power through the ballot process. The standard is based on consensus between these people. But the ownership of SPEC 1170 lies entirely with X/Open and its members. It is a tool that gives them considerable power. And you've all heard the old adage: "power corrupts, absolute power corrupts absolutely . . ."

Theoretically, because 1170 is a complete specification, missing only hardware specific interfaces, no one will ever need to use interfaces outside the specification. Both XPG and POSIX had "missing pieces" – interfaces that vendors could provide to make their system that little bit better. With SPEC 1170, there seems to be an attempt to say "no one will ever need any interfaces we haven't already thought of" – the end of systems programming in UNIX.

After several years, Jim Isaak is stepping down from the post of PASC (Portable Applications Standards Committee) chair. Jim's leadership has seen POSIX become one of the most important and respected standards in the world of Open Systems. I am sure you will all join me in wishing him well in his new role, defining standards for the National Information Infrastructure, or "Information Super-Highway."

His successor in PASC will be Lowell Johnson, of Unisys. Lowell is currently chair of the 2003 test methods groups, and has worked within POSIX since its inception. His vision is of a more efficient process, producing relevant standards faster.

The following reports are published in this column:

- POSIX.5: ADA Bindings
- The Distributed Service Working Groups
- SRASS: Services for Reliable, Available, and Serviceable Systems

Our Standards Report Editor, **Nick Stoughton**, welcomes dialogue between this column and you, the readers. Please send your comments to <nick@usenix.org>.

† This is a reprint from ;login, the USENIX Association Newsletter, Volume 19 Number 4

Report on POSIX.5: Ada Bindings

Robbie Robbins <robbie@lfs.loral.com> reports on the April 18-22, 1994 meeting in Lake Tahoe, Nevada:

The primary charter of the POSIX.5 group is to produce Ada language bindings to POSIX standards. The standard for Ada language bindings to 1003.1-1990 was published in 1992 as 1003.5-1992. The working group is now working on three projects:

- POSIX.5a – A few problem fixes to the 1003.5-1992 Standard
- POSIX.5b, comprised of the documents formerly named POSIX.20, the Ada binding to POSIX.1b, (the real-time extensions, otherwise known as POSIX.4) and Mutexes, Condition Variables and Thread I.D.s from POSIX.20a, the Ada binding to POSIX.1c (the real-time threads extensions of POSIX, otherwise known as POSIX.4a). One day, we will get used to these new numbers!
- P2003.5, the test assertions document for 1003.5-1992.

In addition, at the July meeting, the group will add a MOTIF/Ada project to produce an Ada Binding standard to 1295-1993, the Modular Toolkit Environment standard.

The POSIX.5 Interpretations Committee issued an interpretations document (1003.5-1992INT) in March containing responses to seven problems in 1003.5-1992 encountered by users of the standard. To give a flavor of the work, here is the list of titles and interpretation numbers:

1. Missing parameters from FLUSH_IO generic operations
2. Text on reading from a pipe
3. Text on writing to a pipe
5. Behavior of read when interrupted by a signal
7. Can IS_A_TERMINAL detect/report errors?
9. TEXT_IO files should not have EXECUTE rights by default

The committee also worked on a number of additional interpretations requests:

4. Error checking in POSIX_Configurable_File_Limits
8. Behavior of the Generic I/O Operations With Non_Blocking Option
10. Make Fork and Exec optional (rejected by the committee as a change)
11. File Pointers on Read/Write
12. Access time update on Generic_Read and Generic_Write
13. Blocking vs. nonblocking behavior on Read/Write

Some of these will require amendments to 1003.5-1992. The plan for POSIX.5a is to fix the known errors and rewrite Chapter 6 (Read/Write). The initial P1003.5a document should be ready for ballot after the July meeting. POSIX.5a, when issued, will be change pages, which will be merged into 1003.5-1992 with POSIX.5b after its approval.

POSIX.5b is the Ada binding to POSIX.1b (the real-time extensions of POSIX) and Mutexes, Condition Variables and Thread I.D.s from POSIX.1c (the real-time threads extensions of POSIX).

The first formal ballot on what was then named POSIX.20 was conducted on a "thin" binding version: that is, POSIX.20 did not repeat the underlying semantics of the POSIX Real-time Extensions draft, which is a C-language interface. The ballot showed that a "thick" binding version was clearly favored instead, not requiring references to the C version. Time since January has been spent importing the underlying semantics into the draft. This "thickening" process has in turn exposed problems in the bindings draft. In the time period before the April meeting, some of these problems were worked out and the document was edited for consistency.

Most of the April meeting consisted of group reviews and changes to the thick version, now renamed POSIX.5b, resolving the exposed issues in sections 1-12. The agreed changes to the document will be made between meetings and a final group review will be conducted during the July meeting. The object is to have a new draft ready by August 1 for ballot, scheduled for the month of September.

The POSIX.5 working group, together with the POSIX.4 working group, is still working to resolve the seven objections to POSIX.1c that the POSIX.5 working group submitted in July at a coordination ballot. Five of the objections are considered resolved, although the POSIX.5 group has not yet had the opportunity to review the text changes scheduled for POSIX.1c Draft 9. The remaining two objections are currently under negotiation between the chairmen of the two working groups. The first involves situations where the code of a signal handler needs to ensure that a mutex is locked. The other involves the change brought on by POSIX.1c from per-process signal masking to per-task signal masking.

After a period of funding uncertainty, DISA has provided funding to Jim Leathrum's group at Clemson University's Software Standards and Technology Laboratory to develop the test assertions for 1003.5-1992. The POSIX.5 working group appointed Kathy Liburdy as Vice Chair of POSIX.5 for Test Assertions. The Clemson group plans to produce a draft prior to September when the DISA funding expires. They are also on track to produce a description of their method as an appendix to P2003.

The IEEE Computer Society Portable Application Standards Committee authorized the formation of a group to standardize an Ada binding to 1295-1993, Modular Toolkit Environment (the IEEE standard for MOTIF.) This work was assigned to the POSIX.5 group. Dave Emery plans to sponsor a study group within POSIX.5, starting with an organizational meeting during the July POSIX.5 meeting. The POSIX.5 group concurred with this plan on the understanding that this work should not detract from any current POSIX.5 efforts. The group will need a Vice Chair, Secretary, and additional people dedicated to developing and balloting the proposed standard. The work should take two to four years.

A Tour of the Distributed Service Working Groups

David Cannon <D.Cannon@Exter.ac.uk> reports on the April 18-22, 1994 meeting in Lake Tahoe:

This Spring Tahoe was warm, dry and attractive, with snow on the surrounding peaks reflected in the waters of the lake. But even this scenery failed to attract the POSIX crowds, with attendance overall down to about 145—well below that of the previous meeting.

The Distributed Services groups contributed to the shortfall, with POSIX.8 and POSIX.12 not meeting at all, due to the unavailability of target documents or conflicts with close-of-ballot dates.

The groups' progress over the week is outlined below:

POSIX.8 (Transparent File Access). Progress is stalled on two counts: it's awaiting the production of the document merging both the POSIX.1 and POSIX.4 standards, (*Ed. Note:* the merged POSIX.1b document is now with the IEEE for reproduction and distribution) which the POSIX.8 work will further modify, and the recirculation ballot of its draft. This latter didn't happen on schedule as some of the 'no' voters weren't contacted to confirm that their ballots had been resolved following the first round of ballot resolution.

The July meeting of POSIX.8 will work on the merge of their draft with the POSIX.1b document, which should by then be available, and a further ballot recirculation of the merged POSIX.1/4/8 (a.k.a. POSIX.1f) draft will take place following that.

POSIX.12 (Protocol Independent Interfaces).

The group will be meeting separately in the week beginning 23 May. This schedule locks into the completion of the group's recirculation ballot, due on the 2 May. The group will resynchronize with PASC (the IEEE Portable Applications Standards Committee) in July. It was noted that X/Open

are bringing their specifications in line with the sockets part of POSIX.12 via the X/Open fasttrack process. X/Open will track the changes introduced by the POSIX.12 ballot returns and introduce them to its work. The current POSIX.12 draft states a requirement for both sockets and XTI; this is echoed by the X/Open group.

POSIX.21 (Real Time Distributed Services). The first DS group working at Tahoe, which had up to twelve attendees through the week, and met for the full allotted time.

The group decided to pursue its proposed "Time Services" PAR as an addition to the POSIX.1... series of standards, rather than as an independent 13.xx standard. The group is happy with the overall progress made, given that they are already working on Language Independent text, though this particular meeting had some uncharacteristically slow patches where it revisited old ground. The group plans to have a first draft available in July 1995, and currently intends first ballot for July 1996. The current intention is that all the drafts produced by the group will take the "thick" form.

POSIX.22 (Security Framework Guide). The group met together with the available POSIX.6 people, the POSIX.6 draft being in ballot, (closing on 18 May) and had eight people in over the week. The current draft went out in the March POSIX mailings. The first day was spent reviewing the document from the viewpoint of the anticipated audiences. These fall into two groups; security aware and security unaware(!). This revelation determined the group to restructure the guide, and an executive overview to the draft was crafted during the week.

Steps towards ballot group formation will be made following the Tahoe meeting, with ballot projected to follow the July meeting if the schedule permits.

1238 (FTAM and OSI Services API). The group had three people attending. Their 1351 and 1353 (OSI API) drafts have completed ballot recirculation with 94% approval. By the Monday evening the group had resolved most of the remaining objections, one of which involved substantive changes to 1351—thus requiring a third recirculation ballot, scheduled for 5 July. The hope is to reach the September meeting of the IEEE Standards Board for approval.

1238.1 (FTAM) drafts were available at the meeting. The ballot window is set for July. Closing date for the formation of the ballot group is scheduled for 3 May, but the group is very keen to gain more members of the ballot group and its close will be delayed for as long as possible. The first ballot is targeted for 30 days from 6 June.

The supporters of the ROSE API proposal, notionally targeted for the 1238 group, have not followed up with a Project Authorization Request (PAR) or any other expression of interest. This may die unless some champion (X. Open perhaps?) comes forward to introduce their ROSE API to PASC, possibly through the IEEE fasttrack process.

Report on SRASS: Services for Reliable, Available, and Serviceable Systems

Arun Chandra <achandra@vnet.ibm.com> reports on the April 18-22, 1994 meeting in Lake Tahoe, Nevada:

Are you interested in Fault Tolerance, High Availability, Reliability, Serviceability, Maintainability? If so consider joining the "Fault Management Study Group" at the next POSIX meeting at Nashua, NH in July. The group approved a name change to "Services for Reliable, Available, and Serviceable Systems Study Group." If you see any reference to either of the above two names, its us.'

This group first met in October '93, following BOF sessions at the two previous meetings. The status of the group is a "Study Group" preparing a "Project Authorization Request" (PAR). The PAR will go up for review at the July meeting. If approved we will become an official POSIX working group. Healthy participation at the July meeting would indicate that Fault Management is something organizations are interested in seeing standardized. This is one of the basic criteria for PAR approval, especially in these hard times.

A number of existing documents are being studied as base documents. To obtain a list of these documents or the documents themselves please contact the chair of the group - Helmut Roth <hroth@relay.nswc.navy.mil>. The detailed minutes of the October '93, January '94, or April '94 meetings can also be obtained from Helmut.

The primary task of the working group at the April meeting was to get a PAR approved. A draft was submitted to the PAR Management Subcommittee (PMC), which was approved for recommendation for sponsorship. However, to get this approval, the study group felt that the scope of the standard had been overly reduced. As a result at the Sponsor Executive Committee (SEC) meeting the PAR was not sponsored and will be revised and resubmitted at the July meeting.

The group felt that the PAR's scope must include two major areas:

- Fault-tolerant issues, and
- Serviceability issues.

There was a debate as to whether there was too much initial concentration on the Serviceability issues. The Fault-Tolerance community representatives at the group, who are actually the majority, want to strike a good balance between the two areas. IBM's AIX documentation has been identified as the base document for serviceability issues, but no base document has been identified as yet for fault tolerant issues. All the above issues will be reflected in the revised PAR.

This study group spent an intensive week in the PAR approval and revision process. The group is optimistic that the PAR will be approved in July.

If you are interested in more information on the group, why not contact the group Chair, Helmut Roth, or me, the group Vice-Chair.

Unix Tricks and Traps

If your site is connected to the outside world through a plain ordinary UUCP or similar link, then you are probably using some form of ftpmail to acquire current source code for news readers, file transfer utilities, etc. Many ftpmail servers are now delivering requested items in sections encoded via the shar utility, which allows you to pipe each received part into /bin/sh for automatic decoding and (when all parts are present) re-assembly.

One difficulty is that the shar parts tend to appear when least expected. So you arrive at work one morning and log on to read the day's mail - and find a list of mail headers something like:

```
"/usr/mail/gkj": 10 messages
1 archie-errors@Archi Thu May 26 20:27 125/9340 archie [find kermit] Pt 1
2 bit-bucket@cbis.com Fri May 27 06:26 945/36953 INDEX (complete) ascii
3 ftpmail@connect.com Sun May 29 23:31 554/32813 libdes-3.01.tgz Pt 04/04
4 ftpmail@connect.com Mon May 30 07:08 1057/64104 libdes-3.01.tgz Pt 01/04
5 glad@daimi.aau.dk Mon May 30 23:02 30/1263 Committee Report
6 ftpmail@connect.com Mon May 30 23:32 1057/64104 libdes-3.01.tgz Pt 02/04
7 swoolley@cbis.com Wed Jun 1 06:01 29/1371 ** URGENT PRODUCT TEST **
8 joeblo@sequent.com Wed Jun 1 20:04 233/12972 New Widget Facility
9 cbisa.com.au!pburke Wed Jun 1 20:04 16/570 Needed Yesterday!!
10 ftpmail@connect.com Wed Jun 1 20:04 1057/64104 libdes-3.01.tgz Pt 03/04
```

The conventional way of handling such a situation is to back out of Mail, change to an appropriate directory, re-enter Mail, save each part of the ftpmail packages separately (and there may be lots more than 4!), then back out of Mail again and edit out the headers of each and every saved part before piping it to /bin/sh.

The script below will allow you instead to remain with Mail and enter something like

```
pi 3-4 6 10 "(cd /tmp; unshar)"
```

unshar will then pick the parts out of the single input stream fed to it, stripping out the headers, and feeding each part in turn to /bin/sh. And you can keep on with reading your incoming mail. unshar should execute on almost any Unix platform, provided the nawk utility is available. It relies on there being an exit statement at the end of each unshar part; readers may wish to extend it slightly so that each piece is checked for such a statement before it is piped to /bin/sh.

-oOo-

```
#!/bin/sh
# @(#) unsharProcesses multiple unshar files from an input stream
#          Graham Jenkins, CBIS Aust., May 1994.

if [ ! -t 0 ]; then                                # Create a temporary file to
  if [ $# -eq 0 ]; then                            # hold each Part in turn.
    tempfile=/tmp/unshar.$$; export tempfile
    trap '( [ -f $tempfile ] && rm -f $tempfile ); exit 1' 1 2 3 15
    nawk '{
      if ( $0 == "#!/bin/sh" || $0 == "#! /bin/sh" ) {
        if ( STARTED == "Y" ) {                  # Skip lines until #!/bin/sh
          system("/bin/sh < $tempfile")          # found, then copy to tempfile.
          system("cp /dev/null $tempfile")       # When next #!/bin/sh line
        }                                         # found, feed tempfile into
        STARTED = "Y"                            # /bin/sh, then clear tempfile,
      }                                          # write the line at its start,
      if ( STARTED == "Y" ) {                  # and copy further lines until
        print $0 >tempfile                       # the next #!/bin/sh line
      }                                         # found.
    }' tempfile=$tempfile
    /bin/sh < $tempfile                          # When end-of-file appears,
    rm -f $tempfile                              # feed tempfile in /bin/sh for
    exit 0                                       # the last time, then remove
  fi
fi
fi
```

```

cat <<EOF
Usage: cat file(s) | 'basename $0' .. performs 'unshar' operation on each
      'shar' part contained in file(s).

Note: 'basename $0' can be used at the '?' prompt whilst receiving mail.
      e.g: ? pi 4-7 9 "(cd /tmp; 'basename $0')"
EOF
exit 2

```

The following is a sister program, able to accomplish similar things for those who use the Princeton BITNET FTP Server (otherwise known as BITFTP).

-oOo-

```

#!/bin/sh
# @(#) unbitftp      Processes multiple bitftp files from an input stream
#                  Graham Jenkins, CBIS Aust., June 1994.

if [ ! -t 0 ]; then                                # Temporary file(s) will be
  if [ $# -eq 0 ]; then                             # created to hold Parts.
    tempname=/tmp/unbi$$; export tempname
    trap '( rm -f ${tempname}.* ); exit 1' 1 2 3 15
    nawk '{                                          # Phase 0.
      if ( $1 == "Subject:" && $NF == "(uencoded)" ) {
        tempfile = tempname"."$4
        print "Creating temporary file: ", tempfile
        phase = 1                                  # Phase 1 - "Subject: " found,
      }                                           # new temporary file opened.
      if ( $0 == "" ) { phase = phase + 1 }       # Phase 2 - 1st empty line.
      if ( phase > 2 && $0 != "" ) {              # Phase 3 - 2nd empty line;
        print $0 > tempfile                       # start writing to temp. file.
        phase = 4 }                               # Phase 4 - writing done.
      if ( phase > 3 && $0 == "" ) { phase = 0 }# Another empty line; revert
    }' tempname=${tempname}                       # to Phase 0.
    echo "Uudecoding and removing temporary file(s) .."
    cat ${tempname}.* | uudecode                   # Pipe to uudecode in Part-No
    rm -f ${tempname}.*                           # order.
    exit 0
  fi
fi

cat <<EOF
Usage: cat file(s) | 'basename $0' .. strips mail headers and trailers from
      'bitftp' parts contained in file(s), sorts the parts, and feeds them
      into 'uudecode'.

Note: 'basename $0' can be used at the '?' prompt whilst receiving mail.
      e.g: ? pi 4-7 9 "(cd /tmp; 'basename $0')"
EOF
exit 2

```

Graham Jenkins <gkj@cbisa.com.au>

Please send your contributions for this column to the Tricks & Traps / User Support Mailbox Sub-editor, Janet Jackson <jackson@cwr.uwa.edu.au>, (09) 380 2408.

Articles From the Australian Newspaper

The following articles are as submitted for publication in the AUUG column of the Australian. There might be some minor differences, from what appeared in the Australian as they may have been edited to fit the available space.

The Bandwidth Crisis

Michael Paddon

There's an old saying among programmers: "the amount of data you need to store will always expand to fill the available disk space". To a certain extent, this reflects the packrat nature of computer users, always unwilling to discard those ancient mail files that noone will ever read again. Even allowing for such foibles, however, it is quite obvious that we are storing more and more important information on our computer systems.

Contemporary disks make this all possible by delivering two orders of magnitude more storage at the same price, compared to ten years ago. Not only has the price per byte decreased, but the density of storage has made multiple gigabyte drives a reality.

Similarly, CPU's have benefited from parallel trends. A modern RISC chipset can deliver speeds in excess of a hundred million instructions per second. This is a sobering thought when you compare today's desktop RISC PC against the room sized VAX of the late 1970's. The VAX was ground breaking technology for its time, grinding along at a mere one million instructions per second.

These two successes of technology have made possible the personal computing revolution. In fact, it is so easy to be dazzled by these spectacular leaps and bounds, that it is possible to forget the other key component of the modern computer system: networking.

Unlike disk and CPU technology, networking has languished in time warp of arrested development. Ethernet and other LAN standards (token ring, token bus, etc.) were pretty much nailed down by the start of the 1980's, and were available off-the-shelf immediately afterwards.

Back then, LAN technologies delivered a bandwidth of around 1 Mbps (megabits per second). Today, ethernet still dominates the LAN market and still provides the same bandwidth. In the meantime, our systems are making ever greater demands for communications. Network file systems, distributed databases and multi

media applications are but a few of the bandwidth hungry uses that have been developed since ethernet was invented.

It doesn't take much analysis to foresee that a crisis is approaching. Network bottlenecks are already the determining factor for a system's performance, particularly with the emphasis in recent years on the client-server model of application design. While most extant bottlenecks are occurring in LANS, before long we will be experiencing exactly the same problems with long haul networks.

As a system user, I want to be able to connect my computer into a network anywhere in the country; at work, at home, and even at the hotel when I'm away. I still want instant access to my files and documents. If I'm using images or sound samples (as I increasingly do) I want to be able to manipulate them quickly and easily and perhaps copy them to friends or associates. And if I'm dealing with images, then why not video?

The sad fact is that even LAN's cannot provide this sort of performance, let alone WAN's. A photo-quality colour image can be as much as 48 megabytes (at 4096 x 4096 resolution). That's going to take close to a minute to transfer over ethernet. Over an ISDN link the transmission time is getting towards two hours.

If you don't think that you'll be using images a lot, then consider this. A CD's worth of sound samples consumes much more data than the photograph. Even a TV frame takes about a second to transfer over ethernet, or three minutes over ISDN.

These problems haven't escaped the notice of networking vendors. There are currently two separate proposals for 100 Mbps LAN technologies in front of the various standards committees. Both of these are based on electrical cabling, with one capable of utilizing existing ethernet. Another technology, ATM (asynchronous transfer mode), is based on optical cabling and can provide a bandwidth of 130 Mbps.

While a tenfold increase in bandwidth is welcome, it will come at great expense and the performance increase simply does not compare

with that experienced by disks and CPU's over the last decade. There is no reason to expect that these latter technologies will not continue in their exponential growth, at least for the medium term.

The new networking technologies will buy us some breathing space, but even if we upgraded all our LAN's and WAN's to use ATM, the fastest option available, they would not even begin to deliver on the promise of the multimedia applications being built today.

Business is War!

Roger Fraumann

In the 1980s, the American military discovered its biggest problem was that it had let technology drive strategy. An excellent book on the subject is Alvin and Heidi Toffler's latest; "War and Anti-War." In business, technology has driven strategy as well. The evidence abounds, with examples of the lack of standards, obsolete equipment, and methods. By applying lateral thinking, we can take the simple lessons learned by the American military and correlate them to business.

Starting in 1990, the U.S. Joint Chiefs of Staff spent 18 months and considered 17 scenarios projecting the art of war in the year 2025. The resulting "2025 Study" is an "unfinished" 4,000 page DoD working paper, touching in part, on what advancing technology they will need to meet their changing mission requirements. They found that they could not, with confidence, predict exactly what technologies they needed.

After all their efforts, they arrived at the realisation that the best planning strategy required using time proven basics. They re-discovered the criticality of the "decision cycle" as an essential strategy on which to base organisational and technology decisions.

On the battlefield, the sides who could acquire information about the situation, decide what to do, and execute a plan of action faster than their opponent would stand a better chance of prevailing. It isn't as important whether they are fighting with rocks or missiles, or for that matter, fighting a military battle, war of propaganda, or an economic war. This was a simple observation, but essential. Simply by concentrating on the "decision cycle," they have

begun to restructure -- from their procurement process, to questioning their existing division of Services.

We can apply the "business decision cycle" as an enduring constant in business as well. We have been doing this piece-meal already. As we transition from a production to an information based economy, we have also been shifting from a space towards a time oriented business policy. An example is the "Just In Time" inventory management practice.

What we have not been doing enough of, and it shows in our historic use of computer technology, is to apply the question "How can we apply this technology to improve our competitive decision cycle: our ability to acquire, decide and deliver faster and better than our competition." Information technology investment decisions, when held up to this strategy, take on a new light. We can quickly see the difference between information management as an asset, and information technology management as a liability.

The American military is concentrating on improving their ability to acquire information, their ability to command and control, and the use of smart weapons that can tie into the network. One observation they have made is the resulting "rock syndrome." In conflict, the human element is rapidly becoming incapable to actively participate in the decision process to defend themselves.

Computer controlled offensive weapons, once committed, require the defender to be able to acquire, target and destroy the offensive weapon while it is on the way. This requires the defender to use more powerful computers as part of the process. There is no time in the decision cycle for real-time participation by people, hence the defensive decisions need to be pre-determined and programmed. People will be perceived as participating like rocks during the event, because it will happen all so quickly.

Electronic trading and arbitration are examples of the growing dependency on computers in the competitive business decision cycle. We are seeing layers of middle management who once provided acquisition, filtering, analysis and control systems being made redundant. At the same time, companies are needing people to build competitive advantage through the application of creative and perception skills.

As a result, new creative competitive information acquisition techniques are emerging. "Open sourcing," which is the locating, obtaining and correlation of public and commercially available information, is being used to improve the information acquisition process. An example is searching across the cyberspace of the Internet using World Wide Web into hundreds of databases. Creative acquisition and decision support are rapidly becoming a business imperative.

The American military re-discovered a "constant" -- their need for a strategy to dictate the technology directions necessary to help them in their decision cycle. Business can learn from their example by focusing IT efforts through a decision cycle strategy, instead of continuing to let technology dictate the rules.

Will we see an "Open NT Foundation"?

Phil McCrea

It is interesting to note the number of hardware manufacturers who have announced plans to make NT available on their platforms. Digital of course has been the most vocal supporter of NT, for reasons which are not clear. Possibly the connection with Dave Cutler, the ex Digital employee, who is the chief architect of NT?

Intel based machines of course are ripe targets for NT, and companies like NCR and Unisys, whose machines are based on Intel technology, stand to benefit from the fact that Intel is the principal NT development platform.

MIPS is the next most popular platform after Intel, a legacy of the ill-fated ACE Initiative. It's a pity ACE did not work. Its aims were entirely noble, and in the best interests of open Systems - ie freeing up the dependence of both application software and operating systems from the underlying hardware. The two operating environments in ACE were UNIX (from SCO) and NT.

Recently several other manufacturers have made announcements to make NT available on their platforms. IBM, HP and Data General have all had some press announcement to this effect. These companies, and virtually all other hardware manufacturers, have all identified UNIX as their strategic operating system. They had no choice - users asked for it. Big users

such as the US Government demanded it! Making NT available as well is called "hedging your bets"...

The fact that NT is being made available on a number of platforms raises an interesting question - how compatible will the different versions of NT be? This has been one of the problems - or should I say "features" - of UNIX. You see, UNIX was originally given away free to Universities in the mid to late 70s, the reason being that AT&T, the owner, is a telephone company (like Telecom), and at that time, prior to a major change in US Government anti-monopoly policy, was not able to sell software. So hundreds of Universities round the world - the most notable being right here in Australia - started to add new features to the PDP11 version of UNIX that they received on their tape from AT&T. The result: lots of UNIXes, with resultant portability problems. But, on the plus side, a set of exceptionally rich features - which makes UNIX what it is today.

AT&T made a gallant effort to unify the main strains of UNIX by introducing SVR4 - but in the process alienated several major players, most notably IBM, HP and DEC. There were other factors, but the net result was even more confusion in the UNIX market place, and the birth of the "our UNIX is more open than yours" wars.

Now... if NT is coming out on platforms from DEC, Unisys, IBM, HP, DG... and all these companies have NT source code, there is the temptation for each of these organizations to provide their own little bit of value-add, to provide that "little bit" of vendor lock-in, whilst at the same time being 100% NT compatible! You only have to look at the database vendors - they are all SQL compatible...

So, will we ever see an organization trying to unify NT, such as an Open NT Foundation, or an NT International? Perhaps not in name, but you can bet there's a fair bit of activity behind the scenes in this area.

System Administration: The Emerging Profession

Brenda Parsons

The demise of the one-vendor businesses, the dramatic increase in the number of distributed networks and the trend of down sizing, has led

to the rapid emergence of the profession of System Administration.

System administrators are generalists who must be proficient at installing new software and hardware, fixing bugs, training users, repairing cables, automating mundane tasks, networking, databases and budgets. They must be a jack of all trades, be capable of multi-tasking and they must be able to achieve all of the above on multiple platforms ranging from the desktop PC running DOS to a myriad of hardware vendors machines running a plethora of Open Systems Operating Systems.

The role of the System Administrator is frequently misunderstood by management. System administration draws on knowledge from many fields, and as such, administrators come from a wide range of academic backgrounds. Only recently, has there been courses offered on System Administration as part of a tertiary degree and no one ever grows up wanting to be a System Administrator.

Most administrators obtain their skills working with a more experienced system administrators or by being thrown in the deep end. Nothing can equal the petrifying experience of being alone with a machine that won't boot.

This method produces highly skilled administrators but is misunderstood by employers who tend to focus on other credentials, as is happening in the Novell arena with the CNE accreditation programs. To quote Wendy Nather of Swiss Bank, "The best system administrators are forged, not made".

Employers repeatedly confuse system administrators with operators, since they perform many of the operator functions in the larger or mainframe environments. They are also deemed as under productive when measured against the classification of programmer, where effectiveness is measured in the number of lines of code produced. System administrators generate code as a means to automate or simplify tasks rather than to mass produce programs.

In 1992, the System Administrators Guild (SAGE) was formed in the United States. The charter of SAGE is to promote the profession of System Administration and to this end working groups such as Jobs and Education have been formed in an effort to combat the downtrodden image of the System Administrator.

An Australian SAGE organisation (SAGE-AU) has also been formed and already has local chapters in most states.

The Jobs Working Group has produced a job description booklet which contains a template for employers that details the skills and responsibilities for each of four levels of System Administrators being Novice, Junior, Intermediate and Senior. In addition to the basic descriptions, a checklist of items which can be used to augment the core job descriptions has been generated for the use of employers. The checklists are divided into the following categories: local environment, heterogeneity, programming skills, networking skills, security, site specialities, documentation, databases, hardware and management skills.

The stereotypical image of a system administrator by management seems to be predominantly dependent on gender. The operator or non-technical image tends to be assigned to the female administrators, whereas the guru image is assigned to the male administrators. Neither depiction is acceptable in the light of the profession as many of the worlds most prominent system administrators are women. Perhaps this means that women are better suited to multi-tasking and problem solving, or more then likely, it stems from perception of the job itself, that of being a clerical role, rather than the technical highly skilled profession that it really is.

System administrators are typically overworked and underpaid, simply because the majority of work they do is transparent to the users and management. It's like the oil pump in your car engine, you never appreciate it until it's not working. The industry recommendation is one System Administrator for every 100 users, but very few sites come close to this ratio. Most System Administrators are also on 24 hour call, 7 days a week. Much of the technical work must be done out-of-hours, but they are still expected to work normal business hours for the support of the users. The most productive time of day for a System Administrator is that small window of time before and after normal business hours where they can catch up on all the little things, but again, many employers fail to understand this.

So give your system administrators the recognition and respect they deserve for a hard job well done.

CASE Tools

Margaret Hassall

UNIX developers are not quite the cobbler's children that they are often cast as: they have been raised in an automated environment built from many different tools, mostly single function programs that combine to create a useful and productive system. As software development matures, however, so must our tools.

There are many products on the market that call themselves CASE tools. Computer-Aided Software Engineering provides tools to support all phases of software development: requirements analysis, design, coding, implementation, testing, and maintenance. Software engineering also includes support for project management, reviews, documentation, verification, and validation at all phases of the process.

There are many tools available that support some of these activities. UNIX developers have been using automated tools for a long time to help their coding, including source code and versioning control systems, debuggers, profilers, and make files. But coding is only one part of the software life cycle. The CASE workshop must also include tools that support the other phases. For example, there are graphical design tools that allow developers to create a model of the new system, including control flow, data structures, and algorithm design. This sort of tool can generate structure charts and data dictionaries; they can sometimes provide different views of the design information such as a graphical view of the process inter-relationships, or show the pseudo-code for a process.

Good engineering workshops provide a collection of useful tools and an organised space that allows access to the tools so they can be used efficiently. Software engineering environments also need this. To work efficiently, CASE technology workshops must provide more than a collection of tools, they must also provide an integrated environment.

At the heart of CASE technology is a central repository providing access to information. Information about each phase should be stored and tools provided to access it in a variety of ways. Design diagrams can be analysed for consistency, requirements can be traced, milestones can be checked and so on. The

repository stores information about the functions, processes, interfaces, data, and the system relationships, allowing users to use and view data as they need to. Without this, the CASE system is no more than a collection of tools.

However, a lot of CASE tools are still used separately. Graphical development systems provide quicker and easier ways of drawing data model diagrams. But is this providing better support than our pencil drawings of old did? One criticism of these tools is that we just spend a lot more time doing exactly what we did before. A problem with software engineering in general is that a lot of documentation is produced at each phase, only to be filed or left incomplete as it takes up too much valuable time that could be used coding!

New technology is not much use if it is not used correctly, or if we assume it will do everything! CASE tools will not do away with the need for good management. They may provide information, but the manager must know how to use it. A requirements tool may provide for data flow diagrams and a data dictionary. This information may be used as the starting point for the design phase, but does the tool also provide analysis of the system? Is there a traceability button that checks that all requirements are dealt with (nothing left out), that only the requirements are dealt with (nothing added in), and that the requirements and the design are consistent? This information may be stored in the repository, but there must be a way of extracting this information, and in a form understandable by humans.

CASE technology can provide us with the ability to more easily model and analyse our systems before we build them, but we still need to manage the process - the technology will not do all the work for us. We might be able to generate diagrams very quickly with the new whiz-bang tools, but why spend the time and money if all we do is file them, just like we did with the pencil drawings?

A good CASE system must provide a computerised workshop for the software engineer. There must be a collection of tools to suit all aspects of the software development process. These tools, however, must be integrated to provide easy and useful access to the development information. Only then can CASE technology take the UNIX philosophy of a suite of small, functional tools a step further to

improve our development environment.

Distributed Computing for the Masses

Michael Werner

The ability to perform tasks using multiple computers has been available for some time. In fact, one of the reasons why networking code was added to 4.2BSD was to allow the US defense to take advantage of distributed computing. So in fact distributed computing using current software could have been developed ten years ago. But it is only in recent years that multiple UNIX workstations have become affordable for groups small enough that sharing resources is politically viable.

An example of public domain software for distributed computing is PVM (Parallel Virtual Machine). It uses the defacto standard UDP/IP and TCP/IP protocols for message passing on a heterogeneous network. Use of these standard protocols allows machines from multiple vendors to be utilised in a transparent way. It is also possible to use multiprocessor and massively parallel machines. This allows an algorithm to be developed on relatively inexpensive workstations and then moved to a target massively parallel machine where CPU time may be expensive. Complete portability has not been achieved between workstation clusters and massively parallel machines. Hence, users have to be weary of the limitations when developing their algorithms.

Whether distributed computing can be utilised for a particular problem depends on the type of algorithm and its resource requirements. One important limitation is network bandwidth. Typical environments have 10 Mbps Ethernet LAN's which are already used for network file systems and X windowing systems. Data latency can be affected by packet collisions, kernel buffer limitations and overall system loads. Since synchronisation is an important property of parallel algorithms, blocking I/O can significantly affect overall performance. Other networking technology such as FDDI is available which provides roughly 10 times the bandwidth. An example of a problem which is suited to loosely coupled workstations is monte carlo simulation of initial value problems.

The flexibility of a workstation cluster allows the user to define their own topology for the nodes.

This is implemented by inter-node communication defining a particular type of connected graph. A cluster can simulate many different topologies which may be hardware enforced in a massively parallel machine. For example, different routing mechanisms and speeds for inter-node communication may be presented by the hardware of a massively parallel machine. This provides a grouping of nodes much like workstations on a subnet. In both cases, algorithms may take this grouping into account or simply ignore it and suffer the performance costs.

This same flexibility allows one to perform subtasks on different but more appropriate hardware. For example, one can input data on a workstation which is processed in part by a massively parallel machine and then by a vector machine. The 3-D graphical output then rendered on the workstation. However, if the parallel machine were heavily loaded one can simply use a workstation cluster for the part performed earlier by the massively parallel machine and complete the task as before.

The process of generating the parallel code is typically done by the programmer today. Just as with vector optimisation compilers in the past, current efforts are to produce parallel compilers which utilise existing hardware well. Software of this type exists for massively parallel machines as one would expect. Tools for generating parallel code for workstation clusters is only recently become available by third party software vendors. Parallel compilers are still an area of active research. Once again portability is a major issue here. If generic packages such as PVM were to play the role of intermediate code for message passing calls then some degree of portability would be provided. Such compilers would broaden the spectrum of users able to take advantage of parallelism.

Passing On C++

Paul-Michael Agapow

It's usually a sign of age when, as your peers leap on the latest bandwagon, you straggle behind plaintively asking "But why?". Recently some of our more senior computer professionals must be feeling their age as the market rushes to embrace C++. As once C was heralded, so its object-oriented offspring is greeted as The Answer. Programmers will be more productive.

Code will be reusable, portable, less buggy.

But (as one journal asked) if C++ is the answer, what was the question? Its syntax has bloated, new users are bewildered by its baroque constructs and portability awaits the creation of a good standard. It is not a pure object model, nor does it necessarily promote a tidy, object-oriented style of programming. (Some might call this an advantage.) C++ is far from useless, but it is not perfect. Nor should it be - after only 50 years of computer languages it would be absurd to think the pinnacle had been reached. So what other choices are available?

The simplest of all paths is one a number of developers have chosen : a gentle step backwards to design a better object-based C. Examples include C extensions packaged as development systems like Liana (Base Technology) and Think Class Library and object extensions (Symantec), stripped-down or redesigned C++ dialects like Objective-C (famous as the development language of the NeXT but popular on a variety of platforms) and dialects based on alternative object paradigms like C+@ (AT&T Bell Labs).

There's a lot to be said for this conservative approach. Evolutionary (rather than revolutionary) development is a safer investment. The lessons of the past are more directly applied, less time is spent retraining and recoding, you are only supplied with the features you need. Some of these features are very snappy indeed - C+@ produces portable binaries, several systems incorporate garbage collection and the development system languages make GUI design easier.

Conversely, portability is still a problem in that most C dialects have yet to achieve widespread use. Also, it may be redundant to forsake the complexity of C++ for a stripped down version, if just using a subset of C++ will suffice. (In the long term some of the "superfluous" features may be useful). Finally, regardless of what variant used, you are still bounded by the worldview, the style inherent to C. This style dictates the problems that C is manifestly unsuited for (e.g. high-level symbolic manipulation) as well as those for which it is perfect. The only long-term answer to this problem is to find another language, one suitable for the desired purpose.

While Mathematica (Wolfram Research) has been around for a number of years it has still to

effectively break out of the realms of academe. This is a pity because behind its guise as a "symbolic language for mathematics" lurks incredible power. Syntactically it resembles an interpreted Pascal on steroids, with a huge number of mathematical or symbolic functions built-in and in libraries. It can manipulate graphics and sound, produce interactive documents, act as a process control language ... in fact, it may be too powerful for casual users.

As any good modern system should, Mathematica outputs in a variety of formats (Postscript, PICT, TeX) and its notebooks (documents) are fully transportable across platforms. Thus far Mathematica has been largely heralded as a powerful enabling and visualising tool for mathematicians. Admittedly, it is an unlikely choice for building the next great word-processor or operating system. However it is perfect for those data-modeling and statistical problems that may confront the programmer outside the mathematical arena, in geology, biology, economics etc. But potential fans should be warned that like many new languages it is demanding of CPU time and memory. Coupled with its complexity, investment in Mathematica could be very expensive in hardware and time.

In many ways language designers are caught having to design today for the needs of tomorrow. Developers and managers are likewise trapped, having to bet what will be the best development platform for the future. It's instructive thus to think what a future language might be. One possibility, proposed by Apples Advanced Technology Group, is Dylan (DYnamic LANguage).

In modern development environment, software is getting larger and more complex while development tools have barely changed. Time and investment to market is increasing, while programmers are still required to program at the machine level, hindering portability. Dylan is a "clean sheet of paper" design to meet these problems. As such it ends up a strange-looking critter with impressive claims and perhaps the ability to fulfill them.

Dylan is an OODL (Object-Orientated Dynamic Language), a Scheme (simple LISP) variant that is "objects all the way down". It has a small syntax, no pointers, no need or encouragement to fiddle at the bit level, and memory management is automated. Because Dylan is dynamic not

static (program and object information are read at runtime, not compile-time) code is interpreted and modifiable without recompilation. Programs may be built piece by piece and functions tested in isolation. Error catching and high level object-oriented debugging is inherent.

Dylan is ambitious and while it may not see the market, it seems inevitable one of its descendants will. OODLs are renowned for slow performance and being memory hogs. While Apple asserts that given current hardware trends and an appropriate language this need not be so, the reality has yet to be seen. Perhaps the biggest problem Dylan has is its dependence on a LISP-like syntax, which is sure to be unpopular with many. In anticipation Apple have announced plans to produce an alternative, Algolic syntax for Dylan.

Brave New Technologies

Paul-Michael Agapow

Amongst the barrage of new arrivals on local bookshelves, some strange titles have begun to creep in : "Complexity", "Artificial Life", "Evolutionary Programming". Even in the buzzword-laden computer field, these new arrivals carry their own peculiar jargon and attitude. Some refer to programs as organisms, to growing solutions or to the creation of programs that work in unknowable ways. The sciences of complexity are slowly muscling into the world of computers.

The world of complex systems is (appropriately) a turbulent one, which winds its way from fractals through cellular automata to robotics. Rather than any unifying theory, it has but two central philosophies, "Simple rules can build and do complicated things" and "Complicated things act in interesting, non-random ways".

Neural networks are a classic example. These nets consist of nodes linked in a mesh of connections. If the weighted sum of activated neighbours of a node exceeds a threshold then that node will itself activate. This leads to a cascade of activations and deactivations throughout the network. More importantly, when set in a training mode the network may change the weights used so as to match an input with a desired output.

This simplicity belies the power within. On the

most basic level neural networks can classify, match patterns they were trained on to desired outputs. More to the point, they can take new or noisy inputs and classify them. In fact it's been shown that a net of sufficient complexity can map any linear or non-linear function.

The potential for handwriting recognition, computer vision and other fields is obvious. Note that the builder of the net does not have to know how to do the task, as the network learns itself. Indeed for complex nets, it may not be possible to know how they work.

Thus we have three of main principles of complex systems : 1. a complex system may modify itself, self-organise, to produce desired behaviour, 2. while a programmer may coax a system towards desired behaviour, the fine details of the system are not controllable or necessarily understandable, 3. behaviour is the result of the entire system, not any single part of it.

By now neural networks are almost in danger of becoming old hat. Genetic programs are of more recent origin. Although coming in several flavours (genetic algorithms, genetic or evolutionary programming, evolutionary strategies) with some fundamental differences at the programming level, in overall terms the effect is identical. All aim to find solutions to problems with an unwieldy search space, where there are too many choices and no way to logically find an optimal answer.

A population of possible solutions is created, each with its own random "chromosome" of parameters or instructions. Each solution is tested and rated on how well it does. "Fit" solutions (those that do well) are allowed to reproduce, perhaps breeding with other solutions blending their chromosomes. Some may be mutated. "Unfit" solutions are disposed of and the cycle repeats. In time the population becomes better and better at solving the target problem.

All the evolutionary techniques however have a distinct failing in that their problem must be well-defined and solutions easily testable. Having said that, genetic programs are able to solve many traditionally tricky problems with ease, avoiding blind alleys and homing in on the solution. One set of these breeding programs have been evolved to sort, arranging series of numbers with efficient and almost mystical ways.

Neural networks and genetic programs derive from nature, from the nervous system and evolution respectively. It was somewhat inevitable therefore that this motif of inspiration from the biological world would become a fulltime preoccupation in itself, as it did with Artificial Life. A-life is also unified largely by a single philosophy : biology works with large groups of objects and simple rules interacting on a local basis. Computational systems that fulfill these requirements can emulate the flexibility and power of living things.

This manifesto is embodied in the rapid prowling robots of MIT's Mobile Robot (Insect) Lab. Their small and almost stupid robots walk and navigate by a simple set of loosely linked protocols with no central command. In doing so they are able to avoid obstacles, go up and down inclines and outperform more expensive and cumbersome machines that have difficulty finding their way across a room.

Along different lines, the aforementioned "sorters" were bred alongside with competing parasites that attempted to make more devious sorting problems. Surprisingly, the sorters not only evolved faster, they produced better solutions. Not only was it useful to grow your solutions, but to grow your problems as well. This titbit alone hints at the secrets that A-life may yet divulge.

It's early days yet, before the impact of these new technologies may be felt. Other "revolutionary" technologies have fallen by the way, as complex systems may yet do. But it's appealing to think of a world where you can grow programs and truthfully say that you don't know how they work, they just do.

Take Unix Home to Meet the Family

Michael Paddon

Regular readers of this column will recall the recent "Orange DOS versus Durian Unix" piece, which sarcastically poked fun at the shortcomings of the Microsoft product. Interestingly enough, this article provoked the greatest reaction from the readership that this column has yet seen, proving once again that operating systems are a subject dear to everyone's heart.

So if DOS is so easy to criticize, why does it

dominate the PC market so completely? The two most obvious reasons are low cost and lack of choice. Most PC's are delivered by the vendor with DOS already loaded, and with the cost bundled into the package.

With the advent of both commercial and free PC-based Unix release, the dynamics of this equation are rapidly changing. It's now possible to run a true multi-processing, multi-user open system on industry standard hardware, taking advantage of the low prices that the commodity market has to offer.

Unix is now within the reach of the average home user and, since most versions support the emulation of a DOS environment, it comes without the cost of abandoning your investment in DOS software. Indeed, even Windows 3.1 emulators are becoming more common, making the transition merely a matter of sorrowfully saying goodbye to a quirky and unreliable environment.

The focus of this week's column is the sort of PC you'll need to buy in order to run a modern Unix effectively at home. Let's define our baseline system as a 386 based PC, with at least 4 megabytes of memory and around 40 meg of disk.

Why not a 286? Because the 386 chip was the first of its series to support the minimum memory management functionality required by Unix. A floating point coprocessor isn't necessary, but it will improve performance noticeably.

Of course, nowadays, you'd be silly to buy anything but a 486 based machine. Try and avoid the 486SX chips; they generally perform worse than a fast 386. Other than that, you'll have to decide on the best cost/performance tradeoff for your needs; the two main choices are a 486DX/33 Mhz and a 486DX2/66 Mhz. Or a Pentium for performance that approaches the previous generation of RISC chips.

If you are not running a windowing system, 4 megabytes will suffice for general use, however, a minimum of 8 megabytes and perhaps even 16 megabytes is recommended if you intend to run X windows. Of course, the advantage of X windows is a much more friendly and usable environment.

Similarly, a basic but functionally complete Unix

system (with no X windows or other optional software) can be squeezed into 40 megabytes of disk space, with some room left over for personal files. If you want the works, including the windowing system, you'll need more in the order of 100 megabytes of disk for the system. Since most hard disks sold today are 250 megabytes or larger in size, even a fully installed system leaves ample room for work space.

You'll also want to use a video card that can support 1024x768 (or even 1280x1024) non-interlaced, and a decent monitor to look at (how much is your vision worth to you?)

Another useful purchase is a modem, which can give you access to the globe spanning resources of the internet: gigabytes of free programs, images, sound files, public domain books, information files, USENET discussion groups, real time "chat" forums, online games and a lightning fast electronic mail service. Although you can still buy 2400 baud modems, they are not much cheaper than V.32 (9600 baud) models. Of course the real net runners will find v.32bis (14400 baud) modems even more useful.

Although a system meeting these requirements will satisfy many a budding Unix guru, a few more issues (and expenses) need to be addressed for serious performance.

The biggest bottleneck with PC's today is their system bus. Neither ISA (the standard PC bus) or VESA local bus is ideally suited to the DMA requirements (Direct Memory Access; where peripherals can read and write system memory without CPU intervention) of a multi-processing system.

To maximise system throughput, the ideal architecture is an EISA bus (or the new PCI bus being sold with Pentium systems), with a SCSI adaptor card and SCSI disks (SCSI, Small Computer Systems Interface is an alternative to the more common IDE). All this comes at a cost, however.

The bottom line is that you can set yourself up with Unix capable hardware for a cost of between \$1500 and \$8000 depending on the demands you will make of your system. Of course, if you can press that existing PC into service, the cost of experimenting with Unix is far less.

The next step is to choose the appropriate Unix

software for your PC; an aspect I'll examine in a future column.

AUUG MANAGEMENT COMMITTEE

SUMMARY OF MINUTES OF MEETING 18th July 1994

Location: AUUG Business Office, North Sydney.

Present: Phil McCrea, Glenn Huxtable, Frank Crawford, Chris Matby, Michael Paddon, Lucy Chubb, Rick Stevenson, Stephen Boucher, Peter Wishart.

Apologies: None.

1. Secretariat Report

1.1. AUUG94

80 people responded to early bird registration. This is better than last year.

Budget expenditure is currently on track.

1.2. Other Secretariat Items

WA chapter has requested that chapter secretary's be able to request updates of membership information for their chapter through the Secretariat. Committee agreed. Chapters to be informed.

2. President's Report

PM welcomed Lucy Chubb to the committee. PM to write to outgoing committee and officers (Greg Birnie and Michael Tuke) thanking them for their efforts.

Last year's highlights were: AUUG93, official formation of chapters (5), Kirk McKusik tour, articles in Australian (good PR), and acquiring new business manager and office. They were successful because they were nominated as projects and managed as such. Should consider this as a model for future committee work (covered later in agenda). Should look at another tour (like Kirk), ways to capitalise on networking.

This year should be one of consolidation, last year was for expansion. Need to look at reducing costs or expanding income. Increase services or exposure to corporate world.

3. Treasurers Report

Since the last meeting we have had a cash flow problem, but this is now solved. Moved \$25K from cash mgt to cheque account. Now have \$25K in cash mgt account and \$54K in cheque account. Net assets \$129K (mostly cash at bank). Money is still owed to AUUG from AARNet payments and memberships. statements.

Stephen Boucher was appointed assistant Treasurer.

In summary financial position is about the same as last year.

4. Secretarys Report

Secretary getting some assistance from NSW chapter in reviewing and enhancing membership processing. Some members were sent another renewal letter with their new membership card, instead of a thank you letter. This was a human error at the Secretariat. Wael has sent a letter of apology to affected members.

Membership cards - looking at options for improving the membership card. Should consider credit card

type. Needs to include new logo.

Chapter Council Meeting - To be held at AUUG94. GH to be the committee delegate to the chapter council and coordinate chapter attendance.

5. Corporate Sponsors

Now had 3 corporate Sponsors. IBM, Sun and Digital. Other candidates being pursued.

6. Membership Survey Report

400 replies. 50% collated so far. Averages so far:

- been a member less than a year
- 25-34 years old
- mostly 4 year degree
- majority 1-5 years UNIX
- primarily systems admin (21%)
- business/software dev (15%)
- govt/military (14%)
- UNIX - Sun (20%)
- DOS (86%)
- 78.5% have e-mail access
- 60+% think AUUG meets needs
- 48% think chapters meet needs
- 35% think chapters do not meet needs

7. AUUG96

There have been nine submissions. Committee scanned submissions.

8. 1994/95 Projects

It was agreed to have the following as projects for 1994/95:

Internet Services for Members - FC (Chair), MP, CM.

Chapters - GH (Chair), PW. Summer conferences. Organisation of a major event (ala Kirk).

AUUG Tutorials/Seminars - PM (Chair), GH, RS. Oriented to commercial people. Could involve someone like IIR. Differentiate from membership services. Generate a list of AUUG Members willing to make presentations.

Membership Services - PW (chair), GH, RS. Dealing with Secretariat. Membership lists and databases. Discounts. Membership cards. Application forms.

Australian Articles - LC (Chair), MP.

9. Other Business

9.1. Australian Articles

MP reports that 90% of articles submitted have been printed. Some articles may have been lost (problems at Australian). Need to increase use of Australian in lead-up to AUUG94. Some articles

from speakers at AUUG94 could be used in Australian.

Thanks to FC and PM for always supplying articles. Activity involves lots of continuing effort. It would be preferable to have a rotating 6 month schedule for the coordinator. Subcommittee (project) formed to coordinate (as above).

The call for articles is now in AUUGN and the articles are being published in AUUGN. Other papers have been approaching AUUG offering fees for articles. We have declined due to other commitments. Need to consider in future.

9.2. Misc.

LC is appointed assistant secretary and will take minutes when the secretary is not present.

The contact details list being developed by PW should be added to the FAQ.

10. Next Meeting

The next meeting will be on the evening of Mon 5th Sept 1994 in conjunction with AUUG94 in Melbourne (exact venue and time TBA). Agenda will be mainly AGM and conference related issues.

The meeting closed at 5pm.

Peter Wishart
AUUG Inc - Secretary

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 your membership type, examine your membership card or the mailing label of this AUUGN. Both of these contain information about your current membership status. The first letter is your membership type code, M for regular members, S for students, and I for institutions, or R for newsletter subscription. Membership falls due in January or July, as appropriate. You will be invoiced prior to the expiry of your membership.

Check that your membership isn't about to expire and always keep your address up-to-date. 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.

MEMBERSHIP APPLICATION

INDIVIDUAL



To apply for AUUG membership, complete this form and return it with payment in Australian Dollars to:

**REPLY PAID 66, AUUG MEMBERSHIP SECRETARY,
P.O. BOX 366, KENSINGTON, NSW 2033, AUSTRALIA
Tel: +61 2 361-5994 Fax: +61 2 332-4066**

Tick this box if you wish your name withheld from mailing lists made available to vendors.

NOTE: Please do not 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, and remember to select either surface or air mail.

INDIVIDUAL OR STUDENT MEMBERSHIP:

I, _____ do hereby apply for:

- Renewal/New membership of AUUG \$ 90.00
- Renewal/New Student membership \$ 25.00 (please complete certification portion)
- International surface mail \$ 20.00
- International air mail \$ 60.00
- UniForum affiliate membership \$ 20.00

I agree that this membership will be subject to the rules and by-laws of AUUG as in force from time to time, and that this membership will run from time of joining/renewal until the end of the calendar or financial year.

Signature

Date

TOTAL REMITTED: AUD\$ _____ (Cheque, or money order, or credit card)

LOCAL CHAPTER DESIGNATE:

You can participate in the activities of a local AUUG Chapter. Part of your fee will be given to the chapter to support those activities. By default a regional chapter will be selected for you. If you would rather nominate a chapter, please specify here _____ (indicate NONE for no chapter).

TO BETTER SERVE YOU, PLEASE PRINT YOUR CONTACT INFORMATION:

Name/Contact: _____
 Position/Title: _____
 Company: _____
 Address: _____
 _____ Postcode _____
 Tel: BH _____ AH _____
 Fax: BH _____ AH _____
 email address: _____

STUDENT MEMBER CERTIFICATION: (to be completed by a member of the academic staff)

I, _____ certify that
 _____ (administrator)
 _____ is a full time student at
 _____ (name)
 _____ and is expected to
 _____ (institution)
 graduate approximately _____ (date)

Signature

Title

Date

Over for Institutional Membership

Please charge \$ _____ to my
 Bankcard, Visa, Mastercard
 Account number: _____
 Expiry date: _____
 Name on card: _____
 Signature: _____

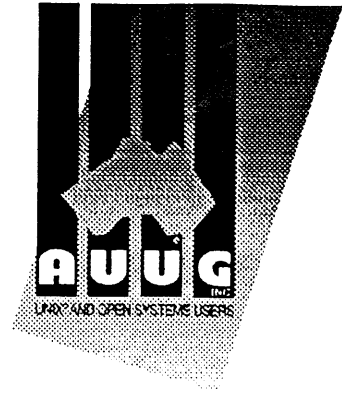
AUUG Secretariat Use

Chq: bank _____ bsb _____
 a/c _____ # _____
 Date: _____ \$ _____
 Initial: _____
 Date processed: _____
 Membership # _____

AUUG Inc. as a user group, exists to provide UNIX® and open systems users with relevant and practical information, services, and education through cooperation among users.

MEMBERSHIP APPLICATION

INSTITUTIONAL



To apply for AUUG membership, complete this form and return it with payment in Australian Dollars to:

**REPLY PAID 66, AUUG MEMBERSHIP SECRETARY,
P.O. BOX 366, KENSINGTON, NSW 2033, AUSTRALIA
Tel: +61 2 361-5994 Fax: +61 2 332-4066**

Tick this box if you wish your name withheld from mailing lists made available to vendors.

NOTE: Please do not 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, and remember to select either surface or air mail.

We, _____
(Company Name)

do hereby apply for:

Renewal/New* Inst. membership of AUUG \$350.00
International surface mail \$ 40.00
International air mail \$120.00

TOTAL REMITTED AUD\$ _____

(Cheque, money order, or credit card)

I/We agree that this membership will be subject to the rules and by-laws of AUUG as in force from time to time, and that this membership will run from time of joining/renewal until the end of the calendar or financial year.

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.

Signed _____ Date _____
Title _____

INSTITUTIONAL MEMBER DETAILS:

To be completed by institutional members only.

Following are our specified contacts. The primary contact holds the full member voting rights. The two designated reps will also be given membership rates to AUUG activities including chapter activities. By default a regional chapter will be selected for you. If you would rather nominate a chapter, please specify in space provided (indicate NONE for no chapter). (Please print clearly or type)

Primary Contact _____
Position/Title _____
Address _____
Postcode _____
Bus. Tel: _____ Bus. Fax: _____
e-mail address _____
Local Chapter Pref. _____

1st Rep. _____

Position/Title _____

Address _____

Bus. Tel: _____ Bus. Fax: _____

e-mail Address _____

Local Chapter Pref. _____

2nd Rep. _____

Position/Title _____

Address _____

Bus. Tel: _____ Bus. Fax: _____

e-mail address _____

Local Chapter Pref. _____

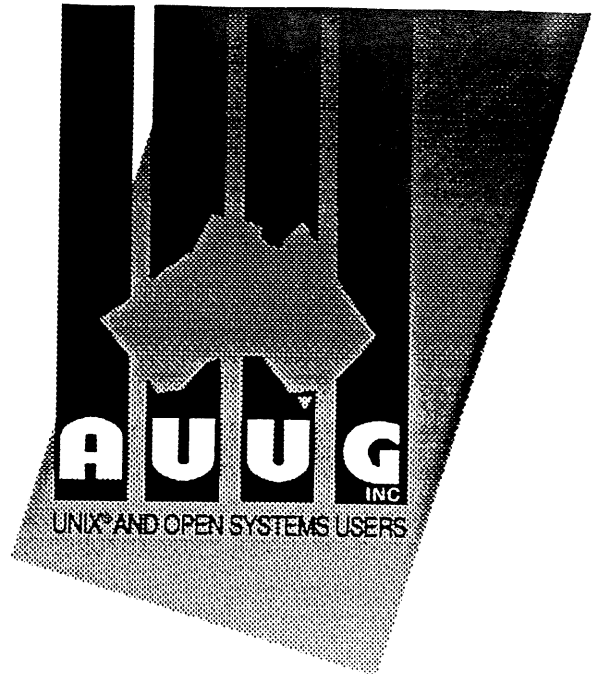
Please charge \$ _____ to my
 Bankcard, Visa, Mastercard
Account number: _____
Expiry date: _____
Name on card: _____
Signature: _____

AUUG Secretariat Use

Chq: bank _____ bsb _____
a/c _____ # _____
Date: _____ \$ _____
Initial: _____
Date processed: _____
Membership # _____

AUUG Inc. as a user group, exists to provide UNIX® and open systems users with relevant and practical information, services, and education through cooperation among users.

Notification of Change



You can help us! If you have changed your mailing address, phone, title, or any other contact information, please keep us updated. Complete the following information and either fax it to the AUUG Membership Secretary on (02) 332-4066 or post it to:

AUUG Membership Secretary
 P.O. Box 366
 Kensington, NSW 2033
 Australia

(Please allow at least 4 weeks for the change of address to take effect..)

- The following changes are for my personal details, member #: _____
- The following changes are for our Institutional Member, primary contact.
- The following changes are for our Institutional Member, representative 1.
- The following changes are for our Institutional Member, representative 2.

PLEASE PRINT YOUR OLD CONTACT INFORMATION (OR ATTACH A MAILING LABEL):

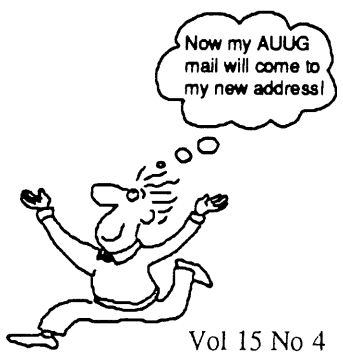
Name/Contact: _____
 Position/Title: _____
 Company: _____
 Address: _____
 _____ Postcode _____
 Tel: BH _____ AH _____
 Fax: BH _____ AH _____
 email address: _____

PLEASE PRINT YOUR NEW CONTACT INFORMATION:

Name/Contact: _____
 Position/Title: _____
 Company: _____
 Address: _____
 _____ Postcode _____
 Tel: BH _____ AH _____
 Fax: BH _____ AH _____
 email address: _____

AUUG Secretariat Use

Date: _____
 Initial: _____
 Date processed: _____
 Membership # _____



AUUG Incorporated

Application for Newsletter Subscription

AUUG Inc.

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
 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.
- 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, 1995

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:

- | | |
|---|-----------|
| <input type="checkbox"/> Subscription to AUUGN | \$50.00** |
| <input type="checkbox"/> International Surface Mail | \$ 20.00 |
| <input type="checkbox"/> 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.
 ** This includes a copy of the proceedings of the AUUG Winter Conference.

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# _____