

AUUGN

The Journal of AUUG Inc.

Volume 19 • Number 4

November 1998



Images of AUUG'98

The future of AUUG:
News from the
Management Committee

Traps & Tricks and Book Reviews

UNIX & OPEN SYSTEMS USERS

AUUGN

The Journal of AUUG Inc.

Volume 19 • Number 4
November 1998

AUUG Membership and General Correspondence

The AUUG Secretary
PO Box 366
Kensington NSW 2033 Australia

Tel: 02 9361 5994
Fax: 02 9332 4066
Toll Free: 1800 625 655
Internet: auug@auug.org.au

AUUG Management Committee

President:

Lucy Chubb
Lucy.Chubb@auug.org.au
Softway Pty. Ltd.
79 Myrtle St
Chippendale NSW 2008

Vice President:

David Purdue
David.Purdue@auug.org.au
PO Box 1013
Artarmon NSW 1570

Secretary:

Mark White
Mark.White@auug.org.au
Tandem Computers
143 Coronation Drive
Milton QLD 4064

Treasurer:

Stephen Boucher
Stephen.Boucher@auug.org.au
MTIA
509 St. Kilda Road
Melbourne VIC 3004

Committee Members:

Malcolm Caldwell
Malcolm.Caldwell@auug.org.au
Northern Territory University
Casuarina Campus
Darwin NT 0909

Luigi Cantoni
Luigi.Cantoni@auug.org.au
STM
Suite 3, 77 Mill Point Road
South Perth WA 6153

Günther Feuereisen
Gunter.Feuereisen@auug.org.au
PO Box 1350
Double Bay NSW 1360

Peter Gray
Peter.Gray@auug.org.au
Information Technology Services
University of Wollongong
Wollongong NSW 2522

Michael Paddon
Michael.Paddon@auug.org.au
Australian Business Access
723 Swanson Street
Carlton VIC 3053

AUUG Business Manager

Elizabeth Egan
busmgr@auug.org.au
Level 4, 90 Mount Street
North Sydney NSW 2060

Editorial	3
President's Column	3
News from the AUUG Management Committee	4
AUUG'98: Special Thanks	5
UniForum NZ '99	5
Zero Risk Legacy System Re-engineering	7
News from the AUUG Business Manager	10
From the pages of unigram•X	11
An interview with ...	17
Images of AUUG'98	19
Shoulders of Giants	24
Book Reviews	39
AUUG Local Chapter Meetings 1998	40
UNIX Traps & Tricks	42

Thanks to our Sponsor:

TELLURIAN

Contribution Deadlines for AUUGN in 1999

Volume 20 • Number 1 • February 1999 : **January 7th 1999**

Volume 20 • Number 2 • May 1999 : **April 7th 1999**

Volume 20 • Number 3 • August 1999 : **July 7th 1999**

Volume 20 • Number 4 • November 1999 : **October 7th 1999**

AUUGN Editorial Committee

Editor:
Günther Feuereisen

Sub-Editors:
Matthew Dawson "Unix Traps & Tricks"
Mark Neely "Book Reviews"

Public Relations and Marketing:
Elizabeth Egan

Please send all correspondence regarding AUUGN to:

AUUGN Editor
auugn@auug.org.au
PO Box 366
Kensington NSW 2033 Australia

AUUGN Submission Guidelines

Submission guidelines for AUUGN contributions are regularly posted on the aus.org.auug news group.

They are also available from the AUUG World Wide Web site at:
<http://www.auug.org.au>

Alternately, send email to the above correspondence address, requesting a copy.

AUUGN Back Issues

A variety of back issues of AUUGN are still available; for price and availability 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 copies of the Conference Proceedings from previous AUUG Conferences are still available. Contact the AUUG Secretariat for details.

Mailing Lists

Enquiries regarding the purchase of the AUUGN mailing list should be directed to the AUUG Secretariat:

Tel: (02) 9361 5994
Fax: (02) 9332 4066

During normal business hours.

Disclaimer

Opinions expressed by the authors and reviewers are not necessarily those of AUUG Inc., its Journal, or its editorial committee.

Copyright Information

Copyright © 1998 AUUG Inc.
All rights reserved.

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

Copyright without fee is permitted, provided that copies are made without modification, and are not made or distributed for commercial advantage.

Editorial

Günther Feureisen
<Gunther.Feureisen@auug.org.au>

Well, it's the last issue of the year – where on Earth did 1998 go?

This issue we bring you some important news on the future of AUUG, and the idea of AUUG Special Interest Groups. Please look at the ideas presented on page 4, and if you have any feedback, please let the Management Committee know. Thoughts and ideas are most welcome.

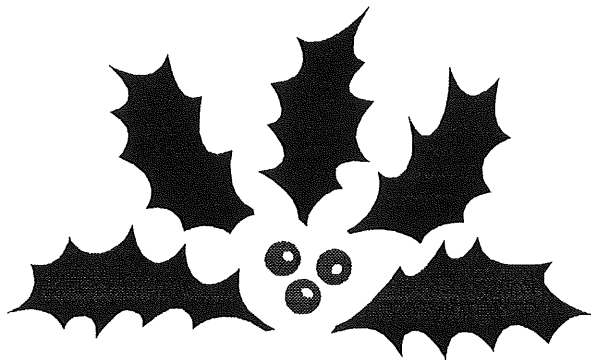
We wrap up AUUG'98 – with pictures from the Conference; thank you to all involved for a wonderful time.

To those who have contributed to AUUGN throughout the year – thank you for your support!

Finally, I would like to thank my AUUGN team; Matthew Dawson, Elizabeth Egan, and Mark Neely. Without their help and efforts, we wouldn't be here.

Merry Christmas everyone – see you in '99!

günther



President's Column

Lucy Chubb
<Lucy.Chubb@auug.org.au>

Well, the winter conference has come and gone for this year and I, for one, enjoyed it. There were some good talks, I had a great time getting together with the other attendees over the meals and in the bar, and an interesting time wandering around the exhibition. This conference had a bit more of a technical flavour than some in the recent past and overall those who came to it seemed pleased with that change. I expect that trend will be continuing. The next winter conference will be in Melbourne in 8 - 10th September. Put it in your diary now so you don't forget.

Some of you (those who aren't on the auug_announce mailing list) may have wondered why you've suddenly started receiving Systems Magazine. An agreement has been made with Systems Magazine that will provide each AUUG member with a year's subscription to that magazine at no extra cost to you. Every second month Systems magazine will contain an AUUG column. This column will feature some of the AUUG roadshow presenters, information about the main yearly conference, and other items of more general interest to AUUG members and other readers of Systems magazine. It would be great to have members contributing the column, so if you would like to contribute, please contact the AUUG Business Manager. I hope you find this a useful benefit of belonging to AUUG.

We have been discussing the creation of a new membership type, associate, which will require the membership to vote on a change to the constitution. At the same time we will be looking at creating special interest groups. These special interest groups could focus on any topic that is broadly consistent with the aims of AUUG, say, Linux. Any member could join a SIG, but associate members would be primarily members so that they could join a SIG and would not receive all of the membership benefits of AUUG (mainly those provided by the SIG). Have a look at the more extensive discussion of the proposed changes later in this issue of AUUGN and get in touch with us if you have any comments.



News from the AUUG Management Committee

The AUUG management committee has been thinking about how to create special interest groups within AUUG, and how existing special interest groups with overlapping aims and interests (such as the Linux User Groups) could become associated with AUUG, if they choose, and gain the benefit of AUUG's existing infrastructure while continuing as low cost groups that cater for different but closely related interests. At the last AGM a show of hands demonstrated that there was a great deal of support from members for the management committee to investigate practical ways for this to be done.

The management committee will be recommending the changes to the constitution needed to introduce a new class of member, to be called "Associate". Associate members will nominate one or more Special Interest Groups (SIGs) that they would like to belong to. (Of course, any full member will also be able to join one or more of these groups as well). As the name suggests, we expect that SIGs will focus on much narrower areas of interest than AUUG in general.

The cost of becoming an Associate member will be significantly less than for other membership classes, but an Associate member will not have access to the full range of member benefits provided by AUUG. A SIG will be in many ways like a Chapter but additional membership fees may be charged by the SIG (which may also choose to charge no extra fees). It may also choose to provide additional benefits to its members. However, it is not envisaged that AUUG normally provide extra financial support to SIGs (as is done from time to time for chapters) beyond the SIG fee collected on its behalf.

The members of each SIG will vote for their SIG management committee but Associate members will not be voting members of AUUG.

A postal ballot of all AUUG members will need to take place for the changes to the constitution which will allow the new membership class of "Associate" to be introduced.

The draft changes are as follows.

In section 7, paragraph 1 changes from:

There shall be four classes of members: Ordinary members, Institutional members, Student members and Honorary Life members.

to

There shall be five classes of members: Ordinary members, Associate members, Institutional members, Student members and Honorary Life members.

In section 7, paragraph 2 changes from:

Any natural person who is eligible to be a member may become an Ordinary member.

to

Any natural person who is eligible to be a member may become an Ordinary member or an Associate member.

In section 9, an additional paragraph is added:

Membership of a Special Interest Group may require the payment of a Special Interest Group membership fee in addition to the Associate membership fee which will be set from time to time by the Special Interest Group Management committee.

In section 35, paragraph 2 changes from:

Each member shall be sent a copy of the association's newsletter

to

Access to member benefits shall be set for each member class from time to time by the management committee.

In section 31 paragraphs 1, 3, and 4 references to "chapters" are replaced by references to "chapters and SIGs".

The constitution of a SIG operates within the AUUG constitution and within such other policies and procedures as are determined by the AUUG management committee. The constitution of a SIG or any amendments to an approved SIG constitution may not be enacted without written approval of the AUUG management committee.

The AUUG management committee welcomes any comments on these proposed changes. You can email the management committee at auugexec@auug.org.au

❖

AUUG'98: Special Thanks

We would like to extend a special thanks to all those involved in AUUG'98, including our Major Sponsors and Exhibitors:

**Advanced User Systems
Attachmate Pty Ltd
Auscom Publishing Pty Ltd
Cisco Systems Australia
CITEC
Compaq
Custom Technology
Dynamic Australasia Pty. Ltd
IBM
Integration Systems Pty Ltd
Network Systems Technology Pty Ltd
Primus Hotkey
Red Hat
SCO
Softway
StorageTek Asia South Pacific Pty Ltd
Sun Microsystems**

A special thanks from everyone at AUUG to SLUG members:

**Jim Donovan
Ken Caldwell
Scott Howard**

for all their help in setting up the email terminals at this years conference.

And finally, a special thanks to the conference and program committee, for all their efforts and hard work.

**Frank Crawford (Conference Chair)
Andrew McRae (Program Chair)**

And to all of those who helped whom we've forgotten to mention – THANK YOU!!



UniForum NZ '99

THE BUSINESS OF CHOICE

16th Annual Conference

13-17 April 1999
Rydges Hotel, Rotorua

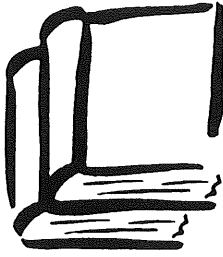
Combine work with pleasure! Attend UniForum NZ '99 and stay on to see the sights of one of New Zealand's top tourist destinations.

The industry event of 1999. Where else can you hear experts in all these topics, under the one roof!

Distributed Systems	Unix/NT integration
Internet	Networking
Cryptography	Legal issues
Year 2000	More!!

Registration brochures will be available mid-February 1999. Contact Ray Brownrigg, Ph 64-4-472 1000 x7068 or fax 64-4-495 5118. Email ray.brownrigg@uniforum.org.nz. Or write to UniForum NZ, P.O. Box 585, Hamilton, New Zealand.





*10% DISCOUNT
TO AUUG MEMBERS
ON OUR COMPLETE RANGE
OF COMPUTER, BUSINESS
AND GENERAL BOOKS*

**AMERICAN
BOOK STORE**

**173 Elizabeth St, Brisbane Queensland 4000
Ph: (07) 3229 4677 Fax: (07) 3221 2171 Qld Country Freecall: 1800 177 395
american_bookstore@compuserve.com**

Name: _____ Date: _____

Address: _____

_____ Post Code: _____

Phone Number: _____

Payment Method: Cheque Money Order Amex Bankcard
 Diners Mastercard Visa

Card Number: _____

Expiry Date: _____ Signature: _____

This is a: Special Order Mail Order Book on Hold

QUANTITY	TITLE	PRICE
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....

SUBTOTAL	\$
LESS 10% DISCOUNT	\$
POST & PACK	\$
TOTAL	\$

POSTAGE AND HANDLING FEES: 1 BOOK \$6.00 2-4 BOOKS \$7.00
BOOKS OVER \$70.00 WE WILL SEND CERTIFIED - PLEASE ADD ANOTHER \$1.50 OR WAIVE CERTIFIED DELIVERY.

FOR SPECIAL ORDERS, PLEASE ENCLOSE \$10.00 PER BOOK AS A DEPOSIT.

Zero Risk Legacy System Re-engineering

OR, HOW TO SQUEEZE 200+ CONCURRENT USERS ONTO ONE LITTLE I486

David Newall
<david.newall@tellurian.com.au>

We helped develop a system that provides users with a new GUI application, which they use in place of several old legacy systems. Instead of replacing the old legacy systems, the new application adds value to them, and does so without them being altered in any way. The new application hides the different legacy systems from the users, and gives an illusion of a single, integrated system. A useful side-effect of this approach is that the life of the legacy systems has been extended: it achieves performance targets for more concurrent users than it could have had those users performed equivalent work without using the GUI.

INTRODUCTION

This is a short paper. It is part technical discussion and part narrative for a system development project.

We were asked to help design and implement a new customer support system for a major corporation. Growth of the number of different computer systems that their support staff needed to use concerned the corporation. They wanted to reduce the number of separate systems that their staff needed to learn. They also wanted to empower their staff to be able to answer questions about any and all products, which meant their staff needed access to equivalent systems. Whilst about it, they wanted to enhance their level of customer support by being able to track contacts with their customer and also the problems their customers' reported. Finally, they felt that the business rules embodied in the legacy systems were hard to define, and that replacing those systems with something new would be a long-term project: They wanted the new functions immediately but could not risk discarding or altering the old systems. Oh; and the project was to be a proto-type, to prove the concept, and consequently had a very low budget.

DESIGN

The application was designed using a client-server model, where the client (that is the GUI) communicated with servers using a request/reply interface. The system architect divided the project into four major components: The GUI; contact and problem tracking; the legacy systems interfaces; and something to co-ordinate all of the components. The

GUI had to run on Macintosh-that was the standard desktop. Most of the legacy systems were hosted on the IBM mainframes, and were designed for use via 3270 terminals. Some further legacy systems were hosted on Digital Vaxen running VMS. Other systems ran on Tandems. Obviously only Unix had the technical features, stability, interoperability, and attractive price/performance ratio needed for a complicated system such as this. Since the budget was very small, Intel architecture PCs were selected. SCO Unix was chosen for the operating system, which provided its own, unique challenge.

Our task was to provide infrastructure to interface to the legacy systems, and to do so in such a way that future changes to the legacy systems would not result in major changes to our system. As things turned out, our brief grew to encompass the complete system, however the design of the GUI, and therefore its 'taste', was "set in concrete" before we took it over. Similarly, we were given the design of the contact and problem subsystem.

A request/reply model is similar to a transaction paradigm, and the system architect nicely followed that paradigm in the design of the server. The server, or "back-end" as it came to be called, was written as a series of transaction-servers. Tuxedo was selected as a transaction monitor; its use was to route requests and replies, and to balance transaction load amongst the available server processes.

Communication between the GUI and the back-end takes the form of a simple text-based stream, and uses TCP/IP. There was no Tuxedo interface for Macintosh (although I believe this is no longer the case), and so we wrote a simple program, which ran on the back-end as a TCP server, that the GUI connected to. It acts as a gateway between the GUI and Tuxedo, and is technically the Tuxedo client (Tuxedo provides a RPC type of client/server mechanism.)

There often is confusion over names we use. 'GUI' refers to the program running on the Macintosh. The 'client' refers to the Tuxedo client, which the GUI connects to, and which serves to translate requests from the GUI into Tuxedo service calls, and to format Tuxedo replies appropriately for the GUI. The 'servers' are the various programs that act on the GUI's requests. The 'legacy systems' are the various systems running on the IBM hosts, the Vaxen, the Tandems and what have you. I describe our system architecture as multi-level client/server.

I shan't describe the GUI in great detail. It's not particularly interesting, and in fact it's not even particularly beautiful. I should go further and say an expert user who had strong ideas about look and feel designed the appearance and operation of the GUI. Without intending to denigrate the design skills of users, the ease of use of the GUI, and in fact the performance of the whole system, did suffer a little

from lack of design experience. On the positive side, this user was also the system champion: the project would have failed without having had someone willing to sell the idea to management; to guide our understanding of the problem; to feed us pizza during the long nights; and to stand up to upper-management when we had some early disasters. I suppose the user is pleased that, far from losing the money that was budgeted (which is what the IT department predicted would happen), the final solution was approximately on-time, within budget, and according to specifications.

I also shall not thoroughly describe the contact and problem reporting subsystem. It is quite uninteresting. Data is stored in a relational database (if you are interested it uses Informix, but is soon to be Oracle) and the GUI can send requests to add, list, retrieve and update data.

A caution as an aside: This system was to be a prototype. Five years later it is still in operation, and it has outlived two serious attempts to replace it. The replacement systems were expensive and cumbersome, and there is now a movement to make this 'prototype' be a permanent part of the corporations IT infrastructure. Lesson to be learned: Never believe anybody who tells you that it's only got to be a throw-away piece of code. Throwaway code is used forever. Another lesson, albeit one of a somewhat political nature: Large software projects take longer to produce less effective results than small projects. If it's really critical to your business, leave it to a small team, say around five people, and give them a sensible budget; this will save millions of dollars, but more important: you'll get results that work.

LEGACY SYSTEM INTEGRATION

Most of the legacy systems are traditional character based applications. In the style of these things, they guide operators from screen to screen, collecting information and providing results as they go. This really does not fit with the request/response design of the GUI. Design of the new system included deciding what features (functions) of the legacy systems would need to be mirrored. These features were described by the result, not by the navigation or menu options; for example, "retrieve the invoice for a particular date," or "change the account type for this account to that type."

The descriptions turned into the interface for the server programs, thus we were able to define transaction-oriented interfaces for an interactive, character based program. Before we could use this transaction interface we needed two things: We needed to document how to operate the legacy applications in order to achieve the appropriate result; and we needed some way to connect to the legacy system that made us look like a person sitting at a terminal.

SNA GATEWAY

Our application had to appear to the legacy system as a 3270 terminal, however we did not want to tie our system to closely to 3270. We realised that our system (despite being a prototype) might well outlast the legacy system, and there was no reason to suspect that legacy system's replacement would use 3270. We defined a gateway protocol that encapsulated the essence of a terminal, without being specific to 3270. A SNA stack with HLLAPI was purchased, and our protocol was implemented. That first implementation was written for Macintosh; it's odd how decisions get to be made, for example, the standard desktop platform was Macintosh, so of course all server platforms should be Macintosh where possible.

The Macintosh version of the 3270 gateway was replaced with a Unix version, again running on i486 computers and SCO, and using an SNA protocol with HLLAPI which was written by Philips. This was much more satisfactory from our point of view: In addition to working properly (which the Macintosh version really did not do), this version was easier to administer, particularly given that the hardware resides in the eastern states and we reside in the central state.

The gateway protocol is quite simple. The purpose of the gateway is to make our system appear to be a 3270 terminal; only a few commands are needed. Commands include press keys; return a section of the screen; and wait for the legacy application to stop updating the screen.

LEGACY CONTROL

It might be tempting to imagine that sequencing of legacy systems is easy and can be table driven: Assuming we are at a known point, say the main menu, you can define a table that shows what keys you press to go from screen to screen; and to define the screens you need to navigate through; and to define the data you need to type in order to perform the work. Actually this is far from sufficient. It certainly does not cover complicated situations, such as when you need to go to one part of an application, capture some data, go somewhere else in the application and enter the data which you previously captured. Character based applications often require this sort of interaction. Not only does the table based approach fail to solve more than the simple interaction, it fails to help with exceptions, meaning when some error occurs.

We wrote a new language to sequence interaction with the legacy system. The language grammar is simple. It provides the usual features, such as variables, numeric and string expressions, conditional and unconditional loops and branches, procedure calls, and also provides features specific to this

application domain, namely pressing keys, reading sections of the screen, waiting for key phrases to appear on the screen. We find that writing programs to control the legacy application is quick and easy.

Each program written in our control language has a name, a list of input parameters, and a list of output parameters. The name corresponds to the type of request that the GUI makes; the input parameters are the provided with that request; and the output parameters constitute the reply that is returned to the GUI.

The programs are compiled to a psuedo-code that is interpreted by a server program. The server program is a Tuxedo server that selects and interprets the appropriate compiled psuedo-code needed to service each request that it receives.

PERFORMANCE

The system works well. Surprisingly well! We have abstracted other applications, and perform scripted operation of those other applications, and the result is usually much faster than an expert operator would achieve sitting at a terminal using that other application. Not only is it faster, but the number of logins to the other applications is much smaller than the number of people using our system. Our system functions like a multiplexor, allowing many users to funnel through a small number of sessions. It keeps those small number of sessions very busy, however we are told by the administrators of the legacy system that the impact on their system is much less than would be if each of our users were logged directly into theirs. They have told us that costly upgrades to the IBM host have been avoided through the use of our technology.

We observe a tremendous economy of facilities in our approach. We support almost 600 concurrent users during peak periods, and use a total of seven small machines. The particular machines have changed over the life of the project. We started with two 50MHz i486 machines, each with 64MB of RAM, one for running the client and the other for the back-end and database. (This ignores the SNA gateway machines.) We chose to limit the number of client connections to the client machine to 220. As our user-base grew we expanded our hardware to give us three i486 machines. At around 500 concurrent users we found that CPU requirements for legacy system control became more than could be handled by a single i486, so we added a second i486 giving us database on one server and legacy systems on the second. Later we replaced those two i486 servers with a single Pentium-120 with 96MB RAM.

Our system is very reliant on correct operation of the legacy systems, and in fact we are usually the first to discover problems with the legacy systems. It is almost a truism that problems reported with our system can be used as a barometer for problems with

the legacy systems. In fact, the infrastructure that we developed has proven so successful that the corporation has built a number of new legacy-style systems with the express intention that they would only be accessed via our GUI. This makes sense because it allows the mainframe programmers to stay in their comfortable environment, writing the same style of code that they have written for the past 20 years.

CONCLUSION

Providing an integrated application on the desktop allowed the customer's staff to focus on their work, rather than on the computer systems. The system was designed around what jobs needed to be done even though there was no single computer system which supported those jobs directly.

Without this system, the legacy systems would not have been able to cope with the businesses growth in user load. Likewise, the training expense, had all users needed to learn all legacy systems, would have been prohibitive.

The system has proven to be very efficient, supporting hundreds of concurrent users on small computers. It has given broad access to large parts of the customer's total computing environment, and done so without requiring legacy system changes or upgrades.



For the latest
news on
AUUG

Check out the AUUG website at:

www.auug.org.au

TELLURIAN

Tellurian Pty Ltd

Come to us if you need seriously capable people to help with your computer systems. We're very good at what we do.

- Unix, Macintosh and Windows experts
- Legacy system re-engineering and integration
- System management and support
- Internet access

Our two current major projects:

- Support and development of an integrated environment covering applications running on IBM3090, DEC Alpha, SCO Unix and Nortel switches. Just imagine the cost benefits of supporting over 500 concurrent users on four little 486 and Pentium PC's.
- From the ground-up implementation of MFC and Windows API on Apple Macintosh. We've got our client's Windows MFC application running, bug-for-bug, on Apple Macintosh.

Tellurian Pty Ltd
272 Prospect Road
Prospect SA 5082

(08) 8408 9600
www.tellurian.com.au
sales@tellurian.com.au

News from the AUUG Business Manager

Liz Egan
<busmgr@auug.org.au>

Hi All,

SYDNEY MORNING HERALD

By now you are probably all aware that AUUG is running a fortnightly column in the Sydney Morning Herald. (Thank you to everyone who has contributed to date.) As such, we are always looking out for good articles. Should anyone wish to contribute please email me your article for submission. If you are unsure whether something is appropriate, an outline of the topic will be sufficient and I will let you know whether it will be okay. Following is a brief outline of the type of article the SMH is looking for.

"The Sydney Morning Herald is targeting its IT section to business, running it after the business section and targeting readership at CEO's down. The

column is a UNIX column (rather than an AUUG column). SMH audience also runs 80-90% of their IT section in the Age in Melbourne. UNIX Column - 600-700 words with final credit to writer and pointer to AUUG Web site."

SYSTEMS MAGAZINE

In addition to the SMH, AUUG is also running articles in Systems Magazine (copies of which AUUG members are now receiving). As such, articles of approximately 1,700 words would also be appreciated. Two pieces will be included in the next edition, as follows: Thanks to Lucy, Mark and Gunther for their contribution.

UNIX: ITS A GREAT TIME TO BE ALIVE
By Lucy Chubb & Mark White

BACK-UP AND THINK AGAIN
Author: Gunther Feuereisen

The next deadline will be 14 December.

Should anyone have any questions, please feel free to email me or call on 02-9858-4542. I look forward to reading your articles!!

❖

From the pages of **unigram·X**

Compiled by:

Günther Feuereisen

<Gunther.Feuereisen@auug.org.au>

"USER FRIENDLY" POWERPC LINUX 4.0 LAUNCHED

LinuxPPC Inc has unveiled PowerPC Linux Release 4, which it says is the first major distribution of the open source operating system to boot into a graphical user interface on startup. Though generally recognized as a capable operating system, Linux has been singled out for criticism for its perceived difficulty of use. Release 4 addresses those criticisms by supporting the KDE graphical environment. The OS ships with a software bundle including Netscape Communicator 4.0.5 and graphics editor GIMP 1.0. PowerPC Linux runs on everything from Power Macs and Mac clones to the G3 series, Apple's Network Server, BeBox, IBM ThinkPad 8050, Motorola's PowerSTACK and several models of RS/6000. PowerPC Linux's G3 support is particularly interesting. Be Inc, which markets a Unix-ish operating system not entirely unlike Linux, has complained publicly about Apple Computer Corp's refusal to release proprietary specs which would help BeOS developers port a version to the G3 platform.

SUN TO SIMPLIFY APPLICATIONS INSTALLATION ON SOLARIS

Sun Microsystems Inc has moved to eliminate one of the key differentiators between its Solaris Unix and Windows NT with the announcement of an agreement last week with InstallShield Software Corp to provide the first point and click graphical method of installing Solaris applications, which until now have been installed using command line scripts. InstallShield, which the company claims is used by at least 90 of the top 100 Windows independent software vendors to install their applications either to Windows-based PCs or over the web, has had a Java version of its product out for more than a year - it's currently on version 2.01. The two companies have taken some of the Java classes in that product, added Sun's RMI remote method invocation Java-to-Java distributed communications technology plus some other internet technology and produced a software development kit which will be available for download from Sun today. Sun is calling this early access version of Solaris Web Start Wizards and says upcoming versions of InstallShield's Java Edition. Version 2.1 is expected within a couple of months - as well as the next version of Solaris, which is referred to internally as 2.7, but has not been given a final name yet beyond Solaris foundation next generation. That is due some time in the fall. Solaris will be the first Unix platform for which

InstallShield's tool has been adapted, although of course its Java version can install applications to any platform that has a Java virtual machine. The Solaris version, like the Java version, will have the same look and feel as the Windows version and it will add some Solaris-specific distributed application and configuration technologies from both companies. At the time as that release of Solaris, Sun will also have co-packaged Solaris applications available from third parties that will use InstallShield as their method of installation, whether locally or over the internet. Solaris version 2.6 included Web Start, a browser-based installer for the operating system and bundled applications - this builds on that and expands it to all application developers, says Sun. Eventually it could be used to install the OS itself. The SDK is at <http://www.sun.com/solaris/webstart>.

SUN LAST TO MARKET WITH FREE INTEL UNIX

Sun Microsystems Inc has finally followed Santa Cruz Operation Inc in distributing a low-end version of its Unix operating system free to students, researchers and enthusiasts. From now on, non-commercial users of Solaris on SPARC or Intel can get the operating system CD-ROM gratis from Sun's Developer Connection. They still have to pay for packaging and shipping. "In the past, the price of Solaris has been a barrier to this creative, innovative community," says Brian Troll, director of product marketing. That's an understatement: Linus Torvalds has often said that he wrote Linux because there was no inexpensive Unix-like PC platform available in 1992. Now Linux presents a real and present danger to Solaris, hence, perhaps, this release. Not so, says Troll. "There is a real Renaissance of super-creative activity going on right now, which to a large extent is fuelled by Linux," he concedes. However, he says: "Sun has always been playing in this crowd. This is the environment Sun sprang from." Well, maybe. It's true that the most serious threat to Solaris doesn't come from the Linux community but from Microsoft's Windows NT. Indeed, Troll claims Linux developers as allies, if not crack troops in the battle for hearts and minds. "A lot of this creative interesting work is not happening on the Windows environment. We're trying to take advantage of and encourage that," he says.

GETTING APPLICATIONS ONTO LINUX

Two recent events have raised intriguing possibilities for addressing the comparative paucity of applications on Linux. One is the ruling by a US federal court judge that Microsoft Corp must hand over parts of the Windows 95 source code to Caldera Inc. Caldera accuses Microsoft of artificially disabling Windows under rival versions of DOS - Caldera's own DR DOS, for example, acquired from Novell Inc in 1996. In doing so, Caldera alleges, Microsoft has defrauded alternate DOS vendors of years' worth of revenues. The case goes to court in June next year. The weakest win for Caldera would see Microsoft forced to modify Windows to run under DR DOS the same way it runs under MS-DOS.

The strongest win for Caldera, however, would have a judge make Microsoft document the interfaces that Windows uses to talk to DOS. With those interfaces public, it might be possible to support Windows on other kinds of DOS - on the Linux DOS emulator, for example. Some performance would be lost, but Linux is pretty fast on the same hardware, so it could all even out. The upshot: Linux users would be able to run applications originally developed for Windows. It's a messy solution, and you'd still have to do something about the parts of Windows that talk directly to the hardware without going through DOS. A more elegant solution has already been proposed, this one by open source developer Don Yacktman. Yacktman has called on executives at Apple Computer Corp to publish parts of the Mach kernel that underlies Mac OS, and to release them with the Yellow Box developer tool kit . Publishing Mach is effectively the same as publishing Mac OS's interfaces. If Apple were to take this step, open source developers could devise a version of Linux to run where Mach is now. Hey presto, Mac OS applications on Linux. Both solutions have the felicitous side-effect of providing Linux with a graphical user interface already familiar to non-technical types. At the moment Linux lacks a strong GUI, not least because development effort is divided between advocates of the partly- non-free KDE and evangelists for pure-open-source Gnome. It's not really a question of slapping a smiley face on Linux, anyway. Linux, Windows and Mac OS are all interesting general- purpose operating systems with different strengths and weaknesses. What users really need is freedom to choose the right tool for the right job. To avoid anarchy in a heterogeneous world, there would need to be a single set of APIs to develop to; fans of the former NeXt environments propose Yellow Box as just such a framework, hence Yacktman's desire to see it released as open source. Others worry that Yellow Box, like the Windows APIs themselves, are old and compromised by the need to support legacy systems. That's how it is with software. Every time you think you've solved a problem, someone comes up with 23 new and subtly different sub-problems.

LINUX COMPATIBILITY STANDARDS EXCLUDE CALDERA, SUSE?

GNU/Linux outfit Debian and Linux company Red Hat Inc have announced plans to produce a written specification for a Linux Compatibility Standard (LCS). The idea is to make it easier for developers to ensure that their applications will run on different distributions of Linux, and to help distributors ensure that their distributions meet community standards. Ever alert for a conspiracy, the Linux community pounced on the perceived exclusion of the other big Linux distributors, Caldera Inc and S.u.S.E, from the announcement. Red Hat spokesperson Melissa London hastened to reassure all and sundry that no one would be excluded from involvement with the LCS. The working group will be co-managed by Red Hat's Erik Troan and Debian's Dale Scheetz. All

interested parties are encouraged to subscribe by mailing a request to lcs-eng-REQUEST@lists.debian.org.

VERITAS WORKING ON COMMON FILE FORMAT FOR NT AND UNIX

Veritas Software Inc is in the process of putting together a consortium with "leading NT storage vendors" in an attempt to unify the different semantics of Unix and Windows NT file formats. A common format looks increasingly desirable as the concept of storage networks gains popularity. But the consortium looks as if it will be forced to come up with a third format in order to embrace the two systems - something that might cause some controversy at the volume end of the market. "I can't see Bill Gates or Scott McNealy agreeing on this one," said Dave Hitz, of Network Appliance Inc, whose company is not currently involved in the proposed consortium. Neither firm would be likely to accept an alternative to their own system. Network Appliance has consolidated Unix and NT file permissions and administration within its own proprietary file system. The Veritas approach might only end up adding a third alternative to the two common file formats currently in use, he said. Storage vendors such as EMC Corp, Amdahl Corp and Sun Microsystems Inc have been working on more basic file sharing facilities between mainframes and Unix at the high-end, while the Santa Cruz Operation Inc and the new breed of thin client network storage hardware vendors such as Network Appliance, have been working on NT/Unix integration. The Veritas "Open NT" file consortium would aim to get its specification accepted as a standard. Veritas does have a development agreement in place with Microsoft Corp, which is using Veritas technology within NT 5.0. And Sun Microsystems is expected to be a part of the announcement of Veritas' "Thor" high availability software launch next month, along with NCR Corp, EMC, StorageTek, Hewlett-Packard Co and Dell Computer Corp. Veritas, which wasn't commenting, is expected to be ready for an announcement next month. Meanwhile, Veritas said it has agreed with storage technology company Symbios Inc to work on storage applications for storage area networks.

SCO TO OFFER TANDEM'S NONSTOP CLUSTERS TO OTHER OEMS

Compaq Computer Corp says it's committed to move the NonStop clustering technology it acquired from Tandem Computers Inc out onto the mass market, and last week finally sealed its agreement with the Santa Cruz Operation Inc over the UnixWare version of the product. SCO says it plans to offer the technology to its other OEMs in the future. Compaq's Tandem division has actually been shipping NonStop Clusters for UnixWare on ProLiant Integrity XC systems to telecommunications companies since the beginning of the year, and will continue with direct sales. But now SCO has the rights to sublicense the software under the name UnixWare NonStop

unigram · X

A global, weekly news service for and about the Unix and open systems community, covering product and company news, Unix and open systems standards, industry politics and the effects of emerging technologies and new developments in software.

Contact Charles for a free trial copy on 61 2 9286 5951

NB: AUUG members get a 20% discount

unigram · X

Clusters. It will now begin an early access program, shipping the software as a bundled package, initially with UnixWare 2.1.2, for the rest of the year. Tandem's software, says SCO, gives it a two-year lead over its competitors. Offering scalability as well as availability, NonStop Clusters uses Tandem's single system image software - the only such technology currently available for Intel platforms - and scales up to six nodes, though systems of up to 30 nodes have been tested. SCO and Compaq intend to jointly establish a third party sales channel for clustered systems, which have traditionally been sold direct, and are beginning evaluation programs that include a free demo version, a series of development and competency centers worldwide, and an enterprise services and support operation. Database companies such as Computer Associates International Inc, IBM Corp's Software Group and Informix Corp came out in support of the initiative. And ICL's European High Performance Systems Group, along with its worldwide parent Fujitsu/ICL, said they were evaluating the product with a view to introducing the full version once it is generally available next year. Systems sold jointly with Compaq will continue to use Tandem's ServerNet as the high-speed link between cluster nodes, providing throughput of 50MB/sec, aggregated to 300Mb/sec over six nodes. But SCO wants to use standard hardware, and hopes to introduce a virtual interface architecture implementation of the product without ServerNet in the future. It says it has already demonstrated the product using Ethernet. Compaq says it doesn't intend to port the software over to other Unixes or to NT, although some aspects have already found their way into Microsoft Corp's "Wolfpack" Microsoft Cluster Server, through the joint Microsoft/Tandem agreement that's still in place. WolfPack is limited to two node failover operation. One thing that might hold up mass market acceptance is a lack of any standard clustering APIs for software developers. NonStop has its own set of service management-oriented APIs, to which Compaq is adding some data center APIs from Digital Equipment Corp's TrueCluster system. SCO and Compaq say they will monitor standards efforts, and expose a set of cluster-aware application APIs once they are settled. The UnixWare 7 version is being demonstrated, and is expected to become generally available in the first half of next year.

SCO SPONSORS LINUX

SCO has joined Linux International as a corporate member, evidently reasoning that the enemy of its real enemy - in this case, Microsoft Corp and its Windows NT operating system - is its friend. The sponsorship means money and support for the improvement and expansion of the use of Linux. Tion Johnson, the engagingly titled czar of free stuff at SCO, says the sponsorship is in recognition of the groundswell of popular support for Linux, once marginalized as a hobbyist platform but now steadily infiltrating corporate America. Johnson points out that SCO was a pioneering member of the 86open

group, which tried to get a common programming and binary interface together for various flavors of Unix. That turned out to be hard. Emulating Linux applications on SCO is likely to be a lot easier, and since most Linux distributions can already emulate SCO binaries, it effectively doubles the potential market for applications, making it a lot more attractive to develop for both platforms. Is this a tacit acknowledgement of UnixWare's defeat? Since Linux showed up in surveys as the only non-NT operating system to be increasing its market share, everyone from Computer Associates to Informix has jumped on the open source bandwagon.

COMPAQ CHOOSES ALPHA OVER MERCED

Compaq Computer Corp has now confirmed that it will use the Alpha RISC chip it acquired from Digital Equipment Corp as the basis for its proprietary Tandem Himalaya S-Series servers in the future - as we suggested it would. The servers, which run Tandem's NonStop Kernel operating system, currently use the MIPS RISC from Silicon Graphics Inc. The news is good for supporters of the Alpha chip, which at one stage looked under threat under the new ownership. The MIPS chip, having failed to crack the general CPU market, is now being aimed at embedded systems. Before Compaq's acquisition of DEC, it had intended to use Intel's Merced for high-end Tandem systems. The first Alpha-based Himalayas are expected to use the EV7 version of the chip and reach the market some time in 2001. Further upgrades to the MIPS chip will be made between now and then. Tandem's Unix-based Integrity servers, targeted strictly at telecommunications firms, and which constitute less than 15% of Tandem's business, are also MIPS-based, but are not expected to move over to Alpha. Efforts in that direction are mostly towards Intel-based systems, and Compaq has been shipping NonStop Clusters for UnixWare on Intel-based ProLiant Integrity XC systems to telecommunications companies since the beginning of the year. Tandem has also licensed its Unix-based NonStop technology to the Santa Cruz Operation and is working with Microsoft Corp to incorporate some of it into Windows NT.

LINUX COMMUNITY SKEPTICAL OVER UNIX DRIVER INTERFACE

Add the Linux developer community to the list of players who are skeptical about the Uniform Driver Interface project. Judging from comments made on the Linux kernel mailing list, free software developers are not enthusing about the project which, led by Adaptec, HP and Intel, seeks to create a common interface for device drivers across different operating systems. That's quite a problem for UDI's backers. Various pro-UDI factions have indicated that they want Linux developers to undertake the daunting task of writing UDI-compliant drivers for existing hardware. The Linux community wonders exactly what's in it for them. UDI's critics point out that judging by the spec, UDI drivers won't perform as well as native drivers, so it's still in developers'

best interests to develop the drivers for Linux alone. Meanwhile, SCO's apparently generous offer to write the reference implementation of UDI for Linux has been greeted with a range of reactions from suspicion to outright mirth. Can SCO possibly write software clean enough to be included in Linux's elegant kernel? That, as one contributor pointed out, is the \$64,000 question.

ORACLE CLARIFIES LINUX, INTERNET PLANS

Oracle Corp will release its own distribution of Linux with Oracle8 on a single CD. The company will also provide full Linux support as well as support for the database running on Linux, VP Mark Jarvis explained. "This is a first for Oracle," Jarvis says. He calls the port to the open source operating system "the most popular announcement we've ever made in the history of the company" noting that 22,300 developers signed up for the Oracle8- on-Linux developer program in its first ten days. "At that point we started to realize that Linux is a credible alternative to NT," says Jarvis. However he believes that stamp of approval works both ways: "We've given it credibility. Linux's credibility is dependent on big companies like Oracle stepping up and supporting it." None of this should be taken to mean that Oracle is pulling away from Microsoft, Jarvis says. "NT is still our fastest growing platform," he pointed out. He added that most people fail to appreciate that Oracle's databases run on all Microsoft operating systems, from WinCE on up - a claim Microsoft can't match with its SQL server, which runs only on NT. Because Oracle wants to concentrate on selling and supporting its database and applications software, the company has said it will spin off its Business Online web application hosting division later this year. "By spinning it off, we ensure that its ability to move rapidly and execute is greater," Jarvis claims. He's unfazed by rival Informix's announcement that it is moving in the other direction, acquiring data warehousing expert Red Brick. "So a dead database company buys a dead data warehousing company," he sniffs, "how should that affect Oracle?"

JAVA FREE SOFTWARE MOVEMENT GATHERS MOMENTUM

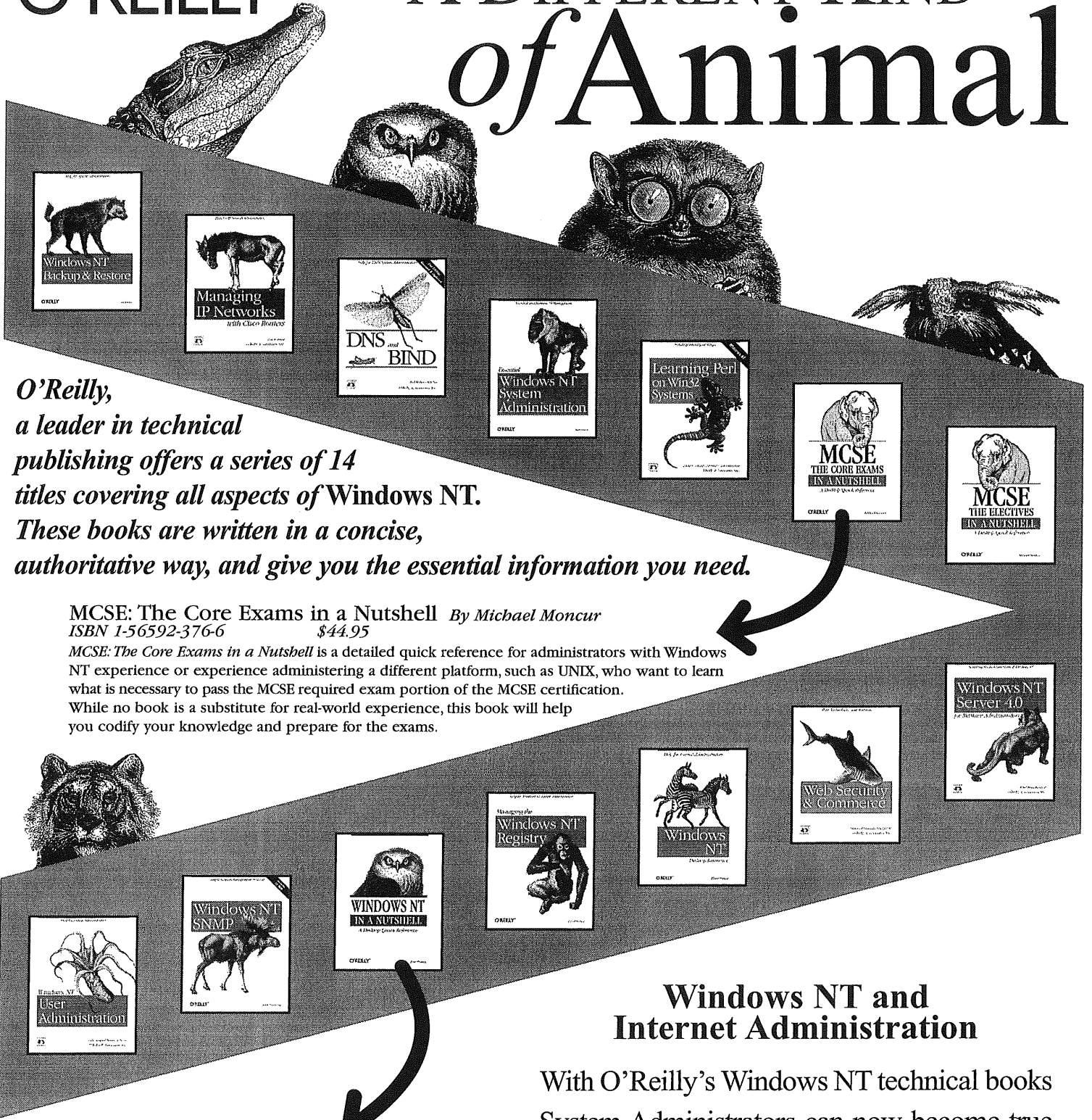
Linux may be the best thing that's happened to the hitherto ghettoized world of free software. Commercial ISVs are flocking to support the Unix-like operating system and soon users will be able to download a royalty and license-free version of Java for Linux (and other operating systems) that is completely compatible with Sun Microsystems Inc's product. GNU, the free software organization is working on a free implementation of the core Java class libraries, know as "GNU Classpath, Essential Libraries for Java." These free libraries, coupled with a free Java virtual machine (for example Japhar or Transvirtual's Kaffe), are said to create a free software environment which can run any Java program. The GNU work will support the complete Java 1.1 API. GNU expects its work to be very

popular amongst the programming community. "Java is a fun language to program in. That's really the main reason that the language has taken off so quickly - it's object oriented programming done right," the group says. "Java was designed for hackers to make their lives easier, and that it does very well. Coding in Java is much quicker than writing an application in C++, and as an added benefit, you get platform independence. But it's not all roses. GNU believes there are also many problems with the Java implementation from Sun. "The language is great, but Sun's implementation could be much better. Sun's implementation is proprietary - it's buggy - and it's slow. "Moreover the problems can't be fixed because Java isn't free software." GNU says Classpath project provides the core libraries and claims Java applications and developers can depend on them working correctly. "If they do find a problem, they can either email us, in which case we'll fix it as quickly as possible, or they can fix it themselves." Other Java free software includes the Guavac compiler, which is written in C++ and is portable to any platform supporting Gnu's C++ compiler or a similarly powered system. GNU/Linux is a free environment in which to run C, C++, Objective-C and Java. To be clear, GNU says we shouldn't use the term freeware to refer to free software. "The term 'freeware' has no clear accepted definition, but it is commonly used for packages which permit redistribution but not modification (and their source code is not available). These packages are not free software." GNU, on the other hand, says its system is only comprised of free software. Free software is code that comes with "permission for anyone to use, copy, and distribute, either verbatim or with modifications, either gratis or for a fee. In particular, this means that source code must be available." A good example would be embedded tools company Cygnus Solutions' recent enhancement of GNU's Java compiler. The Cygnus GNU Compiler Java Edition enables Java source and byte code to be compiled to native instructions thereby eliminating the speed bumps of interpreted or just-in-time models. The enhanced compiler is available free of charge but to be effective needs to be used in conjunction with Cygnus' GNUPro toolkit which enables developers to write embedded Java applications that will run on some 125 host/target systems which Cygnus supports.



O'REILLY™

A DIFFERENT KIND of Animal



O'Reilly, a leader in technical publishing offers a series of 14 titles covering all aspects of Windows NT. These books are written in a concise, authoritative way, and give you the essential information you need.

MCSE: The Core Exams in a Nutshell *By Michael Moncur*
ISBN 1-56592-376-6 \$44.95

MCSE: The Core Exams in a Nutshell is a detailed quick reference for administrators with Windows NT experience or experience administering a different platform, such as UNIX, who want to learn what is necessary to pass the MCSE required exam portion of the MCSE certification. While no book is a substitute for real-world experience, this book will help you codify your knowledge and prepare for the exams.

Windows NT in a Nutshell *By Eric Pearce*
ISBN 1-56592-251-4 \$44.95

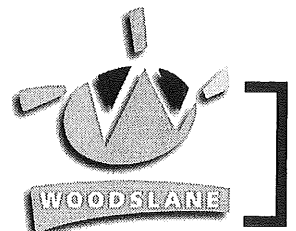
This book organizes NT's complex 4.0 GUI interface, dialog boxes, and multitude of DOS-shell commands into an easy-to-use quick reference for anyone who uses or manages an NT system. It features a new tagged callout approach to documenting the 4.0 GUI and real-life examples of command usage and strategies for problem solving, with an emphasis on networking. Addresses the single-system home user as well as the administrator of a 1,000-node corporate network.

Windows NT and Internet Administration

With O'Reilly's Windows NT technical books System Administrators can now become true Windows NT gurus, pushing the limits of NT's GUI. This series also includes the most comprehensive information available on managing Internet and intranet services.

O'REILLY™

or Available from all good bookstores
Direct from Woodslane on Tel: (02) 9970 5111
1800 686756 (Freecall outside Sydney) Fax: (02) 9970 5002

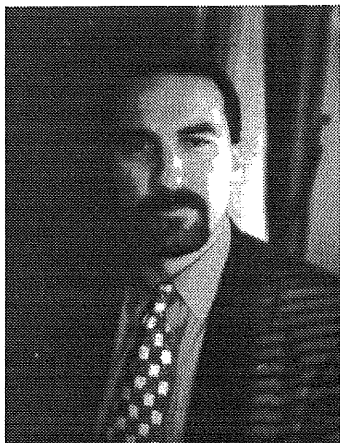


An interview with

...

**PETER VAN
DEVENTER**

*Server Marketing
Manager
Intel Asia Pacific*



[Editor's Note: during AUUG'98, Michael Paddon and Günther Feuereisen spoke with Peter Van Deventer, Intel's Server Marketing Manager, APAC, on Merced, Intel's future directions, and Unix on IA-32 and IA-64.

The following is an excerpt from that interview.]

Looking at Intel's 64-bit initiatives, what do you think it is going to deliver in real-terms?

IA-64 is currently the 64-bit computing platform you are going to hear more about. Merced is the going to be the first product that you are going to be released - the pioneer if you will. The current release date (for Merced) is the second half of 2000, having been pushed back by about half a year on its delivery date.

From an initiatives perspective, we are doing a lot of different things. Intel is working on a lot of initiatives with various vendors to promote a balanced platform - we don't just want to help make processors great, because if we don't work with everything else, such as I/O subsystems, we won't be able to exploit the power of the processor.

We see three major driving forces at hat are pushing the server industry and driving it forward:

First, we see the need for Standarisation and Scale - new products based on the new Pentium Xeon platform are providing performance levels that are staggeringly high - price performance levels that are unsurpassed.

As Intel moves upward into the Enterprise market, this in turn is driving a range of new opportunities for Intel, from the architecture perspective.

Second, we see the rise of the Internet - and the coming tidal-wave of e-commerce as being of major importance. The Internet is exploding, and e-commerce is going to explode. Yuu see a lot of transactions today, some are information some are

commerce, and the need for secure transactions is there.

That will drive a need for a lot of new servers into the industry.

And thirdly, we see the rise of computing in the emerging markets, such as Russia, China and Indonesia as being highly significant. There are a lot of new emerging economies which are implementing new infrastructures, getting connected, and providing a lot of opportunity.

These are the three main areas that we see as driving server growth.

Does this mean that Intel is going to be the new IBM of the marketplace?

No, I wouldn't say that at all. Intel is a building block supplier. We provide the fundamentals - building blocks, if you will. The silicon is the brains of that - it's the fundamental foundation that goes with it.

As the industry is evolving, we provide various levels of integration with those building blocks, to help those who want to buy, get into the marketplace, to compete, as aggressively as they want to do so,

Our concern is to make sure it is as balanced a platform as possible. as I said earlier, if it's just the processor that's getting faster and cheaper, then the whole Intel architecture platform is going to stall. So we work with a variety of players in the industry - Operating Systems, Application, Hardware and Peripherals vendors, to develop and introduce new technologies.

In that way we are more of a catalyst and stimulator of the platform. I don't see us trying to be the big fellow ..

What we do a lot of the time is try to remove obstacles from the path of the processor, such as the introduction of the PCI bus and I2O specification.

Today, from the industry perspective, there are very few companies who can afford to innovate. There are maybe two other PC manufacturers and a few more in the Enterprise who can afford the R&D to do that. What it means, is that you are back in the proprietary situation where only one or two people can afford to innovate, therefore we lose a lot of the dynamics that Intel brought to the PC market. A lot of people competing for the business, causes volume to go up, cost to drop and as a result price performance goes up as well.

That brings up an interesting question - Intel is by far the most successful chip manufacturer on the planet - can your competitors afford to compete? And if not, what is Intel going to do about that? Are we going to see a situation where Intel is in

anti-trust trouble? What happens if Intel is too successful?

It's expensive - it does cost in the billions of dollars to build the next generation processor. Intel is really cooking along very well, effectively developing, building and selling and working through that cycle of continuing the investment that it takes to do this.

Having said that, the number of dollars involved, currently there are three competitors in the marketplace, but the number of dollars involved is going to attract people - it's not something that people are going to ignore.

We've been very effective in our ability to make factory volume in supplying product and help developing the technology and introducing it into the marketplace, and our shareholders are happy to see that.

You're seeing standardisation in the Server space, moving into the Enterprise space. That standardisation is moving up from a one-way Pentium processor design, to the new 8-way processor designs.

We've had several vendors, including Hewlett-Packard and Fujitsu, talking about IA-64 and claiming preferred partnership and preferred platform. Also there have been claims that their Operating System will be the preferred OS on the IA-64 architecture. What is Intel's view?

From the IS view, Intel is Operating System neutral. Merced will be available to our customers the same way all our products have been previously available - based on their orders of the product and Intel's ability to fulfill those orders. We will supply Merced in the same way that we currently supply Pentium II and Xeon products today. Merced will not be tied to specific hardware vendors in any way.

From an Operating System perspective, we are encouraging working with all the major Operating System vendors, so that they have their operating system available as soon as possible.

We are not out to see any one perform better, or promote one or the other.

Does Intel have a special relationship with Microsoft? You certainly have in the past. The partnership between Microsoft and Intel has been historically very successful. Where do you see that going?

We see that continuing to develop, and our working relationship with Microsoft to continue.

The most important thing to face Electronic Commerce, is going to be the need for

cryptography. Authentication is very important. Is Intel, through Merced, addressing this issue?

Intel is working, not only with the Merced architecture, but throughout our entire architecture range - from top to bottom, making it more secure, allowing us to do what would traditionally require hardware dongles, or encryption. We are working with industry to make cryptography and other techniques more processor driven.

We are looking at automating the encryption process at the hardware level, not only with Merced, but also we will be seeing initiatives on the security side from Intel long before Merced becomes available.

What does Intel see its' timing (with respect to the Merced processor) - is it going to hit the server market first, and then try to propagate this technology into other areas, or is there going to be a multi-pronged approach?

The Merced processor will be introduced for the high-end of the workstation and server market. So both will benefit.

At this stage, there is no plan for IA-64 to be introduced to the desktop - as the existing base of 64-bit users is in the server/workstation market.

There is the perception that Intel architecture is PC hardware; not your Enterprise application server - more of a desktop machine, file server or games machine. What are your thoughts?

This is one of the fallacies we are looking at addressing - the fact that price-performance wise, IA-32 can compete with the Enterprise.

Intel is looking at addressing these issues, through the highlighting of UNIX and NT on IA-32 platforms.

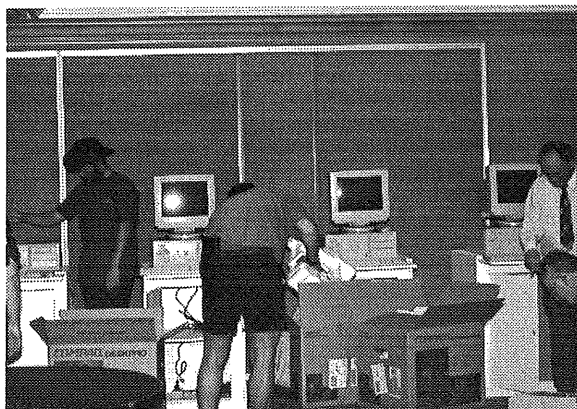
❖

On behalf of AUUG, we would like to thank Peter for taking the time to speak with us, and share with us some of the future directions Intel is taking.

Images of AUUG'98

Photos:
David Purdue
<David.Purdue@auug.org.au>

Comments:
Liz Egan
<busmgr@auug.org.au>



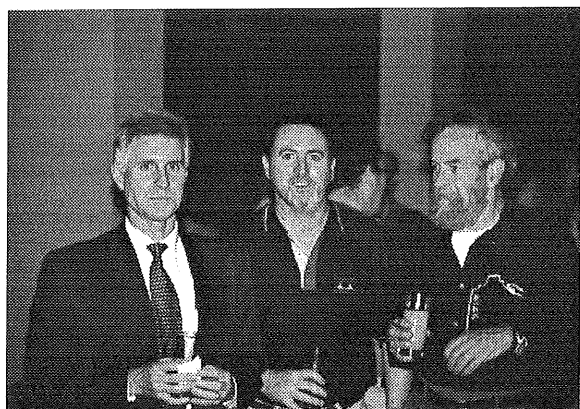
Setup for the big event commences ...



... whilst Lucy Chubb and Greg Rose get down to discussion at the Australian Museum ...



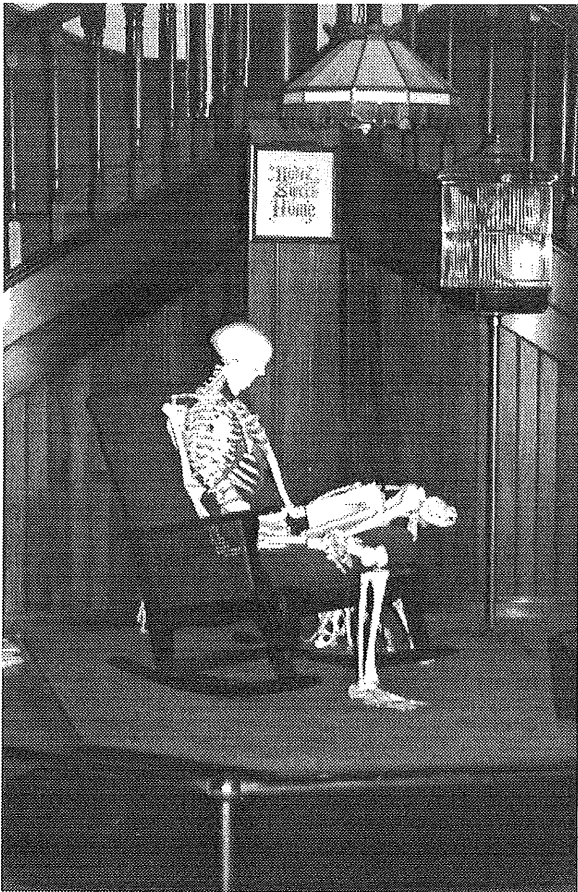
... boxes are unpacked, PCs set-up and stands built ...



Let the socialising begin! Phil McCrea and Andrew McCrae along with Robert Morris at the Speaker's Reception ...



Michael Paddon gets into serious competition with a skeleton ...



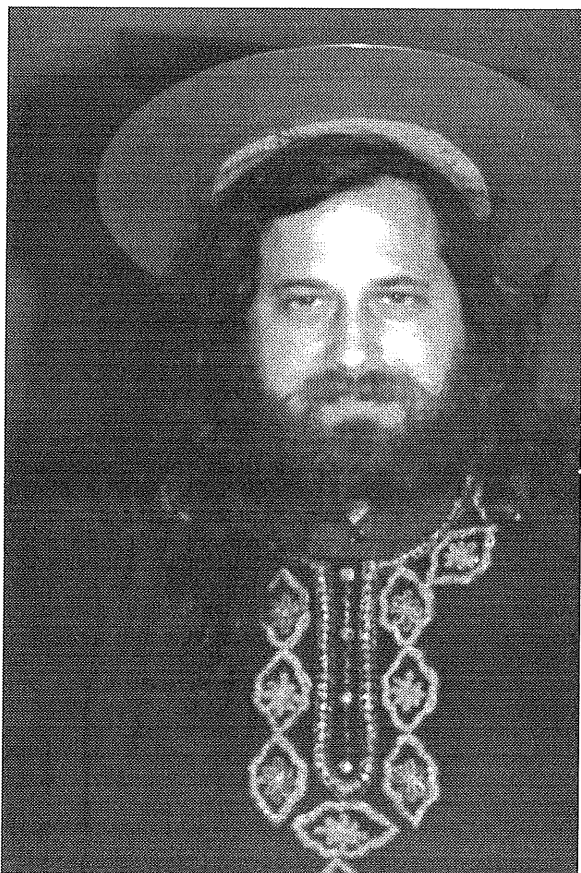
... while his mate takes time to check the conference program!!



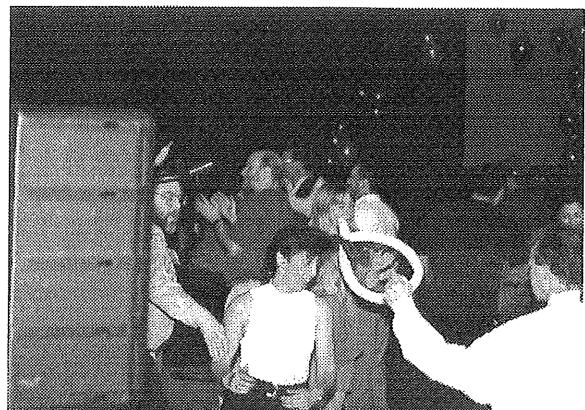
... and the E-Commerce panel continues their discussion ...



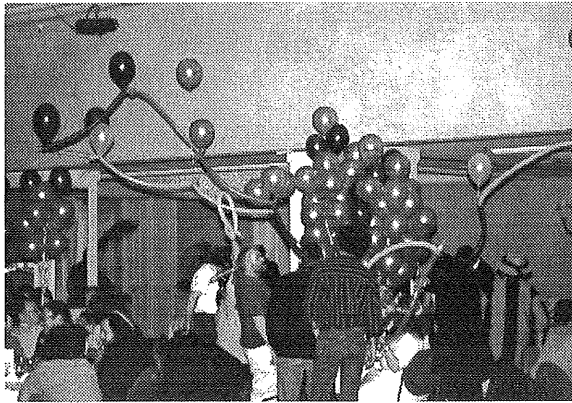
The evening saw everyone let their hair down at the conference dinner ...



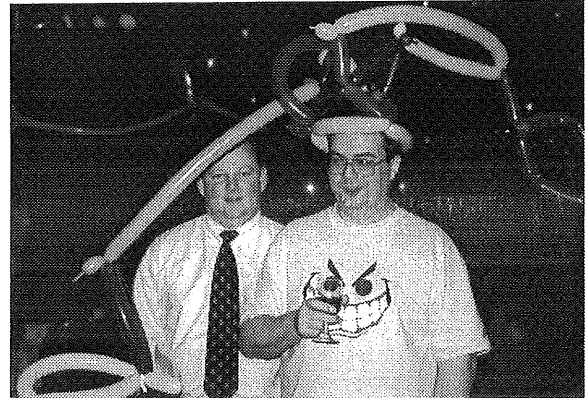
Richard Stallman comes dressed for the occasion, complete with halo ...!



... with Liz Egan appearing to be pulled in different directions ...!!!



For those technically inclined, the ballroom network took an a new dimension



... and imaginative ...



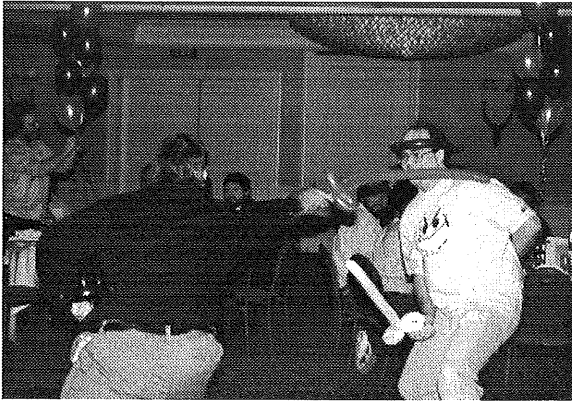
... whilst some were just happy to watch!!!



... in some cases, appearing to be more of a balancing act!!

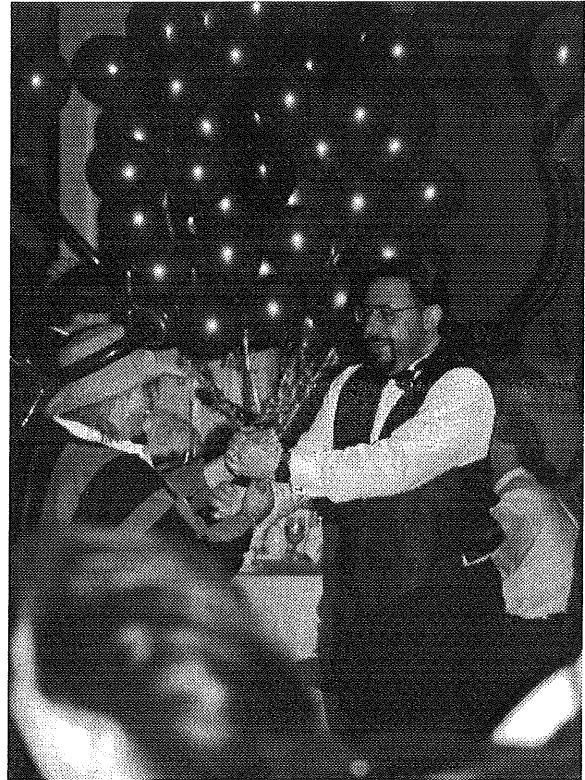


Headgear became very creative ...



... and what better way to end an evening with a "balloon" sword duel ... presenting Sir Greg of Rose and Stephen Boucher, Esquire!!

and finally, no photo essay would be complete, without special thanks to David Purdue, AUUG President, and keeper of the camera gear. Seen here, dancing with Julie Jester ..



See you next year!!



AUUG'99

It's only 10 months away!

AUUG '99 will be held from Wednesday 8th September to Friday 10th of September 1999 at the Carlton Crest Hotel in Melbourne. It will be preceded by two days of tutorials on Monday the 6th of September and Tuesday the 7th of September.

Put those dates in your diary now and start thinking of possible topics for papers. The call for papers will be in the next AUUGN.

Shoulders of Giants

A PAPER ON THE INEVITABILITY OF OPEN SOURCE DOMINANCE

Con Zymaris
<conz@cyber.com.au>

"If I have seen further it is by standing on ye shoulders of Giants." --Isaac Newton¹

ABSTRACT

This paper posits the concept that the open source/freeware software development and distribution paradigm, will eventually become dominant. It aims to show that this will occur as an inevitable process, slowly at first, then with almost critical-mass motion. A number of analogies to other areas of human endeavour, such as Science will be used to underline the power of the concepts behind open source freeware. Also, that the open source movement shouldn't be viewed as an attack against any single closed source vendor, but against the inadequacies of the closed source process. And finally, the hope is for this message to achieve some sense of resonance with enough readers, to add just a little more momentum to the accelerating adoption of the open source paradigm.

DEFINITION OF OPEN SOURCE

The term 'open source' is clearly defined by one of its foremost proponents, Eric S. Raymond². In essence, the basic tenets and philosophy of this development paradigm, are that software is developed and distributed in an open manner. The source code is provided. The software may be distributed freely. Others may extend the application, or derive from it, but always making their contributions available under the same conditions as the original software³. There are a great number of open source applications available. The ones that most people are aware of, are operating systems like Linux and FreeBSD, utility and systems tools like GNU, web servers (Apache), email transports (qmail and sendmail), development tools (Perl, GCC, Python, PHP) and many more.

¹ "What Des-Cartes did was a good step. You have added much several ways, & especially in taking ye colours of thin plates into philosophical consideration. If I have seen further it is by standing on ye shoulders of Giants." --Newton to Hooke, 5 Feb. 1676;

² The Cathedral and the Bazaar, Eric S. Raymond <http://sagan.earthspace.net/~esr/writings/cathedral-bazaar/cathedral-bazaar.html>

³ The Open Source Definition <http://www.opensource.org/osd.html>

Why is it important to consider whether platforms built with the open source method will become the new industry staple? In essence, by showing this is a likely inevitability, this will help it become a self-fulfilling prophecy. We know this method to be a powerful and effective technique, as first IBM then Microsoft have used to it great effect over the past three decades. Since the late 80's, this tool for market penetration has been wielded, to give Microsoft dominance in an increasing number of industry sectors. In short, the reasoning goes: Why compete, when Microsoft will always win? Commercial software vendors, IT consultants, resellers, and developers, all consider the same issues in response. Developers think: If I write software, it may as well be for only Microsoft platforms. They always win. If I write for the MacOS, BeOS or OS/2, I'll have a fraction of the market. IT consultants think: If I learn another platform, and interface, I'll lose out. The world is going down the Microsoft-everywhere path. ISV's think: Microsoft has conquered all 100, 50, 10 and 1 million unit application domains, and is expanding downwards into my product space. I don't have a chance competing. I'll keep winding my way down to smaller and smaller niche domains. All these quite understandable musings arise because people think that Microsoft, with its technology flight-path, is unassailable. This may have been the case with all commercial competitors in the past, but now there is a new player in the game. Not a technology or vendor as such, but a new mode of thought. Open source.

What makes open source such a strong contender for the new titleholder of `_the_` software development paradigm, is its openness and peer review process. Why is this so important? What makes this such a powerful approach? The best response I can think of is to draw an analogy with another area of human endeavour, Science.

OPENNESS OF SCIENCE

Science, in its clearly understood modern guise, is unique. This essentially Western tradition of open inquiry is believed to have developed only one instantiation throughout the whole period of human history. While almost all human societies have developed language, art, and music, open inquiry into the natural and philosophical world sprung only from the eastern rim of the Mediterranean sea, in a number of ancient Greek states, approximately 27 centuries ago. Helped along by the advantages provided by the recently formulated Greek alphabet, the people of this region bought forth the makings of the primary conceptual and philosophical machinery that was necessary to develop an understanding of the nature that surrounded them. In short order, they had conceived ideas which led them to believe that the Universe was understandable, that it was measureable and that it could undergo rational analysis. The philosophical re-conceptualisation of the Universe had its eventual pinnacle in the works of Plato, who introduced us to the sublimely powerful concept of

'Forms'⁴. In Platonic terms, we find both the physical, everyday world in which we exist, and the world of absolutes and eternals; the Forms. Using this machinery, scientists and philosophers have been able to visualise generalisations in their 'idea space', and not merely the imperfect incarnations that exist in real, physical space. Through the advantages bestowed upon the Greeks by virtue of their written language and undoubtedly their open, democratic political environment, these concepts spread. Where once these people would have been manipulators purely of the physical world (pottery, sculpture) they now also became masters of symbolic manipulation. Mathematics, logic, geometry, geography, mechanics, hydraulics, medicine, architecture, astronomy and cosmology, optics and dozens of other disciplines flowered. There has never been a similar period in human history, with the possible exception of the 18th century 'age of reason' the Enlightenment. Even though the power of the ideas from ancient Greek science seem to us obvious, they were patently not so, as even after their demonstrable successes, they were relegated to the recess of history for almost a thousand years, after the fall of Rome and of the great library in Alexandria. Luckily for all of us, these same ideas were storehoused and enhanced by the Islamic world. Eventually, after the fall of Muslim-held Toledo in the 11th century these same ideas resurfaced slowly but steadily, into Western and Southern Europe, to precipitate the Renaissance; the re-birth⁵. The arrival of these Greek texts coincided with the development of the university as a legal entity with political and intellectual autonomy⁶. Once again, the openness to new forms of thought, the cheap, efficient and accurate transmission of ideas through the wonderful machinery of technology (Gutenberg's printing press) brought forth an explosion of creativity and propelled Western civilisation forward. Open source software is a direct descendant of this culture of thought, as it prizes the same properties and philosophy which form the basis of the driving force of science.

Openness is thus one area of the scientific process which is of interest for comparison with open source development. While sizeable tomes have been written about the methodology and philosophy of this facet of science⁷, things generally boil down to the following: part of science is a process of verifying or culling hypothesis, and is in essence an open and self-correcting system. Because of this, progress occurs at a much faster rate and in a more dependable/trusted fashion. This doesn't mean that the self-correction happens in minute, continuously flowing 'chunks'. In

⁴ The Republic, Plato. Trans. H.D.P. Lee, Penguin, 1955.

⁵ The Day the Universe Changed, James Burke, Little Brown & Co, 1995

⁶ The Rise of Early Modern Science, Toby E. Huff, Cambridge University Press, 1993

⁷ The Logic of Scientific Discovery, Karl Popper, 1959 (English Transl.)

reality, corrections arise as mini-revolutions, characterised by philosopher of science, Thomas Kuhn⁸, as paradigm shifts. Nonetheless, over longer periods of time, progress does occur. In many ways, this progress is accidental, as there is often no 'vision' or nomenclature to describe where science is heading, until after it has arrived.

The speed of progress is greatly enhanced by virtue of the fact the practitioners of science publish not only results, but methodology, and techniques. In open source terms, this is equivalent to the source code. This not only helps 'bootstrap' others into the field, to learn from the example set, but makes it possible for others to verify or refute the results (or techniques) under investigation. In an almost guided Darwinian evolutionary fashion⁹, this makes the scientific process a powerful tool for the highlighting, analysis and possible culling of ideas and concepts; less useful ideas and hypothesis die, and likely contenders come sharply into focus. Newton made his famous comment, in part, to indicate that his contributions to the human knowledge could not have been achieved solely. He needed the 'firmament' beneath him hypothesised, tested and confirmed by generations of scientists, philosophers and thinkers before him, over thousands of years. With science, in the medium to long run, all other issues fall by the wayside, and merit alone is the main attribute of the victorious memes¹⁰.

OPEN SOFTWARE CONSTRUCTION

By analogy, making the source code available for peer review and extension, is perhaps open source's most powerful advantage. Besides the verification provided by peer review, refutation for issues such as security is possible, learning of techniques by new practitioners is a great advantage, as is modification and redistribution of the code under similar conditions. It makes it possible for anyone who has a background in, or can acclimatise to the technology and skills required, to continue development, extending the code into whatever direction that they need. This, in turn can be plowed back into the original system, causing eddies of strengthening feedback. This results in an ever growing base (or firmament) of quality code, upon which more and more programmers can benefit from, and contribute back to.

By comparison, multiple generations of closed source software can be seen as an unfortunate process of re-inventing the wheel. As can be imagined, this can be an excruciatingly slow process¹¹. The open

⁸ Structure of Scientific Revolutions, Thomas Kuhn

⁹ Darwin's Dangerous Idea: Evolution and the Meanings of Life Daniel C. Dennett. Simon & Schuster 1995

¹⁰ Richard Dawkins, The Selfish Gene.

¹¹ Guns, Germs and Steel, Jared Diamond, Vintage 1998

communication of source and ideas, while progressing slowly at first, builds momentum as more and more practitioners learn from, extend or revive from obscurity, more and more code. The success of this form of interconnected web of knowledge has been analysed by historian of technology, James Burke¹²)

Another facet of similarity between open source and science, is the respective cultures. Both are strongly technical, perhaps verging on the geeky; both are meritocracies. A large part of the impetus of the scientists is the applause of others. Peer recognition is also perhaps the single most important¹³ reason attributed by open source advocates and developers as to the reason why they pursue open source methods of software distribution.

ADVANTAGES OF OPEN SOURCE

An excellent general introduction to the advantages of open source are covered by Raymond¹⁴. I'll try to make some succinct cases for other possible advantages. Specifically, these are:

- the advantages of not re-inventing the wheel, time and again. Chances are, that somewhere, sometime, someone has written the code you may need to perform the functionality you want. Why spend your time recoding? If you abide by the GPL, you can search for that tool/code/package using the wonderful medium known as the web, and get your application finished faster, and if you choose a base package well, with more likelihood of success
- the advantages of training students with open source material. These utilities, compiler, interpreters, and operating systems go well beyond the simple or theoretical models of development used so frequently in most universities. They can also show the advantages and dis-advantages of various development tools and languages, in real world situations, with the ability to dissect and scrutinise
- now that we have standardised, quality development tools, on multiple platforms, and we have honed many methods of well tested forms of software construction, there is less risk of producing code which will not be around in 20 years. Efficiency will improve. open source tools and libraries will improve and mature. more and more features will be made available. This can be visualised as the slow buildup of

¹² Connections, James Burke, Macmillan London 1978, 1995

¹³ Homesteading the Noosphere, Eric S. Raymond <http://sagan.earthspace.net/esr/writings/homesteading/homesteading.html>

¹⁴ The Cathedral and the Bazaar, Eric S. Raymond <http://sagan.earthspace.net/~esr/writings/cathedral-bazaar/cathedral-bazaar.html>

code over many years, by more and more people, to form a great platform for newcomers or new projects to catapult from; in effect, the 'shoulders of giants'

- Unix. Why base the short term future of Open Source on Unix? At the risk of alienating users of other operating system platforms, I believe that the openness and portability which has been part of the ethos of Unix from its start, and the fact that it has always had an aura of 60's counter-culture, make it the primary contender to accrete the necessary attributes for dominance through open source development. Due to its simplicity and design philosophy. Unix (or Unix-like platforms like Linux) have substantial advantages over most competing platforms. It is perhaps the most clearly understood of operating systems. Detailed source analysis and commentary for Unix originated in the '70s with John Lyons (University of New South Wales, Australia.) This allowed generations of systems developers to learn and understand. In many ways, the open source movement follows on in his tradition. With Unix, pieces can be made to work together in a simple way. It has matured; is solid; dependable. From a user interface and ease-of-use perspective, it can be made to appear as low-tech as resource requirements mandate or as glossy as anything else available. The important thing is choice. For a more detailed examination of the power of Unix, see the paper by John Kirch¹⁵
- at this time of writing, numerous commercial vendors are building systems software comprised of tens of millions of lines of code. News coming to the fore indicates that the products being developed are running months, sometimes years behind schedule, even with the vast development budgets being bestowed upon them. When (if?) these systems get released, they will require a multitude of patches (service packs!) for possibly thousands of problems and security flaws which appear after the years of beta testing already performed. The path thus taken by commercial vendors, whose primary motive in producing software is profit, is leading towards greater code complexity. This in turn is resulting in the same problems bloatware brings to desktop productivity applications. Features keep piling into operating system kernels and development tools, making them less and less likely to be robust and secure, and requiring of more and more hardware resources just to load. In the end, this is the only way that commercial vendors can sell more and more upgrades. If all they did was patch bugs, how could they charge the exorbitant upgrade fees which help keep their stock prices up? Open source platforms, by having the profit motive removed, tend to produce leaner, more technically pure systems software. There is less a

¹⁵ Microsoft Windows NT Server 4.0 versus UNIX <http://www.unix-vs-nt.org/kirch/>

tendency to have an 'include everything, even the kitchen sink' mentality, because profit from upgrades isn't an issue. Fixing bugs, and improving performance is

- open source operating system are a great platform upon which to base many of today's data processing and communications solutions, due to the fact that they come complete with (often) hundreds of tools and utilities, scripting systems and trouble-shooting apps. It would take months of time and considerable expense to replicate this wonderful functionality on commercial systems. The automatic assumption that these tools and scripting systems will be available on a computer system means that developers can greatly reduce their effort in producing solid, working solutions.

IT'S TIME

There is one thing stronger than all the armies in the world, and that is an idea whose time has come. -- Victor Hugo

Why is the current period the 'coming of age' of the open source paradigm? Why, if this idea is so good, if indeed it will become the pre-eminent development process, didn't it arise 20 years ago? What follows is some analysis which hopefully shows that while the open source idea has resonance with many people, lack of facilities curtailed its progress, much like the spread of ideas and technology from the ancient Greeks was withheld from most ordinary people prior to the Gutenberg press. With these facilities now in place, the full power of the paradigm is quickly becoming apparent, and it is already bearing fruit.

The current growth in open source usage and recognition could not have happened 20 years ago. There are probably a number of synergistically contributing factors for this, including the advent and widespread use of the Internet, maturation of quality development tools (mostly of open source origin once again) and the slow and steady spread of the open source meme.

One of the pre-requisites for the current progress of open source systems, is the quantity and quality of development tools on offer. Back when Richard Stallman, perceived by most as the instigator of the modern freeware movement, began writing the GCC compiler, there was a relative scarcity of quality development tools. There were also few well accepted and standardised programming languages which could be used for the rapid development of systems and applications. Many of the emerging languages and methodologies of the 1970's (C, Pascal, Smalltalk), were still relatively immature in both design and implementation. By comparison, now we have a multitude of powerful, well conceived development tool modes (C, C++) and scripting platforms (Perl, tcl, Python.)

Another reason why the open source revolution is occurring now, is due to the Internet. The Internet makes possible numerous processes which have been essential for the growth of the open source movement. Among the major advantages, the Internet makes for a wonderful accelerant of code and idea dissemination. This includes spreading the idea that open source is good. To some extent, this idea is subversive of the current closed and proprietary paradigm, and it's a message that traditional IT magazines wouldn't have pushed too far, for fear of potential revenue losses. While commercialisers of open source products are also advertisers, they are not in the same spending league as the major commercial vendors such as Microsoft and Lotus. With the effective self publishing available on the Internet for everyone (witness Slashdot¹⁶, Linux Weekly News¹⁷ etc) the mainstream IT trade press have no option but to follow suit, even if it does mean revenue reductions. Those that do so and succeed (LinuxWorld¹⁸) may have lower revenue than their more commercially focussed competitors (ZDnet) but they greatly enhance their reputation amongst the industry's technological elite (geeks, hackers, IT technologists) and thus have higher overall influence. It is more often than not, these individuals who slowly rise to the upper echelons of our industry's leading firms, and most interesting startups.

Perhaps the biggest factor in the current crescendo of open source growth is the use of the Internet as the medium through which the mechanics of distributed software development and testing is made feasible. Prior to widespread developer use of the Internet, bulletin-board services (BBS) were the main vehicle for idea and code dissemination of the technomasses. Modern software construction techniques, involving the use of Network File System (NFS), Concurrent Versioning System (CVS) and GNU's tools, have made the Internet far more appropriate for this task.

Yet another reason for the current surge in open source acknowledgment, is improvements in hardware. One of the oft repeated observations of the information technology industry is the rapid improvement in computer hardware performance. This is another major reason as to why the past few years have seen an increase in usage of powerful, Unix-like operating systems. The 32 and 64 bit CPUs, high performance large disk space, and the memory needed to run a modern Unix, or Unix clone, weren't available to the types of people who have fuelled the open source movement until recent years. Ironically, the mainstream OS (Windows) has provided what in effect, may be the vehicle for its own eventual demise, through its bloatedness and

¹⁶ Slashdot -- News for Nerds. Stuff that matters. <http://slashdot.org/>

¹⁷ Linux Weekly News. <http://lwn.net/>

¹⁸ LinuxWorld <http://www.linuxworld.com/>

therefore escalating hardware needs. Merely to tread water, hardware systems have simultaneously advanced and dropped in price, benefiting the first few waves of open source users and developers, such as hackers (not crackers!), students, IT professionals, and industry technology-innovators.

EXTRAPOLATIONS FROM HISTORY

"Prediction is extremely difficult. Especially about the future." -- Niels Bohr

How do we define 'dominance' in the information technology industry? We need some gnomic measurement of what technology or process is dominant now and how it got to achieve this status, and by what means the open source model will overtake it. There are numerous case histories that may assist in our analysis. For this exercise to be illuminating, we must focus on specifics and details. My hope is to show both how closed source systems have been able to capture various IT markets, and why these techniques will not succeed in keeping the open source paradigm at bay. While the text that follows refers to proprietary vendors when invoking specific market dominance cases, we must keep in mind that the intention is to show that the closed source model is at problem, and not specific for-profit organisations.

Our first example, is the way in which Microsoft's Windows holds the dominant position for the desktop. Windows was never the best technical solution. It wasn't the easiest operating system to use. Why did it come to control 90% of the desktop and low end server market? To determine why, we need to contrast the competitors. Windows beat Apple's MacOS because the MacOS didn't run on cheap, commodity hardware. An Apple Macintosh was and will always be 30-50% more than the cost of a comparable low end x86 clone. Apple will claim that Macs are no more expensive than Compaq, Dell or other top tier name-brand PCs. That's not the point. Having access to very low end, low cost systems is the point. These do not exist in the Apple Mac space. And as Independent Software Vendors (ISV) write primarily for the 'mainstream' operating system, this translated to more and more x86 based Windows apps. Furthermore, the main advantage the Macintosh had, was its wonderful simplicity in installation and operation. It is perhaps the best platform for many 'end-users'. This simplicity however, can begin to cause problems for 'power-users' and technophiles. An analogy can be made that the MacOS is like a bike with immovable training wheels, often frustrating more experienced users. This combination of not targeting the 'mass-market' through early OS licensing by Apple, and by being snubbed by the tech-innovators, left the MacOS forever in the shade. As Apple's overall market share slips, so do the number of ISVs coding for the platform, so do the platform's chances of survival.

IBM's OS/2, had, from its inception, a gloomy associated aura. It too was technically superior to Windows. It seemed to be the OS that people either loved, or ignored. Most ignored it, regardless of the tens of millions that IBM pumped into marketing. If only IBM didn't, at that time, still possess its lingering, unpleasant, monopolist odour, it may have had more people warm to OS/2. Also, by virtue of being the underdog in a race with Microsoft, people felt ill at ease in buying into a technology that Big Blue might eventually abandon. This of course, has almost become a self-fulfilling prophecy, with IBM pushing NT solutions now more than OS/2. OS/2's fate will not be the same of open source operating systems like Linux, due to the fact that users who 'buy into' the Linux platform know it will *_always_* be there; there's no bean counting manager in corporate entity in Linux's 'parent company' who'll be able to pull the plug, orphaning millions of users. By way of contrast to the slowly dimming fortunes of both OS/2 and the MacOS, Linux had more installs in 1997 than both of these two platforms combined. Impressive, considering the amount of marketing muscle that IBM and Apple can muster.

Once again, one of the reasons that Windows achieved dominance, is that it is the operating platform that most of the young up-and coming tech-leaders used 7-10 years ago. Unix workstations were too expensive, and Microsoft was not as unpopular amongst their number now as it is today, IBM was held that dubious honour. These individuals, through learning on the Microsoft *_pastures_* were most likely to follow down the Windows path in their ensuing careers. Slowly, this group has migrated from Windows to Linux and other open source platforms and Microsoft is now loathed more than IBM was twenty years ago. As happened in the early Nineties, these tech-leaders are moving into the mainstream, bringing their hacker culture with them, and their insistence on more capable, powerful and open platforms. As for examples of this type of event, witness the growth of Unix into the corporate enterprise space. Twenty years ago, suggesting Unix to mainframe sites seemed a career-limiting move. Now, Unix is the entrenched enterprise player, in part because the generation of 70's and 80's hackers eventually became the MIS heads, Similarly the explosive growth of the Internet shows another example of this process. Ten years ago, most corporates would have eschewed the Net as something of a toy for bearded hackers at universities to fool around on. Five years later, when everyone realised that this is perhaps the biggest improvement in mass communications since the printing press, everyone wanted in.

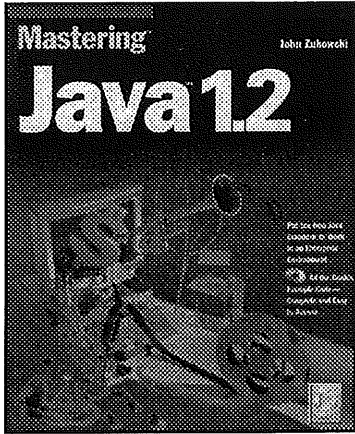
ALL OPEN VS. ALL CLOSED?

It isn't necessary for all development to follow the open source process for open source to become the dominant development paradigm. In fact, there *may*

New

Addison
Wesley
Longman

from Addison Wesley Longman



Mastering Java 1.2

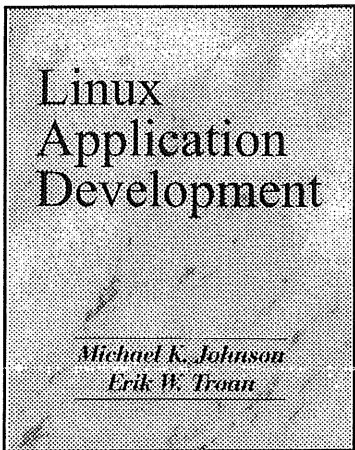
John Zukowski

Mastering Java 1.2 provides the most in-depth coverage of JDK 1.2, enabling the reader to take immediate advantage of the many new and improved features of the latest standard from SUN.

Inside this book there's ample discussion of the basics—with everything that makes Java ideal for simple desktop and Internet applications. Experienced Java programmers will benefit most from the detailed coverage of what's new, including the many capabilities that make Java 1.2 ready for true enterprise development.

ISBN: 0-7821-2180-2

\$99.95 w/CD-Rom



Linux Application Development

Michael K. Johnson and Erik W. Troan

This practical reference guides programmers developing Linux applications or porting applications from other platforms. Linux is fundamentally similar to Unix—therefore, much of the book covers ground familiar to Unix programmers—but this book consistently addresses topics from a Linux point of view. The aim throughout is to provide the detailed information you need to take full advantage of Linux.

ISBN : 0-201-30821-5

\$99.95

AUUG Members receive a 20% Discount off the RRP

order form

To order a copy of these texts, please fill in the details below.

This form should be faxed or mailed (freepost) to Cassy Nacard, Trade Marketing Coordinator at Addison Wesley Longman, Level 1, 2 Lincoln Street, Lane Cove NSW 2066. Ph: (02) 9428 8086 or Fax: (02) 94279922

Book Details

Author/Title	ISBN	Price
_____	_____	_____
_____	_____	_____
_____	_____	_____
Total less 20%		_____

Order details

If you are **purchasing** the books detailed above, please give the following information. (Please note that books can also be ordered directly from bookstores.)

Book(s) price: _____ Note: Prices are subject to change without notice

Postage: 1 book = \$4.00, 2 or more = \$7.00 Total \$ _____

I enclose a cheque made out to Addison Wesley Longman

Please charge my credit card:

Bankcard Visa Mastercard American Express ID No: _____

Card Number: _____ Expiry Date: _____

Signature: _____

Your Name: Mr/Mrs/Ms _____

Address: _____

Daytime Ph: _____

be many classes of applications for which closed source, commercial development is the only available path. This will have to be determined on a case-by-case basis, in an organic way. What does seem apparent, however, is that much of what may be called 'infrastructure', namely operating systems, networks and services, technical and development tools, are the prime candidates for being developed with open source methods.

WEAKNESSES OF OPEN SOURCE

To help us ensure the endurance and growth of the open source paradigm, it is important not to avoid discussion of its weaknesses. Among these are generating revenue to ensure core source coders continue their work, ego battles, and the possibility of fragmentation of standards and protocols.

While much of the development of most open source projects is performed by a myriad of people, generally one or a small group of these are tasked with compiling the disparately generated code and bug patches, quality assurance and then checking it into the primary source code tree. More often than not, these people devote a non-trivial part of their time to his endeavour, sometimes enough for it to have a negative impact on their personal finances. To counter the problem of these individuals moving on from their open source projects due to time constraints, it is important that methods of ongoing financial return be looked at by the open source community. Some work has already been done by various groups to this effect. Also, many of the commercial 'distributors' of open source systems are actively funding core developers, ensuring the continued steady development of important pieces of technology for future open source platforms. Still, more consideration needs to be given to the issue of monetary recompense for the developers of many non-essential open source applications and tools. Some will indeed be able to charge for customised releases. Others through the provision of commercial grade support. Some can earn through the production and printing of manuals and media. Others may be lucky enough to join a company who bundles their open source with hardware or services, and thus sees an advantage in having them on-staff. Not everyone will fall into these groups, so more options are necessary. Some possibilities lie with variations to the main open source licences, such as the one used for the Ghostscript application by Alladin. Alternatively, peer review groups, which can develop along the lines of national science funding bodies, could be organised. This would assist in guaranteeing ongoing development of important open source applications. Funding for these bodies could come from an increasing number of commercial organisations who have witnessed and become convinced of the advantages that open source bestows upon them. Much like large corporations (mostly users of information technology, not necessarily IT firms themselves) helped fund the

movement to open and interoperable systems in the early '80, these firms could be harnessed to fund open source in the new millennium. While in recent times, governments generally avoid 'meddling' in areas of commerce, a case could be made that fueling the development of open source platforms is in most governments 'national interests.' Stated advantages are multiple-source procurement, cost reduction and peer-review security analysis etc. If this idea can be heralded, argued and won, it may be possible to enlist the governments for minority funding, just as they do with science, research and development. This isn't a far-fetched concept. Government bolstering of open source software already happens indirectly through the government funded universities. Just as government funded scientific organisations worked with corporate 'commercialisers' of the R&D output, to help create the 'Computer Era', co-operation with peer-review and standards bodies can help propel whatever follows. While, in the end, the eventual outcome will be the same (open source dominance) this will help speed the process.

The issue of 'ego' in the software world has caused a large slice of the problems we have faced, in terms of closed protocols, wheel re-invention and the Not-Invented-Here (NIH) syndrome. The same issue has helped generate the problems which have kept Unix's market share un-necessarily low. More recently, in the open source world, they have threatened to 'balkanise' Linux through disparate standards projects, and had the potential to create 'tribal' conflict between the various operating systems (Linux, FreeBSD, NetBSD etc.) Luckily, it seems that we are starting to learn from the tactical and personality mistakes of our predecessors. People like Eric S. Raymond and Jordan K. Hubbard (a core FreeBSD developer) are articulate and passionate in their insistence of the open source community avoiding these un-necessary skirmishes. We, as a community, see before us, an opportunity to assist in the production of an open infrastructure for future global computing and communications; an infrastructure where the platforms are as robust as they can be, the protocols are open and clean and the applications can communicate with one-another through common document standards. The IETF and W3C are working hard towards this end, and the open source community should work with them in ensuring freely available implementations for all to use. For some interesting comments on the importance of open protocols, see this paper by Raph Levien¹⁹. There is much more at risk now if we miss this chance, than ever before. This, perhaps more than anything else, will likely help keep the group moving forward, if not in lock-step unison, at least in the same direction.

¹⁹ The Decommoditization of Protocols, Raph Levien <http://www.levien.com/free/decommoditizing.html>

ADVANTAGES TO HARDWARE AND SOFTWARE VENDORS

What advantages are there to vendors in helping usher this new era of open source software development? This is a complex question, and the answer depends on what part of the information technology hardware spectrum the vendor is situated. What follows below is a rough breakdown of the industry into 3 separate segments, which are analagous to the home/small-office, home-office (SOHO) market, the corporate desktop and the departmental server market. If we can assist these vendors visualise the future with open source systems software, and the advantages this will bring, we may indeed accelerate the migration.

Perhaps the biggest advantage that will eventuate for all vendors when open source software becomes the norm, is the rather philosophical notion that no single vendor will have any pre-tournament advantage over them. Presently, this is not the case. Some vendors, in both hardware and software, have sometimes subtle, sometimes acutely obvious advantages over others. This arises by virtue of the advantaged vendors owning the intellectual property (and closed source code) upon which the disadvantaged vendors depend. Business forces sometimes call upon tactics which invoke these advantages from time to time, leaving an unpleasant taste for many players. Migration to open source platforms would remove all these uneven playing fields, for both hardware and software vendors.

It is obvious to most technologists that for a technology to take off and become mainstream, there needs to be a rationalisation of the formats upon which the industry is splintered. When this happens, consumers are happy to buy that which everybody else buys, economies of scale are introduced into the equation, and everybody wins. Consumers purchase at lower costs, vendors sell higher volume. This is where another advantage of open source software can be invoked. It is reasonably obvious that a large part of the splintering in the information technology industry occurs due to the 'Not-Invented-Here' syndrome. One vendor (rightly) believes that if they use another vendor's systems software, or build to another vendor's Application Programming Interface (APIs) they are at a tactical business disadvantage. Of course, each vendor thinks this, and thus, the number of systems platforms and APIs keeps increasing, and more fragmentation occurs. By convincing a large portion of these vendors to start using a fully-fledged open source operating system, in lieu of their current proprietary or 'imported' operating system, we can remove the basis for the disquiet over tectical advantages. Perhaps all these vendors will be happier to use this one open source platform, as they know that their competitors will never have any advantages over them. Perhaps this platform, and its associated APIs can be the one

uniting factor which ushers in an era of greater mainstream acceptance, mass production of systems and lower costs. Perhaps this one platform can be the equivalent of the VHS videotape, or the compact-disk. The vendors can then get on with increasing the functionality and performance of the systems and software, safe in the knowledge that their business will never be at the mercy of some competitors tactical manoeuverings ever again. We would all therefore benefit. This topic is covered in greater detail below.

The immediate advantages to vendors in the home/SOHO sector are cost reduction for each unit shipped, and likely reduction in post-sales technical support. As the hardware costs associate with personal computers keep decreasing (at the time of writing, an entry level PC was around US\$500,) the ratio of closed source, proprietary operating system software to hardware cost for each shipped unit is getting higher. It is now estimated to be around 25% of the cost of the unit. This is perhaps the single biggest component cost, and all PC vendors that address this market segment have no way to reduce this cost, if they stick with the current commercial operating system. By switching to an open source operating system like Linux, they would instantly remove this largest single cost, allowing them to either severly undercut their competitors, or make considerably higher margins. While some may argue that open source operating systems aren't ready for prime time home usage, this argument is fading very quickly, with projects like KDE²⁰, GNOME²¹ and Wine. On the subject of reduced technical support, the general consensus amongst systems administrators¹⁵ is that open source operating systems have considerably lower ongoing support and maintenance costs than the types of proprietary operating systems presently in use for home/SOHO personal computers. This would greatly reduce post-sales support and would put whichever vendor shipping systems with open source operating systems at a great advantage over rivals. Another less tangible advantage is that through using an open source operating system, there is immense scope for customisability. Opening screens, icons, backgrounds and screensavers can all be configured to help either sell the product, or for useful post sales marketing.

For vendors targetting the corporate sector in personal computers and small servers, the benefits are greater security and performance, substantial reduction in lost productivity due to system faults, and reduction of licence fees for server operating systems. The points raised have been dealt with throughout this paper; also see¹⁵. In this market segment, customisability is a substantial advantage over rivals who can only ship what their closed source operating system vendor dictates.

²⁰ KDE - The K Desktop Environment
<http://www.kde.org/>

²¹ The GNOME Project <http://www.gnome.org/>

For vendors of high-level server systems and workstations, who often ship their own operating system rather than licence another software publisher's OS, the advantages are quite different. Instead of pursuing ongoing and very costly internal closed source operating system development, they should accept as a given the inevitable advantages and dominance of open source operating systems, and begin working with the open source community to introduce these operating systems to their hardware platforms and processor families. These vendors can then concentrate on adding value to this base operating system by building extensions for your specific platform, without breaking compatibility.

THE OPERATING SYSTEM THAT ALL VENDORS TRUST (OR THE TOM BOMBADIL FACTOR)

Throughout most of the modern era of information technology industry, it has been taken for granted that one vendor or another must be the standard bearer, near monopolist industry leader, to force direction and establish defacto standards. All known cases have shown that this vendor, eventually, becomes corrupt with the power they have to wield. An excellent cultural reference point re-counted to me recently, is that of Tolkienesque Ring. Whoever possesses the Ring will eventually be seduced by its power. It can be argued, however, that if a vendor became bad enough, they would be toppled by competitors. This does seem to occur, but there is considerable cost in this cyclic swing from monopolist vendor to monopolist vendor. The IT industry is quite competitive, and new contenders in the market are entering every month. While the cost of entry into the industry is not high, it takes a monumental effort (verging on the near-impossible) to wrest control from whoever the current monopolist vendor is. Trying to escape from vendor specific application and architectural lock-in, is equivalent to attempting escape from a black-hole.

Many industry observers believe that operating systems fall within the domain of 'natural monopoly,' where one platform slowly begins to dominate over others, edging them out of the market as more and more users flock to its 'mainstream.' A close comparison can be made for other industries where 'software' needs to be made to operate with 'hardware.' Video recorders and tapes, CD players and CDs, tape-decks and audio tapes are all clear examples. The 'playback' unit is the hardware and operating system while the music is the software. With this tendency towards natural monopoly, it is best for all vendors to converge upon an operating system platform (and infrastructure) that no single vendor can control. Industry-wide acceptance and use of open source software, in both its infrastructure guise and as technical tools, is perhaps our only

chance of forever breaking the monopolist vendor vicious cycle. Extending our Tolkienesque analogy, open source methodologies and output, can then be thought of as Tom Bombadil¹⁹, the character who was impervious to the Ring's seduction.

Consider the following possible scenario. An open source operating system (this could be any of NetBSD, FreeBSD, Linux etc. but for grammatical simplicity, lets choose Linux for example) could become the basis or template for all the major vendors to produce hardware for. Linux could then be available with a near complete set of features for a standard operating system. If any operating system becomes standard, vendors have to find a way to differentiate their products from their competitors. While this is difficult to do with current closed source operating systems, it is almost trivial to do with open source OSs. For example, A major server manufacturer, could extend the base Linux OS with high-availability features/drivers which are specific to their products. While the code may need to be re-released into the open source 'pool,' it will not be of great advantage to their vendor rivals, as they will most likely be using different hardware designs and architectures. Linux, with its wonderful chameleon-like capacity to be all-things to (nearly) all people, would make an excellent choice for the greatly disparate needs of different vendors and industry segments. From the looming thin-server market, to desktops, sub-notebooks, server (small to large) and super-computer clusters, there is much that it can offer. By comparison to all closed source platforms, Linux lends itself to great customisation, as has been shown by its hardware mobility. It is available on a dozen very different hardware platforms. Its closed source rivals can manage at most one or two. As computer technology extends into increasingly non-obvious areas (wearable computers, car-mounted systems, industrial devices, remote-sensing systems etc.) this flexibility of open source is not only an advantage, it becomes mandatory, else progress retardation will occur.

As more and more vendors come to adopt an open source platform upon which to build solutions for their customers, a number of factors which are of great benefit to potential clients will be bought to the fore, or emphasised more than they have. These include a greater effort by vendors on building faster or more robust hardware for enterprise solutions. This doesn't occur enough presently, because most vendors tout advantages of their respective OSs (such as which OS has the most applications.) With a standard open source platform that all vendors use, all vendors will have the same array of applications to offer (give or take a recompile for differing machine architectures.) This will force the emphasis on hardware and service quality. Also, even in cases where many vendors ship hardware with the same operating system, users are still miss out on specifically tailored (for example optimised) operating systems, or operating systems which run

across more powerful processor families. This in turn, forces users to migrate from one platform (say Windows NT) to another, say (a multi-processor Solaris system) when they hit availability or scalability issues. Needless to say, migration of applications with custom built extensions would become a historical curiosity if Linux was used in both the low end enterprise arena (where NT solutions abound) and the high-end (where Solaris sits.)

INDICATORS OF VICTORY

Are there any indicators that the open source model is threatening to become the standard? How do we measure this? My view would be that there are segments of the applications or systems space where growth in use of open source contenders can be measured. The ongoing Netcraft²² analysis of web servers worldwide indicates that even though Apache is one of the more recent web servers, it has more market share than all other web servers put together; and its share is growing every month against contenders like Microsoft's IIS, which is given away 'free'. In fact, with IIS, users have it installed as part of your NT server setup, so there's little effort involved on the part of the installer, in theory, greatly reducing the knowledge gradient needed to bring more IIS servers online. And yet, Apache keeps increasing its lead.

Linux is also increasing in numbers, at what has been estimated as the fastest growth of any operating system. While exact numbers are indeed difficult to ascertain, most researchers have estimated around 8 to 10 million Linux installations (some of these may indeed have hundreds of users, as Linux is indeed a multi-user system.) This number is also said to be growing around 40% per year. What's more, these figures were compiled before Linux hit the big-time in terms of mainstream publicity and acknowledgment from the large software firms which now support it. It's only likely that Linux's growth will accelerate.

On the scripting side, it's obvious for most people that the main scripting systems, on any platform, are open source. These are Perl, which has hundreds of thousands of users, Python and TCL/Tk. GCC is perhaps the most popular cross-platform compiler family. It too is open source.

Unlike their server oriented and technical development tool cousins, open source desktop productivity applications have only recently started to make an appearance. If our model of slow initial take-up and limited functionality for open source projects is correct, it will be some time before truly competitive desktop apps arrive. This indeed doesn't

have to be the case, as The Gimp²³ has shown. The Gimp *may* be an abnormality because there was a severe lack of a quality image processing tool for open source platforms prior to its arrival, by comparison to other applications, like word processors and spreadsheets, of which several excellent contenders exist (Wordperfect, Applixware, StarOffice.) With necessity being one of the driving forces of open source projects, there may be less of a need to produce a quality open source word-processor if one exists for a reasonable (or no) cost. Time will tell if some desktop apps fall outside the gravitational pull of open source superiority.

Another factor which is driving the greater acceptance of open source software is the distribution method. Even five years ago, most computer users would have known shrink-wrap software as the only form of legitimate software distribution and acquisition. Bulletin boards had an air of software piracy and the possibility of virus infection. The Internet has changed all this. Most users who are Net connected are very likely to download and try any number of freeware, demoware, shareware or commercial lite-versions of software on a regular basis. In fact, with higher bandwidths, more and more of PC user's software will come from this channel. There are many advantages. Software publishers can distribute from their web-site for minimal (or no) cost. Second and third level distributors and resellers aren't needed. Manuals are mostly on-line nowadays, and increasingly in platform independent HTML or PDF file formats. Software, if commercial, can be purchased via secure credit card ordering systems. Into this newfound distribution channel, open source software can stride without looking in anyway 'odd' or 'left-of-centre'. Indeed, once past the initial operating system install (for Linux or FreeBSD) most other additional software packages one would need are only a few clicks of a web-browser away. The more fundamental this channel becomes, the more mainstream becomes the default open source delivery method. Indeed, when glossy brochures, retail store packages, shelf displays and sales staff trained to point to higher-earner software vanish, many marketing oriented companies then need to play on a level playing field on software quality and reputation. My guess is that they may be found lacking.

The web has further assisted the spread of open source software, through the rather egalitarian nature of web-info availability. Slashdot.org¹⁶ can garner as many influential readers as the rather more well funded ZDNet. And unlike ZDNet which has a possible vested interest in pushing organisations like Microsoft (Microsoft are perhaps the world's biggest advertisers on the Internet, and thus a likely source of income to commercial company like Ziff-Davis whose online and printed revenue source is primarily advertising,) Slashdot is an example of an 'open

²² Netcraft Web Server Survey
<http://www.netcraft.com/survey/>

²³ The GIMP -- The GNU Image Manipulation Program <http://www.gimp.org/>

source' web-site. Peer-recognition is one of the driving forces behind its founders' aims. Thus, Slashdot, Freshmeat²⁴, Linux Weekly News, LinuxToday²⁵ and LinuxNews²⁶ have garnered not only a cult following amongst the open source faithful, but grudging recognition and acknowledgment from their commercial competitors as places where interesting things occur (news breaks, story leaks from industry insiders etc.)

WHY OPEN SOURCE BECOMES EVER MORE IMPORTANT

In a word, the Internet. It is the factor that has perhaps contributed more towards the open source paradigm becoming the most effective and prevalent. Using open source is even more important now that computer systems are generally not standalone. Internet usage rates in most Western countries stand around 35% of computers. This, in time, will increase. There may be a point in the near future when nearly all systems will be Internet connected. This brings into play a multitude of factors that were never a consideration when all these systems were stand-alone. Security, robustness, ability to use open standard tools are now a fundamental requirement. The open source world delivers on these better than any other development model. The Internet has changed everything. While vulnerabilities abound on all platforms, open source platforms have been shown on all occasions, to plug security holes and potential denial of service attacks faster than others. In fact, there is a definite attempt by commercial systems software vendors of not making too much noise about gaping security flaws in their products, in case customers get spooked and look elsewhere for secure platforms to transact increasingly in this on-line world. This psychology has a close analogy to banks and financial institutions not divulging cracker attacks, for fear of losing clientele. While 'simple' operating systems like Microsoft's Windows could, in some ways, be thought of as secure by virtue of providing no network services, recent events, such as the release of the BackOrifice²⁷ system with virus-like propagation, shows this is not the case. Indeed, by virtue of the fact that Unix-like systems have had to contend with network security from its inception, means that users and administrators of such platforms are more prepared and know the issues involved in security more-so than their Windows counterparts. To quote Linus Torvalds himself on the efficacy of closed source vendors on security issues:

If you look at security bulletins, Linux is impacted by security bugs as much as any other vendor. The thing

²⁴ Freshmeat <http://freshmeat.net/>

²⁵ LinuxToday <http://linuxtoday.com/>

²⁶ LinuxNews <http://linuxnews.com/>

²⁷ Cute name belies gravity of latest NT attack <http://www.infoworld.com/cgi-bin/displayNew.pl?security/981019sw.htm>

is that, when you get the bulletin, Linux already has a patch for it, while the other vendors tend to say, "We are investigating." Microsoft is just horrible. They don't even care.' -- Linus Torvalds
<http://www.linuxworld.com/linuxworld/lw-1998-10/lw-10-torvalds.html>

Viruses, while a common and an ever-present danger for the Windows and MacOS environments, are virtually non-existent for most open source platforms. Linux, and other Unix-like operating systems, were designed from their inception to be used by multiple users, simultaneously. This forced their designers to implement the types of security provisions which renders almost all virus-like attacks harmless. There have been no conclusive analyses done, on the costs organisations bear, in both preventing virus outbreaks, and mopping up after them on Mac and Windows platforms. Microsoft in particular, seems to have a knack in producing platforms which attract these undesirable programs. Initially there were many thousand DOS and Windows based virus executables. Then followed a plethora of MS-Office macro viruses, which were, in a sense, portable to MacOS based MS-Office systems. Later came Microsoft's concept of downloadable ActiveX controls. These, in particular, had considerable potential for damage, as was shown by the German crackers who demonstrated using ActiveX to insert bogus payment authorisation transactions into Windows based personal finance systems, and then let the personal finance manager upload these legitimately to their PC owner's bank. While ActiveX controls, were, in theory, supposed to be authenticated to particular developers, this didn't stop them from doing whatever they wanted, once downloaded to your Windows PC. Thankfully the Internet world completely ignored them. Once again, while open source platforms like Linux can never claim to be 100% virus free, they make the act of spreading viruses much harder than most other platforms. Also, by virtue of the very rapid and open announcements and subsequent fixes to security holes, open source platforms will always be more secure in this area.

SCALABILITY OF OPEN SOURCE RESOURCES

Part of the impetus of the developers of open source software, is the concept of developing and nurturing a space in the IT World where ideas, actions and code are enacted for the public good, and not primarily for the pursuit of corporate profit. Hackers, while getting paid to write code during the day, often feel it necessary to eschew corporatism, a growing philosophy, outlined by John Raulston Saul²⁸.

Another point which drives open source coders, and advocates of open systems in general, is the potential

²⁸ The Unconscious Civilization, John Raulston Saul, Anansi Press, 1995.

for the IT industry to slide back into the dim and ugly days of closed, proprietary systems, or, even worse, the potential to slide back into sole-vendor control of IT. For those who come from a technical background, this would be anathema to the current blossoming of interoperability, source code portability, and with vendors vying with one another on the quality of their implementation of open standards, not by pursuing the old quicksand technique of vendor-lock in. While this, in some ways, may produce results which aren't as slick, as marketable, or as fast to market, it always produces better long-term informational systems. Case in point, is the drive by certain web-browser makers encouraging web developers to create content for solely their web browser. This is in direct contrast to the wishes of the creator of World Wide Web, Tim Berners-Lee, who has stated: 'Anyone who slaps a "this page is best viewed with Browser X" label on a Web page appears to be yearning for the bad old days, before the Web, when you had very little chance of reading a document written on another computer, another word processor, or another network.'²⁹) It should be fairly obvious that open source systems software (and highly important applications like web-browsers) are perhaps our best chance of enforcing open standards. As more and more people understand this, open source will be further strengthened.

Some of the questions I've been asked about these ideas can be paraphrased as: Why should it be important that software systems be open source as much as possible? We've managed for the past 40 years with mostly closed, proprietary code, why change? In short, (and at the risk of trumpeting our own importance,) because IT is now amongst the most important industries on Earth. It is firmly entrenched in all Western countries, and becoming so in most others. Almost everything you touch, watch, hear, read or commute in, is produced or controlled through some computer related process. This is becoming more-so each day. While we, as IT professionals, may have few qualms about endorsing games, maybe some utilities, perhaps even some business applications, in which we know bugs exist, or that we feel aren't very robust, how would we feel about the flaky software in medical equipment, airline control towers or car airbag control systems? As IT permeates more and more of our lives, we *must* have the confidence to know that either a) systems are incredibly robust, or b) we have access to the source, so we can verify the degree of robustness claimed by the system's authors. If you've written much code, you will know that 'a)' is perhaps an impossibility. For a system to be that robust, it must be in existence for some considerable amount of time, to prove itself. In the computer industry, this lapse of time would introduce new development tools and techniques, new hardware, new developers, new operating systems, any one of which could perhaps

²⁹ Tim Berners-Lee (Creator of the World Wide Web) in Technology Review, July, 1996

destabilise the incredible robustness of our claimant. Added to this, is the potential for the closed source developer of 'a)' to cease to exist, leaving no source code. Option 'b)' however, rings of surefooted pragmatism. It may not claim to be bulletproof yet, but it will likely achieve this status faster than 'a)' invoking the open source development and code feedback loop cited elsewhere in this paper. Chances are, that option 'b)' will also have been written to industry standard open APIs, with portability of platform and operating system in mind, and with open source tools which can be maintained to the levels desired by the developers of 'b)', thus preventing degradation of code fidelity through changes to development tools outside their control. Added to this is the fact that there is no concern if the originators of the code in 'b)' cease to exist, since we have the full source to continue on our path regardless. All this, should make a compelling case, pro- open source software as perhaps our most likely (but not perfect) hope for achieving quality, reliable software for all of us in the future.

In recent months, it has become clear that the biggest threat to the open source movement, Microsoft, has begun targeting the movement. Some analysts from the traditional IT publications are henceforth proclaiming that Linux's (and thus open source's) days are numbered. They are correct in saying that the Microsoft marketing machine is the best in the industry (if it wasn't, Microsoft would be nowhere.) They are also correct in their prediction (fast becoming true, as witnessed in the recent comments by the head of Microsoft France) that that same marketing machine is about to ignite a major campaign of FUD (fear, uncertainty, doubt) against Linux and open source. What this will achieve, may in fact, be the opposite of what the pundits and Microsoft believe. Unlike their campaigns against Java, or OS/2, or indirectly against the MacOS, targeting Linux with bad-publicity will only cause the gaze of the computer using public, blissfully unaware that an alternative to Windows actually exists, to turn to Linux. While IBM (OS/2), the Mac (Apple) and Sun (Java) have very deep pockets of their own with which to shower their products with publicity, Linux has almost none. While some distribution vendors (Caldera, Red Hat, SuSe, PHT) do their best, they mainly cover that thin spread of IT journals, websites and magazines whose readers are probably Linux aware, and increasingly Linux users. With Microsoft's help, we are now in a position to read about Linux in our daily newspapers, possible TV commercials, and business journals, on an ever increasing basis. If you must spread FUD, you have to name your target. This will help elevate the word (and possibly a caricature of the open source philosophy) to a level higher than would otherwise be achievable. Why won't this work? Even if suddenly 100 million people who use PCs understand that there is a competitor to Windows, why would they consider Linux? Simple. Unlike most of the past targets of Microsoft's FUD, Linux is free. This means

that the people who suddenly realise that there is an alternative, can try it full, legally, in greater numbers. Further, any extended stream of mud-slinging by Microsoft may backfire, in the same way that negative political advertising often backfires, Linux is not owned by anyone. It has an almost saintly image. Microsoft hurling mud at the 'John Lennon' of operating systems would probably cause minor revulsion amongst the computer using community. In short, Any publicity for Linux is good publicity; Linux (and open source) wins in the end.

Yet another bolstering force, primarily to Linux, but also for open source in general, is the 'Domino' effect of software vendors. By way of description, many ISVs have felt for quite a while that one vendor, Microsoft, has too much power and is constantly increasing its slice of the market, at the expense of their earnings. Furthermore, it does this in a manner which the United States Department of Justice (and 20 other States) agree is illegal. Thus, there is a pent-up wall of frustration from many of these ISVs, as they feel unable to compete on a fair playing field on any Microsoft platform. If a viable alternative platform does appear (and it has with Linux) many of these vendors will migrate their apps and systems. At first, it will be (and has been) a slow trickle. Then, the dam-walls will burst, and the Linux market will flood with hundreds of commercial apps that weren't there before. This is the point in which we are at now. Already, there are moves in place by Corel to fast-track their desktop application ports to Linux, by helping finish and then using WineLib³⁰. With a fully functioning WINE (a system for running Windows applications in Linux and other platforms³¹) in place, most vendors will then only need to test their current generation of Windows applications against Linux, not re-write them. With this in place, there will be a broad rush to support Linux, as these ISVs must know that Microsoft will be the last one moving core applications to the Linux platform, and that they will therefore be safe in pursuing business in this new space. In the end, this may precipitate the reverse of the events which led to their current monopoly status. Around ten years ago, a major shift in platforms occurred, when users started migrating from DOS to Windows. By virtue of Windows being their own new platform, Microsoft was first with a word-processor, a spreadsheet, a Windows database, and many of the other 'staple' desktop applications. This placed them in pole position to snare as much of the applications market as possible. By the time Lotus, Borland and Software Publishers developed equivalent apps, Microsoft had a big lead. Product bundling, likely anti-competitive practices, possible use of secret APIs to enhance performance and capabilities of their own apps and closed proprietary

³⁰ From: Gavriel State Newsgroups: comp.emulators.ms-windows.wine Subject: Announcement: Corel involvement with WINE Date: Fri, 30 Oct 1998 16:14:58 -0500

³¹ WINE Development HQ <http://www.winehq.com/>

file formats allowed them to wrest the remainder of the remainder of the market from their competitors. Linux will herald a new age, when Microsoft will be the last to market, and will not have the ability to use secret API calls (as Linux source is open to all ISVs.) Microsoft, if they ever decide to join the Linux bandwagon, will most likely be late, and will have to play like everyone else.

Following further in the footsteps, or perhaps flavour of science, let's make some prognostications for the (near) future. What immediate changes will become obvious if open source becomes more and more prevalent, in both use, and acceptance in the IT industry. Some possibilities are:

- Commercial vendors will start modifying their approach to software distribution and support. Vendors like Microsoft are even now opening up their support unto previously underutilised paths, like mailing lists (which have been the vehicle of information delivery for core os software from the beginning.)
- There will be a steady reduction in software prices. There will be more 'free', cut down versions given away as loss leaders by major closed software vendors. In the past these firms used this as a tool to increase market share. Bear in mind that market share in app areas such as tools, business productivity apps and operating systems is king. People buy and use MS-Word because so many others do. It is what people write complex macros in; it's what everyone uses for standard file-formats; it's what staff are trained in; it's what 300 books have been written about. Due to its market leader status, it acquires all these as 'extra' defences against competing products. While the paradigm stays as is, everything moves along steadily. No matter what Microsoft's competitors do in many desktop apps sectors, there will be little change. But, with the looming industry catharsis wrought by Linux, a platform which will likely overtake Windows in a few years time, there will be a much jockeying by ISVs for market share in this new market. This can be visualised as a race by the sapling trees to reach the tree canopy, once the current 'king' which has been showering them with shadow, is toppled.
- Incorporation of more open-source systems software amongst various vendors. This too is already starting to happen, with IBM putting the Apache web server into many of its e-commerce bundlings. There is considerable scope for rolling out GCC, Perl, Python, TCL and dozens of other powerful technologies as standard, in more and more commercial operating system platforms.
- A major uptake by computer appliance vendors. Products like the Qube, and other Linux based thin server, thin client and network appliances. This may include set-top web-boxes, portable

and car-based MP3 players etc. These are starting to appear.

- More commercial vendors releasing their code as open source, and providing packaged versions, with official support as the main form of earning. This may happen only tentatively at first, and in instances where the vendor sees no substantial strategic loss in constricting source code supply (say with older applications, or applications about to be dis-continued.) Once these early releases start producing results, and the companies that produce the 'branded' versions remain profitable, more will follow.

CONCLUSION

Using current eddies to predict future waves isn't an easy task. My intention here has been to point to a number of areas where I see that open source has substantial advantages, and that these will be recognised by more and more people with time. As this 'new' development process becomes mainstream, and people feel more comfortable with it, this effect will snowball. I have also attempted to show that the open source process and philosophy mirrors another highly successful human discipline, namely science.

As science has gone through numerous phases of increasing effectiveness, so has the open source paradigm. It should from this point on, be looked upon as our best method of writing secure, robust, efficient, extendible code for our future data processing needs. If this paper has assisted in adding even a *minute* push to the open source movement, it makes the prediction which we began with, just that bit more likely.


ACKNOWLEDGEMENTS

My thanks to the following people for their ideas, suggestions, comments and corrections:

- Geoffrey D. Bennett (geoffrey@netcraft.com.au)
- Raph Levien (raph@acm.org)
- Don Marti <http://www.electriclichen.com/>
- Raoul Golan (raoul@ind.tansu.com.au)
- G. Richard Keech (rkeech@cyber.com.au)
- Axel Boldt (axel@uni-paderborn.de)

All trademarks are acknowledged.





IT Consulting
Services

Cybersource

ACN: 053 904 082

Cybersource has been a Professional Services consultancy, specialising in the areas of Unix, Windows and TCP/IP since 1991. Cybersource also offers accredited, professional-grade support for Red Hat Linux and other open source (free) software. Therefore, the last 'valid' reason for not taking advantage of great software like Perl, Linux, SAMBA and Apache has just disappeared. Organisations can benefit from the robustness, flexibility and value of open source software, and know they have an experienced team of IT professionals available to provide commercial-level support, when needed.

Contact us for full details.

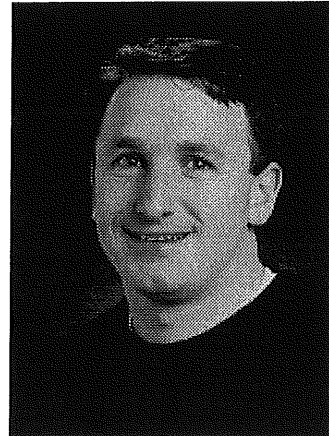
Telephone:	03 9642 5997
URL:	http://www.cyber.com.au/
Email:	info@cyber.com.au

John Lions Award - 1998 Winner

The John Lions award has been instituted to recognize the leading role that John Lions has played in bringing UNIX to Australia, the formation of AUUG, and the promotion of the values held by the open systems community. It is given for undergraduate or postgraduate student research work on interesting uses of open systems technology, contribution to understanding of open systems, programs, tools or knowledge about UNIX and open systems.

This year's winner of the Lions award for work in operating systems was Steve Blackburn from the Australian National University in Canberra. Steve describes his work like this:

"The challenge of scalably managing complex data is becoming more and more pressing as the growth in communications and computing accelerates and demand for information snowballs---the so-called 'information explosion'. While relational, object-relational, and object-oriented databases each play a role in addressing this problem, the most elegant approach to persistent data management lies in the use of 'orthogonal persistence', which seamlessly integrates the notions of persistence and resilience familiar to users of database technology with the richness of programming languages. Orthogonal persistence thus removes the 'impedance mismatch' so familiar to DBMS developers.

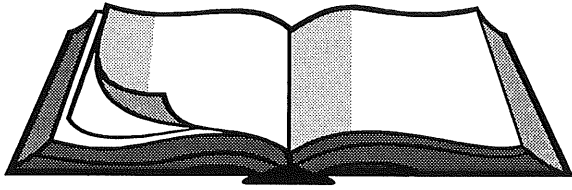


Unfortunately the promise of orthogonal persistence has, despite more than fifteen years of research, yet to be delivered. Analysis of this failure suggests that a key problem lies in the very mismatch that orthogonal persistence seeks to address---the gap between database and programming language research and technology. It would seem that no single research group can summon the energy to address the enormous range of concerns covered by the two fields. The Persistent Store Interface (PSI) is a realisation of an open interface between runtime systems (such as the Java Virtual Machine) and storage systems (such as database engines). By providing such an interface, PSI opens opportunities for collaboration and reuse in the research community, thereby providing a bridge for this technological and social disconnect that has hampered the development of what promises to be a significant technology in the management of complex data. Recently published results indicate that PSI provides a highly efficient means of integrating leading edge technology from the two fields."



Book Reviews

Sub-editor:
Mark Neely
<accessnt@ozemail.com.au>



FOUNDATION FOR OBJECT/RELATIONAL DATABASES: THE THIRD MANIFESTO

C.J. Date and Hugh Darwen
Addison Wesley Longman, Inc. 1998
ISBN 0-201-30978-5, 496pp.

Reviewed by:
Michael Haldey, PhD
<mhaldey@ozemail.com.au>
and
Alex Jouravlev
<alexj@zip.com.au>

The information technology industry faces a number of critical challenges. One of the primary challenges is the future of object and relational data and database management systems. With many academics seemingly distracted by 'religious wars', industry professionals, including DBMS vendors, system architects and tool developers are left anxiously seeking guidance and waiting for key issues to be resolved. In this environment, the contributions of respected commentators, such as C.J. Date, are eagerly awaited.

The authors of *Third Manifesto* attempt to develop a "detailed proposal for the future direction of data and database management systems". However, they deliberately omit any implementation details.

Part I of the book outlines the foundation of the authors' extended relational data model, which encompasses object oriented concepts. Their model is based on the general assumption that 'class = domain', which they claim never contradicts the existing relational model. Such assumptions allow the authors to state that the relational model can be extended to include complex datatypes with the associated methods, inheritance and other features required by object oriented technology. The authors express the opinion that the relational model was misunderstood by the industry, especially the database vendors who 'failed' to implement it properly in their products, including the 'failed' SQL.

It is beyond of the scope of this review to discuss the proposed and existing models of data, however we do note that some statements in Part I are extreme and quite controversial.

Part II of the book contains the formal specifications for "D", a language that implements an interface to the proposed model. These specifications consist of prescriptions, proscriptions and very strong suggestions, grouped as RM (relational model) and OO (other orthogonal) components. The descriptions are quite detailed, and yet are more readable than would usually be expected of a formal document. However, it is suggested that Part III be read first as it explains the same concepts, though in a less formal manner.

Part IV contains both formal specifications and an informal discussion of and explanations for subtyping and inheritance in the proposed model context.

The book contains numerous Appendices, which provide detailed descriptions of the "D" calculus, further discussions of some design issues, comparisons of "D" with SQL3 and ODMG proposals, and a 1994 interview with the authors in which they discuss their previously published paper on the Manifesto. The final Appendix presents a list of references with comments by the authors on some works.

Overall, the authors present a well documented, clear and consistent position on a contentious subject. The book is a must for those involved in academic research on the data models and related issues and those lucky few who determine the directions of future product development.

Yet the book is disappointing in some respects. For instance, it lacks a good conceptual overview of the authors' proposed model. To understand it, one is expected to carefully read the entire book and work through an enormous amount of detail. In addition, the authors often state their position or conclusions without providing a detailed discussion or reasoning. An example of this is their treatment of SQL.

As such, we doubt that the book can correctly be described as a manifesto, as its audience seems to be limited to the few specific groups mentioned previously.

For those interested in this area, the authors' have published a paper on the subject which provides an overview of their ideas: <http://www.acm.org/sigs/sigmod/record/issues/9503/manifesto.ps>.

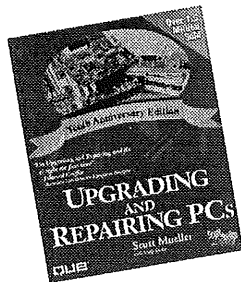
❖

AUUG Local Chapter Meetings 1998

CITY	DATES	LOCATION	OTHER
BRISBANE	24 November	Inn on the Park 507 Coronation Drive Toowong	For further information, contact the QAUUG Executive Committee via email (qauug-exec@auug.org.au). The technologically deprived can contact Rick Stevenson on (07) 5578-8933. To subscribe to the QAUUG announcements mailing list, please send an e-mail message to: <majordomo@auug.org.au> containing the message "subscribe qauug <e-mail address>" in the e-mail body.
CANBERRA	10 November 8 December	Australian National University	
HOBART	Each month, although dates can vary. Often will fit in with the schedule of a speaker should one be available.	University of Tasmania	
MELBOURNE	18 November 16 December	Various. For updated information See: http://www.vic.auug.org.au/auugvic/av_meetings.html	The meetings alternate between Technical presentations in the odd numbered months and purely social occasions in the even numbered months. Some attempt is made to fit other AUUG activities into the schedule with minimum disruption.
PERTH	18 November 16 December	The Victoria League 276 Onslow Road Shenton Park	Meeting commences at 6.15pm
SYDNEY	19 November 17 December	The Wesley Centre Pitt Street Sydney 2000	

*** All dates are subject to change.**
Up-to-date information is available by calling AUUG on 1-800-625-655.

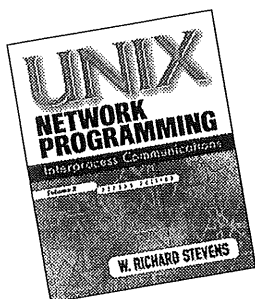
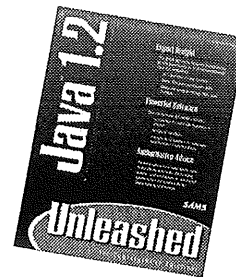
AUUG BOOK CLUB AND PRENTICE HALL AUSTRALIA



Upgrading and Repairing PC's
 Scott Mueller
 07879716364 \$109.95 87.95
 Super value - all of the information from Scott's \$1,300 3 day seminars in this handy reference. Upgrade and troubleshoot your entire PC - video, audio and network cards, drives, processors, memory and motherboards. Learn the real benefits and drawbacks of new technologies from an experienced PC authority.

"The most comprehensive and up-to-date edition ever - used by over 1.5 million satisfied readers."

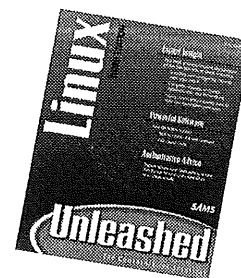
Java 1.2 Unleashed
 J Jaworski
 1575213893 \$109.95 87.96
 The unleashed series takes you beyond the introductory discussions of the technology giving you practical advice and in-depth coverage. With these extensive guides, you'll obtain the skills, understanding and breadth of knowledge to unleash the full potential of Java 1.2



**Unix Network Programming
 Vol 2 - Interprocess Communications**
 R Stevens
 0130810819 \$89.95 71.96
 This volume presents a comprehensive guide to every form of IPC, including message passing, synchronization, shared memory, and Remote Procedure Calls. Stevens begins with a basic introduction to IPC; then step by step you'll learn how to maximise both system V IPC and the new Posix standards. You'll find extensive coverage of Pthreads, and you'll master every current IPC technique and technology.

Also available Unix Network Programming:
 Networking API's: Sockets & XTI Vol 1 -
 \$89.95 \$71.96

Linux Unleashed
 T Parker
 0672313723 \$79.95 \$63.96
 This book presents focused explanations of the core features of Linux to meet your needs; complete coverage of System Administration and Networking, and how to use Linux programming tools to help you accomplish your goals.
 It includes 2 cd roms



MAIL, FAX OR PHONE US WITH YOUR ORDER TODAY!



MAIL
 Reply Paid AAA0243
 Marketing Department, Prentice Hall Australia
 Locked Bag 531, Frenchs Forest NSW 1640



FAX (AUST)
 (02) 9453 0117



PHONE (AUST)
 (02) 9453 2302

Name: _____ Position: _____

Company: _____

Address _____

Telephone: _____

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

Charge to me OR Company purchase Order No. _____

Please charge my Bankcard Visa Mastercard AMEX

Expiry Date: _____ Credit Card No:

Signature: _____



PRENTICE HALL
 AUSTRALIA

AUUG1098

UNIX Traps & Tricks

Sub-Editor:
Matthew Dawson
<dawson.matthew.ms@bhp.com.au>

Hi Everyone! Welcome to another edition of UNIX Tricks & Traps - the column designed to provide insights into how your fellow AUUG members make their day-to-day usage of UNIX easier.

This issue's UT&T is a little smaller than most that I have submitted since taking over the column. There is only one reason for this - very few tips/traps have been sent to me since the last edition of AUUGN was printed. In fact if it wasn't for the two tips received during the previous fortnight there would be no column at all!

So I'd appreciate it if you could all take a few moments to jot down those pearls of UNIX wisdom that you regularly use, and send them my way. Your fellow AUUG members will thank you for it!



MAKING A COLOURED PROMPT IN BASH

From: Richard Keech <rkeech@cyber.com.au>

A coloured prompt does more than simply add a bit of life to a colourless terminal window. When you've got a long sequence of command/output/command etc on the screen, having a different colour for the prompts aids in quickly visual recognition. Another point is that having a distinctive colour for a root prompt aids in giving root shells due respect. I use red for root and green for user.

The following is how I do coloured prompts in bash. Not all xterms support the ISO colour codes. I know it works with xterms in Red Hat Linux 5.x and Solaris 2.{5,6}. I know it doesn't work with SunOS 4.x. The line `USER=`id -un`` will vary with some different OSes.

```
#-----
# ISO 6429 character sequences for colors etc

# lc=Leading character.
lc='\[\033[1;'
#foregrounds----backgrounds-----
BLACK=${lc}30m; B_BLACK=${lc}40m
RED=${lc}31m; B_RED=${lc}41m
GREEN=${lc}32m; B_GREEN=${lc}42m
YELLOW=${lc}33m; B_YELLOW=${lc}43m
BLUE=${lc}34m; B_BLUE=${lc}44m
PURPLE=${lc}35m; B_PURPLE=${lc}45m
CYAN=${lc}36m; B_CYAN=${lc}46m
WHITE=${lc}37m; B_WHITE=${lc}47m
#-----
BRIGHT=${lc}1m
UNDER=${lc}4m
FLASH=${lc}5m
RC=${lc}0m # reset character
#-----
if [ "x${USER}" = "x" ]
then
  # USER is not set
  USER=`id -un`
fi

# set pc, the prompt color
if [ "$USER" = "root" ]
then
  pc=$RED
else
  pc=$GREEN
fi
#-----
# set the prompt
if [ $TERM = "dumb" ]
then
  # no color if a dumb terminal
```

```
pc=" "; RC=" "
fi
```

```
PS1="$ {pc}\ | \u@\h \W\ \$ {RC}\ ]"
#-----
```

❖

TIDYPATH CONTINUES

From: David Keegel <djk@cyber.com.au>

At the risk of perpetuating the "tidypath" saga, I wanted to make a short comment on Jason Tyler's shell script in V19N3.

A technique from 'real' programming called "sentinels" can be quite useful for shell scripts in general and dealing with \$PATH in particular.

Instead of

```
case "$n" in
    $i|$i:*|*:$i:*|*:$i) ...
```

you could use

```
case ":$n:" in
    *:$i:*) ...
```

This improves readability, maintainability and possibly even performance.

❖

SOME MYSTERIES OF FIND(C)

From: Luigi Cantoni <lui@stm.com.au>

Often in the UNIX manuals it is difficult to determine exactly what is meant. Here is an excerpt from my manuals for the command find(C) and some practical examples to explain what it really means.

Manual Extract

Some primaries are automatically true (for example, -print or -depth). It is possible to negate a primary (so that it evaluates to the logical negative of its real value) using the `!' operator; when used in conjunction with the -o (logical-OR) and \(..... \) (grouping) primaries this permits the construction of complex logical conditions.

(expression)

True if the parenthesized expression is true. Usually used with the -o operator (see below), parentheses are used for grouping. Parentheses are special to the shell and must be escaped.

expression -o expression

Alternation of primaries (the OR operator). Placing the -o operator between two primaries creates an expression that is true if either of the two primaries is true. It should be used with parentheses (that is, \(-perm 644 -o -perm 664 \) is true if the current file has permissions 644 or 664).

Practical Examples

```
find / \( -name core -o -name "*.out" \) -atime +7 -exec rm {} \;
```

After reading the find manual page the behavior of this command seems easily predicted. My reading of this (and I am sure its not the only way to read it) indicated that you use the parentheses for bracketing like you normally would to bind expressions together. You look at the example that's given and you say "Ah yes, it binds the `OR`ed expressions together before the implied `AND` of the -atime. Great - it all makes sense to me."

What is not made clear, though when you know what to look for it is in the manual, is that brackets are required by all -o operations. This is especially important given that almost every find I have ever seen uses either an -exec or -print. Primaries such as these are always true but I am sure most people would expect this find:

```
find . -name "*.out" -print
```

to mean:

Execute find in my current directory; if there is an entry with a name that matches "*.out" then print out its name.

WRONG! This is what the command actually means:

Execute find in my current directory; if there is an entry with a name that matches "*.out" *AND* print out its name.

You see there is an AND operator hidden in there. This makes no difference when there are no OR operators in the command but it sure as hell does make a difference when there are.

Here is a slightly more complex example.

```
find . -name core -o -name "*.out" -print
```

If you interpreted this similarly to the previous example you would expect the behavior to be:

Execute find in my current directory; if there is an entry with a name that matches core OR a name that matches "*.out" then print out its name.

The command's behavior will actually be:

Execute find in my current directory; if there is an entry with a name that matches core OR a name that matches pattern "*.out" *AND* print out its name.

What you have to remember is that -print is just another expression and that the AND logical operator has a higher precedence than OR. Putting the implied brackets into the previous find makes it look more like this:

```
find . -name core -o ( -name "*.out" -a -print )
```

This shows what will actually happen - file names that match the second pattern (*.out) will get displayed while the ones that match core will not. This, I am sure, is not the desired result.

Here is a practical little script that shows how it all works:

```
#!/bin/sh
mkdir /tmp/test
cd /tmp/test
>aa
>bb
>cc
echo "This find will show aa"
find . -name aa -print
echo "This find will show bb"
find . -name aa -o -name bb -print
echo "This find will show cc"
find . -name aa -o -name bb -o -name cc -print
echo "This find will show all three aa, bb and cc"
find . \( -name aa -o -name bb -o -name cc \) -print
```

In summary you need to put parentheses around **all** OR operations because there is always an implied AND operation before a primary (e.g. -print, -exec) that I am sure you want done for all the OR cases.

❖

Application for Institutional Membership

Section A: MEMBER DETAILS

The primary contact holds the full member voting rights and two designated representatives will be given membership rates to AUUG activities including chapter activities. In addition to the primary and two representatives, additional representatives can be included at a rate of \$70 each. Please attach a separate sheet with details of all representatives to be included with your membership.

NAME OF ORGANISATION: _____

Primary Contact

Surname _____ First Name _____
 Title: _____ Position _____
 Address _____
 Suburb _____ State _____ Postcode _____
 Telephone: Business _____ Facsimile _____
 Email _____ Local Chapter Preference _____

Section B: MEMBERSHIP INFORMATION

Renewal/New Institutional Membership of AUUG \$350.00
 Surcharge for International Air Mail \$120.00
 Additional Representatives Number @ \$80.00

Rates valid as at 07/96

Section C: PAYMENT

Cheques to be made payable to **AUUG Inc** (Payment in Australian Dollars only)

*For all overseas applications, a bank draft drawn on an Australian bank is required.
Please do not send purchase orders.*

-OR-

Please debit my credit card for A\$ _____
 Bankcard Visa Mastercard

Name on Card _____
 Card Number _____
 Expiry Date _____
 Signature _____

Please mail completed form with payment to: _____ Or Fax to: _____

Reply Paid 66 AUUG Inc
 AUUG Membership Secretary (02) 9332-4066
 PO Box 366
 KENSINGTON NSW 2033

Section D: MAILING LISTS

AUUG mailing lists are sometimes made available to vendors. Please indicate whether you wish your name to be included on these lists:

Yes No

Section E: AGREEMENT

I/We agree that this membership will be subject to 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: _____
 Title: _____
 Date: _____

AUUG Secretariat Use

Chq: bank _____ bsb _____
 A/C: _____ # _____
 Date: _____ \$ _____
 Initial: _____ Date Processed: _____
 Membership#: _____



UNIX® AND OPEN SYSTEMS USERS

Membership Application

AUUG Inc Secretariat

PO Box 366, Kensington NSW 2033, Australia

Tel: (02) 9361 5994
 Free Call: 1 800 625 655
 Fax: (02) 9332 4066

email: auug@auug.org.au

ACN A00 166 36N (incorporated in Victoria)

<http://www.auug.org.au>

AUUG Inc is the Australian UNIX and Open Systems User Group, providing users with relevant and practical information, services and education through co-operation among users.

AUUG OFFERS SOMETHING FOR YOU!!

AUUGN
Technical Newsletter
 AUUG's bi-monthly publication, keeping you up to date with the world of UNIX and open systems.

Education
 Tutorials
 Workshops

Events.....Events.....Events

- Annual Conference & Exhibition
- Overseas Speakers • Local Conferences
- Roadshows • Monthly meetings

DISCOUNTS
to all AUUG events and education.
Reciprocal arrangements with overseas affiliates.
Discounts with various internet service providers, software, publications and more....!!

Connections
 • Newsgroup
 aus.org.auug

Application for Individual or Student Membership

Section A: PERSONAL DETAILS

Surname _____ First Name _____
 Title: _____ Position _____
 Organisation _____
 Address _____
 Suburb _____ State _____ Postcode _____
 Telephone: Business _____ Private _____
 Facsimile: _____ E-mail _____

Section B: MEMBERSHIP INFORMATION

Please indicate whether you require Student or Individual Membership by ticking the appropriate box.

RENEWAL/NEW INDIVIDUAL MEMBERSHIP
 Renewal/New Membership of AUUG \$100.00

RENEWAL/NEW STUDENT MEMBERSHIP
 Renewal/New Membership of AUUG (Please complete Section C) \$25.00

SURCHARGE FOR INTERNATIONAL AIR MAIL \$60.00

Rates valid as at 07/96

Section C: STUDENT MEMBER CERTIFICATION

For those applying for Student Membership, this section is required to be completed by a member of the academic staff.

I hereby certify that the applicant on this form is a full time student and that the following details are correct.

NAME OF STUDENT: _____
 INSTITUTION: _____
 STUDENT NUMBER: _____
 SIGNED: _____
 NAME: _____
 TITLE: _____
 DATE: _____

Section D: LOCAL CHAPTER PREFERENCE

By default your closest local chapter will receive a percentage of your membership fee in support of local activities. Should you choose to elect another chapter to be the recipient please specify here:

Section E: MAILING LISTS

AUUG mailing lists are sometimes made available to vendors. Please indicate whether you wish your name to be included on these lists:
 Yes No

Section F: PAYMENT

Cheques to be made payable to **AUUG Inc**
(Payment in Australian Dollars only)

For all overseas applications, a bank draft drawn on an Australian bank is required. Please do not send purchase orders.

-OR-

Please debit my credit card for A\$ _____
 Bankcard Visa Mastercard

Name on Card _____
 Card Number _____
 Expiry Date _____
 Signature _____

Please mail completed form with payment to: Or Fax to:
 Reply Paid 66 AUUG Inc
 AUUG Membership Secretary (02) 9332-4066
 PO Box 366
 KENSINGTON NSW 2033
 AUSTRALIA

Section G: AGREEMENT

I agree that this membership will be subject to 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.

Signed: _____
 Date: _____

AUUG Secretariat Use

Chq: bank _____ bsb _____
 A/C: _____ # _____
 Date: _____ \$ _____
 Initial: _____ Date Processed: _____
 Membership#: _____

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) 9332-4066 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 # _____



