Sun, Surf and X in California.

Andrew McRae

Megadata Pty Ltd.
2/37 Waterloo Road
North Ryde
andrew@megadata.mega.oz

ABSTRACT

"It's Heisenberg's Principle of Uncertainty all over again - you can either tell where the mouse is, or that it is moving, but not both"

Xhibition 89, an industry conference focusing on the X window system, was held from June 24 to June 27 in San Jose California. The conference consisted of a number of tutorial sessions, combined with a technical program and an industry trade show.

This paper attempts to summarise the salient features of the conference, and to highlight key issues. Particular focus is given to new developments, and the strategic directions occurring in the X world. The goal is to provide a 'snapshot' (screendump?) of the current state of X and associated developments, especially as it relates to the UNIX† world.

Introduction.

Xhibition 89 comprised of 18 tutorials, and 37 technical sessions, as well as a parallel industry exhibition, held over 4 days. The conference was well attended, with some sessions attracting several hundred participants. Whilst the central focus was on the X Window System, a number of tutorials and papers dealt with related topics in the general area of graphical windowing systems. The tutorial sessions ran for a half day each, except for a two part introductory tutorial which spanned two sessions. The technical program had a number of formats, ranging from panel discussions, to technical presentations, to multiple lectures within the same session. Keynote addresses were given by David Tory of Open Software Foundation (OSF), and Thomas Macy of Unix International (UI).

As an industry conference, it highlighted the strategic importance of the X Window System, and was a good testimony to the acceptance of X as a standard in graphical windowing systems.

Overall Impressions.

I came away from Xhibition with a clear and unmistakable sense that finally the computing industry has come up with a standard that *no one* is arguing about. The other conviction was that there is still a lot of debate about which window manager to use, which look and feel to adopt, which toolkit to base your application on, and which operating system to run on your computer.

One of the most frustrating arrangements of the Xhibition was that up to five sessions were held at the same time, thereby making it impossible to cover more than one in four sessions. The tutorial sessions overlapped the technical program for a day, and the industry exhibition was held over the final two and a

† UNIX is a registered trademark of AT&T UNIX System Laboratories.

half days of the technical program. At the end of the four days I felt that I had missed out on a number of sessions that may have been extremely worthwhile. I even changed sessions mid stream if I felt that the content was not as good as I had expected. A better balance may have been achieved if the conference had been a least a week long, and perhaps fewer papers.

The tutorials were partially disappointing, in that some tutorial material was covered in the technical program, and some of the tutorials were not presented very well. Other comments I heard rated the material from poor to very good, depending on the tutorial. Some material presented in the technical program belonged in the tutorial schedule, and vice versa. A summary of the tutorials is presented in an appendix.

The conference technical program was generally excellent in its coverage of a wide variety of subject matter, and in providing relevant and up to date information. Key areas covered included Look and Feel issues, X Terminals, Server issues (porting, evolution, improvements), Toolkits, Window managers.

The trade show was on the outside not nearly as large as I expected, but once I scratched the surface I was impressed with the commitment most vendors had in supporting X. There was lack of X based applications. More about this later. Most large vendors such as DEC, IBM, HP etc. did not have large stands or many workstations running X. Software houses selling X orientated tools such as user interface builders and compilers, or tools related to X seemed to make up the bulk of the Exhibition. A number of vendors were selling X terminals, which in themselves are interesting products. Overall the exhibition represented a good proportion of the smaller vendors and software houses specialising in X products, but the turn out of larger vendors was disappointing, especially as most of them are claiming conversion to the X cause.

Most of the systems running at the show were connected via ether (‘shownet’), and a T1 line provided Internet connection, which allowed some very nice demonstrations of interconnectivity, and proved that X really did work as a machine independent graphics platform. *Framemaker* was the main application everybody would run to prove that their X box actually worked (the Lotus 1-2-3 of X terminals?), and I suspect that demonstrating it on most vendor’s hardware did more for selling *Frame* than it did for selling the platform. It was instructive to watch it running on an X terminal, a PC X server, and a high end 20 Mip workstation. It certainly proved that a floating licence worked.

The State of X.

Since the release of R3, X has stabilised, and generally the attitude was that X is here to stay, and we had better all do something about it. It was encouraging to see that in the midst of double standards such as OSI & TCP/IP, OSF/1 & System V.4, NSF & RFS ad nauseum, X is seen as an accepted standard. It is also generally recognised that the closely associated portions of the X distribution such as the X toolkit are also considered part of the X standard, even though all applications may not use them.

Now that the basic platform is stable, a number of important extensions are forthcoming to address particular needs or future issues. The development of these extensions is not aimed at fixing deficiencies in the basic X standard, but are a natural evolution designed to build on top of a stable foundation, much as protocol layers are built on top of transport mechanisms, and link mechanisms.

One issue is how X will be ported to operate over an OSI based communications link. It highlights the fact that X does not rely on a particular transport protocol, but it does raise the question of where do you put X in the protocol stack? It contains session and presentation information, and may interact at the application level via window managers and inter client communications. It certainly leaves the purists in a dilemma.

PEX (PHIGS Extensions to X) is a maturing extension which enhances the server in providing 3D operations. The goal is to allow PHIGS/PHIGS+ clients access to 3D operations on the server display, without mandating the level of support (colour, hardware assist) the server is required to have. In the same spirit as X, PEX does not define the policy of the operations, but provides a mechanism by which higher layers may provide policy. PEX will be key in the acceptance of X in higher end graphic applications such as imaging, rendering, simulation etc. The provision of PEX will mean greater portability and market opportunities to software written to the standard, and the emergence of hardware dedicated to this area, as well as the use of standard workstations (maybe with accelerated graphics).

XIE (X Imaging Extension) is designed to support digital imaging and rendering under X, and is still undergoing some development in the X consortium. The idea is that XIE will provide server extensions to allow more sophisticated imaging and image data processing to occur in the client and server.

VEX (Video Extension to X) deals with low end video handling and transfer of video data; a nice example of this was a demo program running at the show which had ‘Star Trek II’ being shown real time in a window, which could be resized, moved or closed. As X is increasingly used as the standard graphics transport, and applications such as video conferencing become accepted, this is an important extension.

At a display level we’ve seen the development of sophisticated window managers; a new term has been invented to describe a program handling the connection and control of a display to a host (such as logging in and accessing applications), called a **session manager**. The session manager may be the same program as the window manager, but is usually a separate program controlling the access to various host resources. My understanding of a session manager is somewhat sketchy, as the definition overlaps with some window management functions, resource allocation and host accessing.

The adoption of the Inter Client Communication Convention (ICCC - nicknamed ‘ice-cubed’) has helped to standardise on the method of client interaction by generalising such operations as selection, text cut and paste etc. It is expected that ICCC will evolve to handle graphic object manipulation (picture cut and paste), and to formalise the communication between clients, window managers and session managers.

Release 4 of the MIT X distribution promises to have an even more overwhelming volume of software, and it is clear that X users and developers are getting the benefit of a large amount of software development in the industry. The openness of X is encouraging, with a large number of vendors placing software (as well as fonts) into the public domain via the X distribution. R4 will expand the core distribution by migrating some of the contributed software into it (I can visualise now that getting *your* window manager into the core distribution will bring showers of praise and elevate your standing to heights undreamt of), and start to address some of the gaps that the current distribution has, such as more window managers, useful clients, security measures, optimised servers etc. Older or obsoleted software will be retired to an accompanying ‘unsupported software’ section.

David Rosenthal and Jim Fulton chaired a discussion on X and Security; to sum it up, there is no security. Yet. R4 will have a start by putting some security hooks in, and it is likely that the general direction is to base it on authentication via *Kerberos*. Apart from the obvious problems to do with workstation security, and network transparency, X servers usually have no restriction on which clients have access to the display, and can even allow clients to ‘grab’ other windows and read the contents. An X based virus is too nasty to contemplate. Speaking to David afterwards he noted that it is unclear whether Kerberos will be allowed to leave the shores of the USA by the DoD (I guess he meant the DES stuff, not the generic Kerberos code), and he sympathised with developers in the Antipodes (‘You could always move to the US’).

To summarise, X seems to be a success in its basic goal to provide a platform independent graphics transport mechanism, and the current efforts are directed towards provided extensions to address specific technical needs without clashing with the fundamental nature of X, and to standardise on supporting structures such as toolkits, and look and feel.

Look and Feel.

The standards debate seems to have moved to a higher level away from X (e.g. X vs NeWS etc), and the Look and Feel question is currently the predominant issue that is being discussed; without doubt the two contenders being OSF’s Motif, and AT&T’s Open Look. Much the same attitudes abound as with the generic OSF/1 and System V.4 debate, with the difference being that you can at least look at and feel the contending products that result.

Open Look from AT&T is a Look and Feel specification, which defines the style of window presentation, and provides guidelines on the window operations. AT&T is selling a Open Look toolkit which implements the Open Look specification. Sun Microsystems have developed XView, an Open Look conforming toolkit, which is going to be included in the X11 R4 distribution, along with the X11/NeWS merged server. The AT&T Open Look implementation is based on the standard Xt intrinsics, whilst XView is not (a fact that may have significance, depending on the evolution and importance of the toolkit).

OSF/Motif, on the other hand (screen?), is also based on the Xt intrinsics, and the Motif specification not only includes the widget descriptions, but defines a Motif Style Guide (similar to Open Look in principle), and also includes MWM, the Motif Window Manager. The Motif Look and Feel is compatible with

Presentation Manager. I was impressed with Motif as a Look and Feel, as it defined push buttons with a 3D effect, and the window borders had a similar 3D look to them.

It seems that most vendors are aligning themselves one way or the other depending on their basic loyalties, but the software developers are not revealing their hands until they feel one or the other is pre-eminent. In fact it looks likely that developers will have to support both toolkits to properly cover the X market.

The word **standard** was probably the greatest misnomer used when the various parties were describing their own Look and Feel packages, as it is difficult to claim to have a standard product when no copies have shipped to the real world. The best summary I heard was from Richard Stallman, stating the obvious that it matters not whether look and feel A is easy to use, or that look and feel B is nicer to look at, but that we only have **one** look and feel. I can only agree.

OSF and UI.

The keynote addresses were a disappointment, and I felt that I didn't learn anything new, except a new appreciation for rhetoric and the number of ways you can combine the phrases *open standard, open software, open solutions, open specifications, open systems, open extensions, vendor independence* and *vendor neutral.* The sound of beating of drums, and grinding of axes was unbearable.

I spent some time in discussion with some of the OSF staff, and it was enlightening to hear about the decision making process within OSF. I think that OSF is becoming aware of the growing disenchantment of the end users and software developers, and the general problem of not knowing the specific directions, intentions and timetable of OSF, and perhaps we shall see a more active role in distribution of information.

Tom Mace from UI seemed to come to grips better with the subject of real standards, but then he is in an enviable position of seemingly having a real product to sell.

All parties agree that the key entities that must be considered above all are the application software developers and ultimately the end users; it is useless to develop wonderful and creative standards if the field of play is deserted due to incompatibilities.

X Applications.

Due to the fairly recent acceptance of X, there was a noticeable lack of X applications. It is obvious that writing a sophisticated X application such as a WYSIWYG word processor is not trivial, and probably more complex than an application targeted to a particular hardware platform such as a PC or a Sun workstation, the reason being the application must handle indeterminate and unknown server problems (lack of memory, missing fonts) and other network orientated problems (response and round trip times etc), as well as the very reason that makes X unique; the ability to run the same application on a large number of different hardware graphic platforms.

Several approaches are taken to porting an application to X. Developers who designed their product to be fairly device independent usually placed an intermediate layer between the application and the graphic device handling; it was then reasonably straightforward to develop an X interface to replace this layer. The application would often undergo some basic changes to cope with the existence of a window manager, and to allow the use of the X event handling scheme.

Other developers are currently rewriting their applications from the ground up, placing a degree of trust in the stability of the X standard to provide a secure foundation. This scheme suffers from the (usually large) amount of work needed for software rewriting, but presumably benefits from better performance.

Older applications such as Uniplex II were developed around the tty interface, with graphics added for some smarter screens (e.g. Tektronix terminals). The approach taken to port Uniplex II to X was to develop an X filter to act as a smart terminal, converting the terminal sequences to Xlib calls, and filtering the X events back to the core application.

A number of software developers are releasing X based 'graphic shells', which provide a friendly, Mac like interface to Unix. Combined with the low price of X terminals, it is certainly a extremely cost effective and attractive solution as opposed to building a PC or Mac network. If a large vendor such as DEC or IBM bundled such a package to give Unix a visual interface onto multiple screens for their more powerful hardware, then it may do for Unix what the PC did for personal computing.

X based tools for developing graphical user interfaces provide a means for custom tailoring applications and user interfaces, without a large amount of developing. These meta-tools I envisage being used by large scale software developers to easily build new interfaces. It provides an impetus for the developers to concentrate on providing the solutions, rather than developing their own front ends.

It was also clear in the panel sessions that X applications must provide a greater level of robustness than the less commercially orientated X clients in the public domain, which typically do not handle, or handle poorly, the issue of screen resolution difference, server errors (out of memory etc.), font differences, Inter Client Communication Conventions (ICCC) etc.

To aid in the porting from older style windowing systems, some vendors such as Sun are providing migration tools such as XView, in an attempt to lever the software effort and crank out real applications based on X.

1990 will probably be the year that X based applications will start to hit the market in a much bigger way then they have up to now. Due to the wide acceptance of X, platform dependent software will fade, and X based clients will allow the easy mixing of a broad range of graphics hardware and applications. I heard of a number of packages that are being rewritten for X, or being ported to X, such as CASE tools, word processing software, CAD tools etc.

X Terminals.

X terminals are a fascinating market segment, and the panel discussion on X terminals was in my opinion the highlight of the conference. A number of factors has contributed to the rapid definition and growth of this area, such as the acceptance of X as a standard, the use of Ethernet and TCP/IP as a communications backbone, and the price advantage over more powerful workstations. One quote succinctly summed up the situation as "Ethernet is the RS-232 of the 90's, X is the ASCII of the 90's, and X terminals are the dumb ASCII terminals of the 90's" (The quote went on to say that "NeWS will be the EBCDIC of the 90's").

In many ways the growth of X terminals parallels the early growth of the workstation market, especially when users are starting to request more memory and more power in their display. The difference is that the X terminal will only be a display server, and will not support X clients (except perhaps a local window manager or telnet client). The X terminal vendors saw their market window cutting off at half the price of a workstation; as the price of workstations continue to decrease, the X terminal market may well be severly constrained. One vendor saw that the 3M principle applied i.e. an X terminal will have a 1 Megapixel display (monochrome), a 1 Mip processor, and a Megabyte of memory, usually with some ASICs to handle blitting.

The second half of this year should see the introduction of colour X terminals, but due to the cost of display hardware, it is likely that it will be more cost effective to have a workstation instead of a dedicated X terminal.

Critics of X terminals claim that network performance will suffer unduly when a number of X terminals are placed on a net, but it was clear from the presented network loading statistics that X terminals present a minor loading. A much more obvious imbalance is the CPU performance loading when you have a large number of sophisticated clients. A catalyst for the growth of X terminals will be the provision of applications that effectively use the display characteristics without being server bound.

Another problem for X terminals is the current reliance on TCP/IP for the transport mechanism, which in all reality precludes the X terminal from running over a slow communications channel (e.g. a 9600 baud modem link). SL/IP was not considered to be a fast enough link for X. A novel approach to this problem was shown by GraphON Corp., which runs the bulk of the server in a host, and communicates to a graphics terminal using a proprietary protocol; the solution worked very well from the performance viewpoint.

X terminals may signal a fundamental change away from the distributed workstation concept, to a ‘super timesharing’ system, which has a powerful central CPU running a number of X terminals over a LAN; this has the advantage of easy system administration, centralised data storage (ease of backup etc.)

and appeals to the empire builder in all of us. Indeed the release of an X terminal from DEC may herald a sneak attack on the workstation market, especially if the huge installed base of VAXen is addressed using low cost X terminals, and a friendly and smart user interface.

X on the PC.

It is difficult to see where the PC will fit into the X world. A number of vendors sell X servers for the PC, but the fundamental problems are that a PC needs to have a network connection, and that the server usually has to run under some form of multitasking. The most common approach to running X was to embed the majority of the X server code in a dedicated graphics board, to which a simple PC resident program would feed all the requests that came in via the network. It was obvious that a PC with a memory resident X server, and network connection, left little memory or CPU time for any other processing to occur. The smaller resolution meant that some applications (such as Frame) had some limitations in the display capabilities.

Why is there interest, then, in running X on a PC? Simply because the installed base of PC's is so great, and that a lot of those PC's are on networks, and may wish to directly access Unix based X clients, and also run local PC applications such as Lotus (and Larry).

Window Managers.

Window managers are in the same category as editors; everyone has their favourite, no two are the same, they are usually customised to an unrecognisable degree, and everybody thinks that extra features should be added. It is a testimony to the basic philosophy of X that window managers are many and varied, and you can terminate the current one, and start a new one at any time.

There is a strong move to migrate some window management into the server for speed, and this is evidenced in the X11/NeWS server, which has part of the NeWS window manager in the server.

I attended a presentation at DEC WRL by Colas Nahaboo of Bull Research on GWM (Generic Window Manager). GWM is the Emacs of window managers, able to be extended via a lisp style language, providing a large degree of customisation. Interestingly enough, the project under which the software was written was named KOALA, hopefully not because they were out on a limb.

A number of vendors were selling their own window managers, usually combined with their graphical front ends. Window managers are also used in conjunction with the Look and Feel specifications, such as MWM (Motif Window Manager). Along with all the other window managers such as AWM, GWM, UWM, XWM, TWM and WM, I suspect you need to get in early to name your own WM, as there are only 26 letters in the alphabet.

Server Issues.

Some of the more technical presentations oriented around the X server, and most of them were stimulating and informative. The feedback from the Server Internals tutorial was disappointing; the technical presentations were better. Joel McCormack from DEC explained the great lengths he went to achieve 60K characters per second on the PMAX 3100 dumb frame buffer. Another speaker from DEC described porting the X server to a smart frame buffer (i.e. a display with a dedicated graphics processor). At the figures shown, it was questionable whether a graphics processor is an advantage or not, as often the overhead of accessing and formatting the requests is higher than if you did the operation using the main processor. Certainly it is a loss if your hardware doesn't quite do the 'right thing' e.g. draw lines ending on the wrong pixel.

It was considered that to achieve the goal of a working server, the server must satisfy the X protocol, and it must be reliable and robust. In the example of the DEC server, it was estimated some 15000 lines of maintainable code was developed, and some 2 man years of effort. It is key to have a fast and reliable server, as performance sells the product.

Richard Stallman prompted a fair degree of brainstorming when he suggested some improvements to X and the X server. One of the major problems is the dependence on fast round trip times for mouse movement; it was suggested that a programmable server would provide extensible functions to handle a portion of window management, and also handle mouse movements locally. Also a novel approach was the concept of pie shaped menus, with the mouse being centred and the the direction of movement selecting the option; thus was born the notion of ‘click ahead’, allowing a number of mouse selections to take place without the appearance of the associated menu.

Lightweight processes should prove to be a boon for X servers, with a thread for every window providing a more elegant design and perhaps easier porting and scalability of X servers. R4 may well have some server work using LWP, and will be interesting to see.

The encouraging element was that ideas are being considered for the evolution of X and the X server, and whilst some are far fetched, a few will be incorporated into the protocol and the servers, and will continue to improve X.

Sun and X

I had arranged with Sun Australia to visit Sun’s Mountain View HQ, but I was told the morning of my departure that all the engineers would be on a holiday, so I went to DEC instead.

With the release of the beta X11/NeWS server, and XView, Sun Microsystems may well have said ‘if you can’t beat them, join them’, but from the demonstrations and time I have had to experiment, NeWS and X work very well together. Whilst the die-hards contest that NeWS is technically superior than X, and provides a better foundation for the future, X is here to stay, and Sun have responded to that.

A number of technical features of the release are worth examining. Without question the use of NeWS as an imaging model is an advantage. It highlights the role that PostScript (Display PostScript or NeWS) has in functioning as an imaging engine on top of X. I’m not sure what future NeWS really has in the X world (will it really be the EBCDIC of the 90’s?). NeWS has some fundamental differences to X, and it may just be that both benefit from the marriage i.e. that some cross fertilisation will result.

I have to be honest and say that I do not think it is possible for NeWS to even contemplate overtaking X at this stage, due to the current impetus that X has, but perhaps X may be improved to operate with some of the nice features of NeWS, such as the concept of a programmable server, with lightweight threads for each window, and perhaps some embedded imaging in the server. NeWS may be viewed as a graphics engine running on top of X, giving access to sophisticated imaging, with some form of filtering to allow it to operate on generic X servers.

NeWS will probably fall into the category of some of the extensions to X (XIE, PEX, VEX), which not all users will operate, and not all servers will support. This is not to belittle NeWS, as it is obviously an extremely powerful and elegant system. It is probably a testament to the difficulty involved in merging the two diverse systems, and to the complexities of NeWS, that Sun has slipped the shipping schedule of X11/NeWS back to August.

The performance of the merged server is quite impressive, also showing how much effort has gone into making it work. I have played with the server on a Sun Sparcstation 1 with the GX graphics accelerator, and it is *fast*. I was told the X server would run some 5-7 times faster than the MIT distributed server on a colour Sun 3/60, which was nice to hear for those amongst us who have experienced the latter. My bet is that most people will use the server as an X server (except for existing NeWS users), rather than migrate applications to NeWS based displays.

XView may well be the dark horse of toolkits, as it is one of the only ones not based on the Xt intrinsics (widget based) toolkit. The object orientated programming interface is friendly and powerful, and providing SunView compatibility is a smart move (and very important). I liked the Open Look based Look and Feel of XView, and it was easy to drive; XView will be part of the contributed software to R4, so it will be interesting to see if it will be ported to many more machines. Sun themselves have ported the toolkit to (I believe) a DEC workstation to prove to themselves it is easy to port.

Miscellaneous Issues.

A number of fringe issues were discussed, which I won’t go into detail on. They included internationalisation, with emphasis on text encodings, and standard symbol and icon definition (just what is the international symbol for a program?); porting X to VMS and Primos; different language bindings; Open Fonts in X; scientific visualisation.

Integrated Computer Solutions (the conference organisers) has given preliminary permission for reprinting conference papers in AUUGN, and I will try to obtain some of the key papers when the conference proceedings finally arrive.

Summary and Conclusions.

X has reached the level of accepted standard, and it is clear that virtually all graphic or workstation orientated applications will eventually migrate to X and allow independence from vendor hardware and software, similar to the manner in which Ethernet and TCP/IP allows interconnectivity of diverse architectures and vendor equipment.

Since X is considered stable at the protocol level, we are seeing higher layer standards emerging, albeit not without some labour pain. The jury is still deliberating on Look and Feel, and hopefully reason will prevail.

There is a proliferation of toolkits, and window managers, and this will only serve to stimulate ideas and growth in the market, as users and software developers build on the developments.

The acceptance of X has created new markets, such as X terminals, and it is worth observing how the larger vendors will respond to this potentially huge market. The only thing stopping X is a lack of application clients, which are coming in the near future.

X is also an evolving animal, and we must give credit to MIT and the designers of X that allow it to do so without major havoc. It is also a credit to the developers of X, both educational and commercial, that X is an *open* system, in the true sense of the word open.

A final conclusion is that micro-breweries are a redemption to U.S. beer, and you'll regret ever attending a Palo Alto Chili Cook-Off.

Boring Bits at the End.

The opinions expressed herein are mine alone, and I apologise profusely for any misquotes, or information that may be incorrect.

Open Look is a trademark of AT&T.

The X Window System is a trademark of MIT.

OSF/Motif is a trademark of OSF.

DEC is a trademark of Digital Equipment Corp.

PostScript and Display PostScript are trademarks of Adobe Systems.

XView, NeWS and Sparcstation are trademarks of Sun Microsystems.

* is a wildcard trademark of [matching organisation].

Appendix A

Tutorial Summaries

Below are summaries of the tutorials; some are from memory, others from the notes and word of mouth feedback. I have a complete set of tutorial notes if anybody has a particular subject they would like to follow on with.

Programming the X Window System.

Basically an introduction to X, aimed at new X users with little or no experience with X. I attended, but hoping to get a more formal grounding in Xlib and the standard toolkit, but found it too shallow, and too many interruptions from the attendees to maintain a good flow.

Using Widgets.

Describes how to program using the Xt standard toolkit. Widgets are class orientated X objects, used to provide a meta level of X programming.

Object-Oriented Programming with C++.

Nothing to do with X *per se*, but C++ is a natural language for manipulating graphics based objects, so we should see a growing interest in C++ in the context of X programming.

Programming User Interfaces With InterViews.

InterViews is a C++ based toolkit for X, developed originally at Stanford (I believe).

Creating Andrew Applications.

The Andrew Toolkit was developed at Carnigie Mellon University, and is part of the Andrew Development Environment Workbench; it is guaranteed against viruses.

Colour.

A tutorial on the intelligent use of colour in graphic displays, and how colour is used and implemented in X.

Getting In Tune With Motif.

A detailed look at Motif the toolkit, Motif the style guide, Motif the user interface language, and Motif the window manager.

Server Internals.

I was told that this tutorial was not very worthwhile, so I'm glad I didn't go. It basically was a summary of the MIT standard server(s), and the internal structure therein.

Widget Writing.

How to write a Xt toolkit widget.

Writing Portable X Code.

Ralph Swick once said that there is no such thing as portable code, only code to be ported. This tutorial aimed to give guidelines on how to write clients to handle different server and host constraints.

Fundamentals of Interactive Computer Graphics.

Where have I heard that title before? Describes the general concepts behind computer graphics. The tutorial notes has the best set of graphic references of books and papers I have ever seen.

User Interface Designs.

With the growth and sophistication of graphics, it is becoming much more important to have an interface that works with the user rather than against them, and this tutorial aimed at identifying the elements that comprise a good user interface.

XView, An Open Look Toolkit.

An enjoyable tutorial, as it gave insight into the design processes behind XView; aimed at the program interface level.

Tour Of The Xt Intrinsics.

The Xt toolkit is considered almost part of the X standard, as it gives a higher level of programming abstraction than Xlib itself. Basically described the design and available facilities of the toolkit.

Application Development With The AT&T Open Look Environment.

Describes the three major portions; the Window Manager, the Workspace Manager, and the File Manager.

Inter-Client Communications Conventions.

The ICCC describe how clients can cooperate in window and session management, sharing data, and using the X server as a common point of communication. The ICCC are a fertile area for growth, in providing a generic IPC mechanism, and in allowing sophisticated object sharing.

Display PostScript.

Describes how Display PostScript can be used an imaging engine on top of X.

PEX: The 3D Extension To X.

The extensions to X which give PHIGS/PHIGS+ 3D rendering within the X environment. I have the PEX draft documents for anyone who wants a copy.