Andrew Consortium
School of Computer Science
Carnegie Mellon

The Andrew View

Volume 3, Number 1
April, 1994

The big news this month is our new public release !

AUIS Release 6.2

For members, the most interesting innovation may be the **html editor** mode. You can now edit html files and see them looking as nice as AUIS files. For non-members, innovations beyond release 5.1 include:

prefed - preferences editor
figure - drawing editor
image - display embedded images
bdffont - font editor

AUIS is now supported on Telmat, Linux, Solaris, and SGI in addition to platforms from IBM, SPARCstation, HP, Dec, Apollo, and SCO.

FTP We have a new ftp home in directory pub/AUIS on ftp.andrew.cmu.edu (ip address: 128.2.35.66). See the README file for details.

WWW View us on the World Wide Web at URL (all one line):

http://www.cs.cmu.edu:8001/afs/cs.cmu.edu/project/atk-ftp/web/andrew-home.html

Contents

My Own View

Wilfred J. Hansen, Director

When you consider that AUIS is a million lines of code in over 5000 files, you might think it a miracle that the whole thing ever compiles on any platform. Well, it compiles--really, it does! Our own platforms, however, are only a minority of all the platforms we support. A number of kind people have built the 6.2 release on other platforms and reported problems. The result is the new patch, 6.2.1 for Sun platforms, as reported below.

There will be at least one further release of AUIS 6, numbered release 6.3. It will come out with the contributed section of X Windows System V11R6 in June and will be the last AUIS release in C. Subsequent releases will be in C++ and numbered 7.1 and higher.

At SIGCHI (April 24-28, 1994, Boston, Massachusetts) I will present a tutorial entitled "Introduction to User Interface Systems for HCI Developers and Researchers". This will discuss the Human-Computer Interface challenges and opportunities of user interface systems like AUIS, Fresco, OLE from MicroSoft, and OpenDoc originally from Apple and now from the Component Integration Laboratory. These systems offer the challenge of defining user interface schemes that extend across a wide spread of applications; they offer the opportunity to conveniently capture user interactions in order to study user behavior.

I’ll repeat the systems comparisons portions of the tutorial at Xhibition in San Jose (June 21-23) and at the Andrew Technical Conference (Pittsburgh, 1 July).

Andrew Technical Conference and Annual Meeting

The Andrew Technical Conference this year is scheduled for

Friday, July 1

The Annual Meeting of the Consortium will be held during and after lunch.

As part of the Conference, Fred Hansen will present the tutorial he has been preparing on the various alternatives available to AUIS: OLE, OpenDoc, Fresco, and so on.

If you have a paper you would like to present, please contact Fred Hansen (wjh+@cmu.edu, (412) 268-6788).

As part of the annual meeting, we discuss future directions for development of AUIS. These discussions drive the work of the Consortium staff for the year ahead. If you have any suggestions as to directions we should take, please send email to Fred Hansen at the address above.

New and Revised Features in Version 6.2

AUIS users everywhere can now receive Release 6.2 of AUIS. The full set of changes between 6.2 and 5.1, the previous public release, is given as the file Changes (or Changes.ez in AUIS format) in directory pub/AUIS of our new ftp host:

ftp.andrew.cmu.edu

For members, here are the changes from 6.1 to 6.2.

Everywhere

Fixed bugs, coreleaks, and copyrights. More posixification. AUIS code is posix, but utilizes some BSD extensions.

- Straightened everything out so colors can be destroyed without leaving dangling pointers anywhere.
- Fixed event queue so events are not lost. This means the clock will not stop.
- Revised ATK text data stream slightly so whitespace is treated better when text is converted to ASCII.
- Fixed ez so that ˆG to the initial file prompt will provide a scratch buffer.

Platforms

AUIS is now supported on the following new platforms (the names with underscores are the subdirectory of the config directory in the source).

Linux (i386_Linux) Telmat (telmat_svr4) Solaris (sun4_51)

In addition, support for SGI machines has been fixed:

SGI (sgi_4d)

We have tested the system on these additional platforms

pmax_mach pmax_ul4 rs_aix3 sun4_mach sun4_41 hp700_80

The following platforms are included in the release, but have not been tested

apollo68k hp300_80 hp800_80 i386_mach mac2_51 next_mach20 pmax_3 pmax_41 ps_aix12 rt_aix221 rt_aos4 rt_mach sco_i386 sun3_35 sun3_4 sun3_41 sun4_40 vax_3 vax_43

Applications and Objects

Changed background colors in scollbars, dialogs, sbuttons, menubars, and Motif-style popups. (Can be turned off by setting preference UseNewShadows to No.)

Updated text to work with Ispell 4.0.

Ness has been revised throughout so that very general filenames can be used, including $XXX to access environment values. Files for execution are sought in the directories listed in the nesspath preference.

Ness's launchapplication function can now launch applications that are objects other than text.

HTML editor mode has been added in ./contrib/srctext/html. Files can be saved in html format and edited in ez as formatted text; the reader need not look at embedded html formatting tags. To enable this, the contrib directory must be built and installed and then a line like

addfiletype .html html

added to ~/.ezinit. (For instructions on the latter, say 'help initfiles'.)

Translation between AUIS format and RTF. The source files contrib/mit/2rtf/* and contrib/mit/rtf2/* are installed as the applications 2rtf and rtf2; these convert, respectively, from AUIS to RTF and from RTF to AUIS. (Sometimes, conversion from RTF to AUIS is better done with the ness program fromrtf.n. See src:atk/ness/nesslib/fromrtf.n or $ANDREWDIR/lib/ness/fromtrt.n.)

New facilities

The Andrew logo is available as an icon in src:./overhead/xicons/alog.icon.

Support has been added for translating menu items and message line messages into other languages. This is enabled by setting the TranslateMenus preference to On, in which case, menu titles and items are translated according to the ANDREWLANGUAGE database. Documentation and example files for this mechanism are unavailable. (This feature is used primarily by the port to the Telmat platform.)

All the srctext insets such as the C and C++ modes now understand dogtags. See atk/srctext/dogtags.help (or help dogtags) for more information.

When porting the system to a new platform, it has been necessary to write an assembler routine, entry.spp. This can now be avoided by defining CLASS_CTRAMPOLINE_ENV as 1 in system.h for the platform.

AUIS Release 6.2.1

Release 6.2.1 of the Andrew User Interface System is now available as a patch to release 6.2. If you already have 6.2, we recommend that you pick up 6.2.1 only for Sun OS 4.1.* and Sun Solaris. It fixes a number of minor bugs found in release 6.2 but breaks on non-Sun systems. Another patch which will be generally applicable will be made available soon.

The patch is available as file patch-6.2.1.tar.Z via anonymous ftp from ftp.andrew.cmu.edu (IP 128.2.35.66) in the pub/AUIS/patches directory. This patch introduces several new files as well as moving the <sources>/atk/syntax/bison directory to <sources>/overhead/bison. It utilizes the new parsing technology described in the Bison article below.

The "unknown" inset: an upgrade for AUIS 5.1

Rob Ryan

When AUIS release 5.1 encounters a file create with release 6.xx containing new insets, problems will occur, possibly including core dumps. Ideally, you will avoid this problem by upgrading to AUIS 6.2.1 or some later system, but if you would rather not, you can install the "unknown" inset. (NOTE: This is only for 5.1.)

The "unknown" inset distribution introduces proxy insets having the same names as the new insets, but functioning only enough to inform the user that the inset is not supported. This package is available for anonymous ftp from ftp.andrew.cmu.edu in directory pub/AUIS/5.1-SafetyNet.

Projects in Progress

The Consortium staff is at work on three major projects undertaken in response to past annual meetings.

C++. The port to C++ is largely complete, including complete plans for dynamic loading; but the latter is still pending implementation. Progress is on hold this month while we complete release of version 6.2. We expect to make a complete release of the C++ version later this year.

Printing. In the C++ release, text and many other objects now print directly to postscript. Many niceties remain to be added, but the new printing facility should be ready for the C++ release.

Widgets. Work on the C++ version has occupied more resources than we had hoped. (It was never actually scheduled at an Annual meeting.) The result has been complete slippage of the widgets project. We have some exciting concepts, but have not made implementation progress.

WorldWideWeb. In our first issue of last year, the **Andrew View** reported on the WorldWideWeb. In the last few months usage and contents on this web have shown phenomenal growth (in the range of 70,000 percent at one point). We are working toward closer integration of AUIS into the WWW environment, having made a start by creating our own home page, as noted above. Two AUIS projects are related:

Nick Williams of the Systems Architecture Research Center in London has created htmltext, a source text editor which simplifies construction of files in the HTML format utilized on the WWW. An early version of this package is in AUIS 6.2; more recent versions are available for anonymous FTP from ftp.cs.city.ac.uk in directory pub/htmltext.

Tom Neuendorffer, who created ADEW and many other Andrew components, has agreed to create an AUIS application which will serve as an interface to the WWW. This will have many advantages for AUIS users:

> familiar AUIS interaction: scrollbars, menus, window resizing, ...
> AUIS search operations
> copy / paste from HTML documents to AUIS documents
> open-ended - add your own operations

Another desirable goal is creation of a "weblink" inset which works like the existing link inset, but can point to any WWW node. We also hope to create a WWW server which reads AUIS documents and delivers them across the web as HTML documents.

Whither Troff Printing?

Andy Plotkin

As part of the development of the new PostScript printing system, I am considering the fate of all the troff-generating routines currently in place. There are three obvious possibilities:

1. Remove all the old printing routines, including the view::Print method.
Ups: The source code and executables would only have one set of printing routines, which saves space. Ezprint would subsume the functionality of ez2ps.
Downs: Troff output would no longer be available at all.

2. Put all of the printing code under cpp conditionals; say, #ifndef and #ifdef PSPRINTING_ENV.
Ups: Sites could choose which method to use. Executables would have only one set of printing routines...
Downs: ...but the source code would have both. Users would not be able to choose a printing method themselves.

3. Leave everything in place. The print::ProcessView function already accepts a flag which tells it whether to generate troff or generate PS output (using psdit); that mechanism could be altered to choose between the troff and PS printing routines.
Ups: Users could invoke either printing method (probably by either a preference or different keybindings.)
Downs: More source-code and executable bloat. Twice as much printing code for the Consortium to support.

This decision will impact all Consortium members, since everyone uses printing (even people who don’t use "impact" as a verb.) Please send us your thoughts, so that we can take your needs into account.

Proposed Grant Levels

The Andrew Consortium exists through the generosity of our members, who provide grants on our behalf. For the first three years we have had only three levels of membership--Full, Contributing, and Associate. In some cases these levels have not been appropriate; primarily in being beyond the means of small, start-up companies who might otherwise participate for mutual benefit. At the last Annual Meeting, it was proposed that a new grant structure be introduced offering a greater variety of levels.

This note describes a proposed structure of categories. The idea is to fit existing members into one or another of these categories at a grant level no more than slightly higher than their current level. If you have suggestions about this proposal, please contact Fred Hansen (wjh+@cmu.edu, +1-412-268-6788.)

The plan proposed below will be voted on at the annual meeting. Grant levels, expressed in $000's, depend on size of organization and class of membership.

Size of organization. Organizations are classified as small, medium or large, depending on the type of organization. This section defines the size of a "medium" organization of each type. "Small" or "large" organizations are, respectively and perhaps not surprisingly, smaller or larger than medium organizations.

Academic institution, medium size: 2000-10000 students
Non-academic research organization, medium size: 8-40 researchers
Commercial organization, in house use, medium size: 100-1000 users
Commercial distributor, medium size: 1000-10000 customers
Government, medium size: 200-2000 users

Participating Members. A new grant level is proposed for organizations desiring direct involvement in AUIS development. Participating members have all the rights of full members and are also eligible to send staff to Carnegie Mellon where they can work side-by-side with the Consortium staff. In this way they can more immediately influence AUIS development, they can develop new software based on the latest version, and they can acquire the intimate understanding of the system which is only possible from immersion. Grant levels: 150, 300, 600.

Full Members. These members have voting rights in consortium decisions and merit direct contact with Consortium staff. Grant levels: 45, 90, 180.

Contributing Members. Contributing members have the right to specify particular code to be included in the AUIS distribution. This level of membership is also appropriate for organizations which depend on AUIS for their own work. Grant levels: 15, 30, 60.

Associate Members. At this level, members establish a relationship with the consortium, and get access to the latest source. Grant levels: 0.5, 2, 5.

1994 Survey

Susan Straub

Once again we are polling the world to see how people are exploiting AUIS and to determine their needs. If you've not already filled out this survey, please take a few moments to complete the enclosed survey and return it to susan+@andrew.cmu.edu. We will share the results of the survey at the Annual Meeting.

(It is not our intention to publish individual answers other than to quote comments; however, we are not taking cryptographic precautions and cannot guarantee the confidentiality of what you send us. If we've asked questions you'd rather not answer, please feel free to omit them from your reply.)

~~~~~~

1994 Survey of Andrew User Interface System Users

~~~~~~

Name:

Net address:

Company/division/site location:

Number of machines at site running AUIS:
1? 2-10? 10-100? 100-1000? over 1000?

Number of AUIS users at site:
1? 2-10? 10-100? 100-1000? over 1000?

Number of AUIS users at other sites for the same company?

What system type(s) do you run AUIS on:

Are you or anyone at your site using AUIS for:
Mail, bulletin boards, news groups access?
Document production
User interface prototyping or development
Other uses (please list)

What additional service(s) would you like to see provided by the Andrew Consortium?

Other comments:

Bison for C and C++

Bison is the GNU parser generator; an alternative to yacc. To enhance AUIS portability, we have adapted Bison for all parsing requirements in AUIS 6.2.1. To avoid encumbering the entire source code with the license imposed by the Free Software Foundation, the adaptation removes the default Bison parser and installs an AUIS parser. The C++ version even creates a Class from the grammar. The package is available as a separate entity from ftp.andrew.cmu.edu in directory pub/AUIS/bison.

The parser generators have these advantages

No symbol conflicts. Each parser is represented by a single name.

Shared parser code. All grammars use a single instance of the parser object code.
C++ object. For C++, the generated parser is an object. Utilize it with
GrammarName *grammar = new GrammarName;
License free. The license on the parser code permits effectively unhindered use.

Bison-A2.2 has these advantages over other versions of Bison:

Errors in the grammar do not terminate input; all errors are found in one pass. A complete example and test case is the file mess.y in the distribution.
Tokens may now be specified as multiple-character strings: "<=" can appear where formerly would have to be LESSEQ.
The -n switch produces the parser tables without including the parser; a project can now use its own parser and avoid the GNU license for the resulting application. (The AUIS C and C++ parsers do not actually use this switch.)

AUIS also includes a lexical analyzer generator--gentlex in directory <sources>/atk/syntax/tlex--tailored to this version of bison.

Rainbow

Some Consortium members have expressed interest in adopting SGML. The following extract of a note on "Rainbow" describes one tool available for conversion from existing formats to some particular SGML/DTD. We could, for example, support conversion of AUIS documents to SGML via this path. (Howver, AUIS documents to a large degree already have the structure information descrbed below.) Or we could support conversion of other documents to AUIS via a pathway through SGML using these tools. If either of these efforts is of value, you should include it in your list of suggestions for the Annual Meeting, as noted above.

Accelerating Conversion to SGML via the Rainbow Format

David Sklar
Director of Applications
Electronic Book Technologies

***** The Problem: Inefficiency in Conversion Efforts**

"Up conversion" -- the translation of a document from a proprietary word-processor (WP) format to an SGML document conforming to a useful DTD -- is one of the thorniest problems an organization faces when it adopts SGML. Up conversion is typically performed by building (or hiring a consultant to build) a custom conversion application via a translation-enabling tool like OmniMark, TagWrite, or FastTag. The conversion application typically involves two phases: 1) extraction and interpretation of the formatting codes in the WP format, and 2) identification of content and structure.

The second phase is the most sophisticated one, for it involves creating something (structure and true content identification) from "nothing" (WP formats which are typically flat and lacking in content identification). A considerable amount of planning and thought is necessary for implementation of this phase, and it is almost always necessary to custom-build this phase for each organization.

But the first phase -- extraction of the WP formatting codes -- is not very sophisticated, for it is simply a translation. The most difficult part of implementing this phase is becoming an expert in the proprietary WP format itself. .... How can we prevent organizations from redundantly implementing phase one? Imagine a world in which all up-conversion efforts start from a single stable, well-documented input format instead of a variety of unstable proprietary ones....

***** The Solution: Rainbow**

Electronic Book Technologies, in conjunction with several other key SGML vendors and promoters, has designed a format that is suitable for acting as the starting point for up-conversion efforts. We call that format "Rainbow", because it represents a unification of the wide spectrum of proprietary formats.

The Rainbow format is actually an SGML DTD.... The Rainbow DTD is publicly available, and can be used and modified by organizations and individuals freely. An FTP server has been created to provide a forum for the distribution of Rainbow-related data, and a mailing list has been created to keep interested parties informed of Rainbow-related developments. To subscribe to the mailing list, send email to rainbow@ebt.com; you will receive a reply with specific information on how to access the FTP server.

As described above, the primary purpose of Rainbow is to insulate up-conversion implementations from the dynamic and eccentric world of proprietary formats. Obviously, this noble idea is not practical unless the SGML community has easy access to "Rainbow Makers" (software programs designed to convert WP-format documents to Rainbow documents). It is our hope that members of the SGML community will share Rainbow Makers that the build, thus helping us eliminate redundant efforts. [EBT is making available Rainbow-Makers for RTF (Microsoft Word), MIF, Interleaf, and WordPerfect. Another vendor is doing one for the Ventura format.]

The Andrew User Interface System

The Andrew User Interface System (AUIS) is a word processor for compound documents, a collection of related applications, and an environment for building new applications and embeddable objects. AUIS runs under X11 and provides a dynamically-loadable object-oriented environment wherein objects can be embedded in one-another. Thus documents can contain not only multiple fonts and other typography, but also embedded raster images, spreadsheets, drawings, equations, simple animations, and so on. These embedded objects may themselves contain other objects, including text. The release includes many objects, including those mentioned above, along with a help browser, a directory browser, a font editor, a system monitoring tool, an editor-based shell interface, and source-text editors for C, C++, Lisp, HTML, Pascal, and other languages.

The underlying Andrew Toolkit architecture (ATK) supports cooperation between objects not only for screen display, but also for file storage, cut/paste across windows, an application construction environment, an extension language, and printing. There is full support for programmers to create new objects and new applications.

The Andrew Message System(AMS) is a component of AUIS and provides a multi-media interface to mail and bulletin-boards. AMS contains many advanced features including authentication, return receipts, automatic sorting of mail, vote collection and tabulation, enclosures, audit trails of related messages, and subscription management. It also provides a variety of interfaces that support character-based terminals and low-function personal computers in addition to high-function workstations.

Andrew has been successfully used on most platforms from these vendors: IBM, Sun, DEC, HP, SGI IRIX 4.0, Apollo, Macintosh II MacMach, 386/486 Mach, Linux, Telmat, Apollo, Mac2, Next, SCO I386.

The following are components are available in version 6.2 and later versions. Some respond to frequently requested applications in X: word processor-ez, drawing editor-figure, mail and news reader-messages, font editor-bdffont, documentation browser-help, directory browser-bush, HTML editor-htmltext.

AUIS editors that are both applications and insets

eq - equation inset
fad - animation editor
figure - drawing editor
layout - inset for arbitrary layout of enclosed insets
lset - display two adjacent insets
ness - extension and string processing language
org - display and edit hierarchies
page - allow flipping between pages
raster - editor for monochrome bit-mapped images
table - table / spreadsheet inset
text - text, document, and program editor (the heart of ez and other applications)

AUIS applications

bush - directory browser
chump - schedule maintainer
console - shell interface / terminal replacement

ez - word processor and program editor
bdffont - font editor (for fonts in bdf format)
help - documentation browser (includes all AUIS help files)
launch - provides a menu of AUIS applications
messages - mail and news reader, manager, and composer
pipescript - viewer useful as stdout
prefed - preferences editor
sendmessage - application for sending a message or news posting
typescript - shell interface (terminal substitute)

Commonly used non-interactive AUIS applications

datacat - concatenate AUIS files
ezprint - print an AUIS document
nessrun - runs a Ness script (the extension and string processing language)
preview - preview an AUIS document on screen

File format converters

from AUIS to: RTF, ASCII, PostScript, troff
to AUIS from: ppm, Scribe, RTF, troff, X window dump
convertraster - various raster formats

Embeddable insets not usually used as applications

clock - analog clock
eq - equations
fad - rudimentary animations
header - specify document headers and footers
image - store and display images in JPEG, gif, tiff, and other formats
link - hypertext-like link
month - display a calendar for one month
note - annotation
text - the text object itself
timeoday - digital clock
writestamp - displays the time the file was written

Editing tools

compchar - type characters for European languages
compile - support compilation and error review
complete - filename completion for typescript
dired - directory browser
dsearch - dynamic search
ezdiff - compare two ASCII files
filter - process a region of a document through a shell command
isearch - incremental search
lookz - style table editor
spell - spelling correction (uses ispell)

Source text editing tools for

assembler, C++, C, HTML, Lisp, man pages, Modula, Modula-3, Pascal

Tools for constructing insets and applications

arb - ADEW application builder interface
bison, mkparser/parser - GNU parser generator (modified)
class - preprocessor for AUIS's ClassC version of C
createcon - ADEW tool to generate C code for interface
createinset - create a prototypical inset under a new name
genmake - Makefile generator for classC
gentlex - lexical analyzer generator
makedo, doindex, whichdo - create, store, and locate dynamically loadable objects
runadew - ADEW application builder

Additional insets available for building applications

button, display text string, entry of labeled text string, four position switch, multiple buttons, multiple sliders, multiple string entry, on/off switch, slider, slider controlling array of strings, text list, thumb knob

---

Consortium Services and Offerings

For information about services and offerings of the Andrew Toolkit Consortium please contact us at:

Andrew Consortium
School of Computer Science
Carnegie Mellon University
5000 Forbes Avenue
Pittsburgh, PA 15213; USA
phone: +1-412-268-6710
email: info-andrew-request@andrew.cmu.edu

We offer memberships at three levels: Full, Contributing, and Associate. We also can provide source tape, CDrom with both source and executables, bibliography, copies of papers, conference proceedings.

AUIS sources and binaries are available online via anonymous ftp from **ftp.andrew.cmu.edu** (IP address is 128.2.35.66) in directory ./pub/AUIS. See the README there. Another ftp site is **export.lcs.mit.edu** in directory ./contrib/andrew; this is cloned on many servers, world-wide. On the nationwide AFS file system, AUIS is available in **/afs/cs.cmu.edu/project/atk-ftp/.**

To try AUIS, use our Remote Andrew Demo service: **finger help@atk.itc.cmu.edu** (IP address is 128.2.203.218)

News groups: Send subscription requests to **info-andrew-request@andrew.cmu.edu** and submissions to **info-andrew+@andrew.cmu.edu.** This is a distribution list offering the news with full AUIS formatting; For a clone in plain ASCII, see news group **comp.soft-sys.andrew.**

Send requests to info-andrew-requests@andrew.cmu.edu
Bug reports to info-andrew-bugs@andrew.cmu.edu