Andrew Consortium
School of Computer Science
Carnegie Mellon University
Volume 4, Number 2
Summer 1995

The Andrew View Newsletter

Highlights in this Issue

My Own View
"Software Engineering Techniques "
Enhancements in the Works
"widgets attributes architecture"
Software Developments
"Widgets Work"
"Function Locator"
"Official Patch 6.3.4"
"Contributed Port and Patch"
Member Mentions
"Andrew at MIT"
News from the Consortium
"1995 Survey of Andrew Users"
"Conference & Annual Meeting"
"Employment Opportunities"

IMPORTANT DATES TO REMEMBER

Andrew Technical Conference
September 21-22, 1995

Paper Proposals - July 25th

Acceptance - August 11th
Final papers - September 8th

My Own View - Software Engineering Techniques

Wilfred J. Hansen, Director

When last I wrote, I was in the middle of teaching a Software Engineering course to thirty-eight students here at Carnegie Mellon. We got through it--all of us learned a great deal about software engineering, Andrew, and working together.

Should the Andrew developers have adopted more software engineering techniques, and put more emphasis on specification and less on coding? Well, "yes and no." Their first steps were very exploratory; the specifications changed rapidly as coding revealed what could be done and what we wanted to accomplish. At this stage, software engineering would not have helped. Later, however, Andrew was rewritten in an object-oriented environment; I now believe that that effort would have benefitted from software engineering basics. There are a number of places where similar functions are coded in diverse manners (posting keys versus posting menus; diverse approaches to extensible lists; names of proctable functions; differing coding styles; differing comment conventions; etc). Nonetheless, a workable system did emerge and maintenance is not as terrible as software engineers would predict. Largely, I think the ease of maintenance is due to the object-oriented design: changes in any one area are relatively well isolated from changes in other areas.

Since the course, I have been cleaning up a bunch of little things to clear the way for Widgets. They are:

Bison I reworked the Andrew bison and parsers to make a few improvements--especially in installability--and to integrate the recent changes made to the GNU version. The latter no longer comes with the restriction that code produced with it must be covered by the GPL (which is antithetical to proprietary code). Consequently, we can once again try to merge bisons and avoid having an Andrew version of it. (See ftp://ftp.andrew.cmu.edu/pub/AUIS/bison.)

Batmail Batmail is an emacs interface to the Andrew Message System. It has now been adapted and released as part of the contributed code library for version 7.2. Personally, I use batmail when I am logged in from home via kermit.

Gofig Another contributed set of code for Andrew7.2 is a displayer for go diagrams. When mail arrives with a go diagram created from periods, X's and O's, a simple menu option converts it to a print quality diagram.

Xpty Maintenance of typescript has become burdensome as vendors continually produce incompatible and incorrect terminal interface code. To reduce the total maintenance load, I have extracted the pty/tty interface from xterm and produced versions of both xterm and typescript that work atop the same interface library. Remarkably, the converted version of xterm worked the very first time I tried it!

Widgets Well, I didn't finish *all* the little things I wanted to do, but we did set a deadline and have begun full-scale work on Widgets based on what was learned from the software engineering course. The first component is the resource/attribute design outlined below.

Enhancements in the Works

widget attribute architecture In implementing the new widget package, one goal is to introduce a notion of attributes to objects. In an X implementation of Andrew, this will mirror X resources. This note describes the architecture.

{...starting now, lower case has been used to emphasize that this is a preliminary specification>>>>suggestions for improvements to the proposed architecture are eagerly awaited...}

r
eview of x resource concepts

in x resource files, each resource file entry has the form

resourceidentifier : value

attribute values are strings; they are converted for use when the attribute is fetched. resourceidentifiers are a sequence of names with some wild-carding characters. when an object requests a resource value, a resourceidentifier for the object is constructed from its location in the view tree and the corresponding value is fetched from one of several sources. for instance, to make all buttons in ez have a blue foreground, we could write

ez*Button.foreground : blue

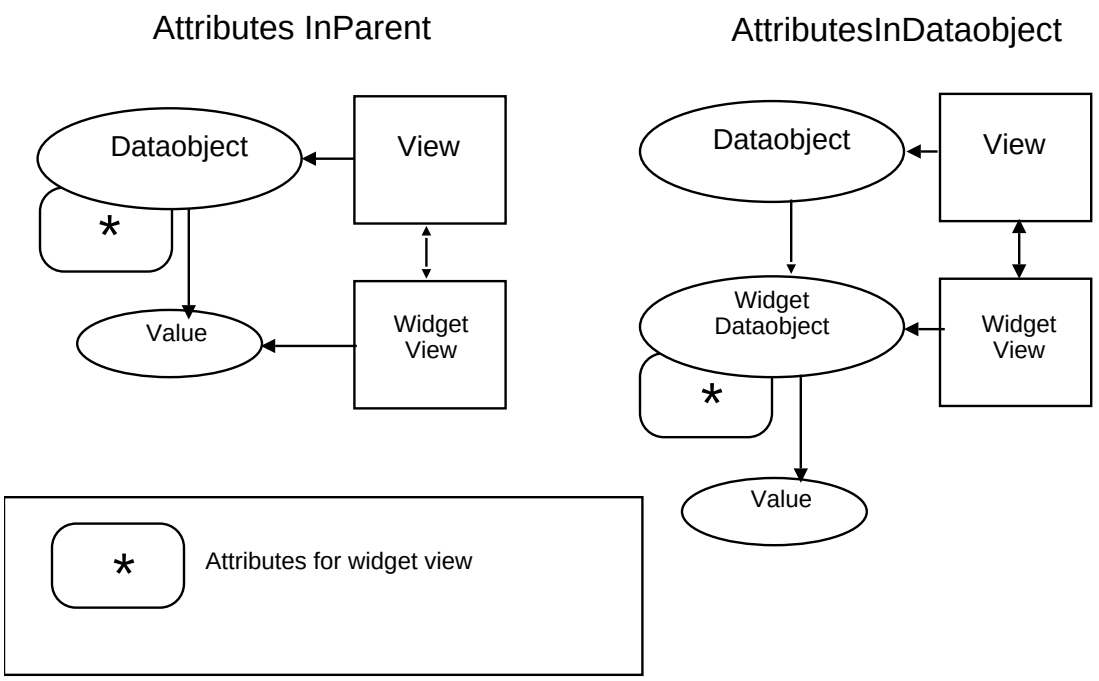
the button object in ez fetches its attribute by asking for "foreground" and the resource machinery fetches the value and converts it to a color specification.

b
asic approach of andrew widgets

in andrew, every object visible on screen is an inset represented internally with both a view and a dataobject. this permits multiple views of specific data objects. for widgets, this architecture is inadequate; there are two kinds of information to store for a view: its specific view attributes and the fundamental value that is being manipulated. consider an integer. the fundamental value is an integer, and there may be multiple views, but some views may be dials and others textual.

we considered two alternatives which can be distinguished by where they store the attributes for a view: attributesinparent and attributeindataobject. in attributesinparent, each child view requests and sets attributes by methods invoked to or by its parent view. the parent view then stores those attributes in its own dataobject. that is, every object which is a substrate capable of embedding other objects must be able to store attributes for each of its children. the dataobject for the child view is just the value which is to be displayed, be it integer, string, or some other thing.

the attributesindataobject also has value objects that just store a primitive value. in addition, each view object has a corresponding data object which stores its attributes. the two schemes are illustrated here.



at present, it looks as though we will implement the attributesindataobject scheme.

data object tree

since computation sometimes takes place on data objects without views, we currently intend to implement linkage between data objects in the form of a bidirectional tree more or less parallel to the view tree. unlike x, naming of resources will go through the dataobject tree. in most cases, this will be identical to the view tree, so we hope it will not have negative consequences for users or programmers.

search order

the resource mechanism looks in numerous places for a resource value, taking the first it finds. the sequence of searching is:

1) site immutable resource file - resources specified in this file cannot be overriden by any of the other mechanisms. this file should be quite small or non-existent.

2) command line - the usual x syntax for specifying resources on the command line is used.

3) data stream - when an author creates a widget, some attributes can be tailored to its use. these are stored in the data stream so they are fetched for readers. {we need some rule so that users can override these values. perhaps we can say that explicitly addressing an object by name (or fully qualified name) will override its datastream value.}

4) a parent in the data object tree - the author of a document may associate overriding resource values with a substrate. all children that match the specification would then get this value.

5) XUSERFILESEARCHPATH - each file and directory on this path is examined for a resource meeting the request. if the path element is a directory, the file matching the widget class is examined. typical entries on the path might be:

personal application resource file
personal global resource file
project application resource file
project global resource file
site resource file
widget specific resource file ($ANDREWDIR/widgets/)
source system resource file ($ANDREWDIR/lib/global.resources)

new data object methods

get(resourceIdentifier), put(resourceIdentifier, value)
called from view
these are overloaded in two ways:
the resourceIdentifier may be a string or an atom
the value may be one of several types:
integer, real, boolean, char *, ...
and arrays of same
(except: get cannot override based on return type, sigh.)
get implements the search order outlined above.
put will store the value into the dataobject.

putforchildren(resourceIdentifier, value)
call by view
the resourceIdentifier can be a fuzzy match as in x, but the root of the string will be this data object. this resourceIdentifier will be checked against each request from a child and the value supplied if there is a match. (this describes a system in which an object must explicitly request resource values. we could consider having a mechanism to explicitly set them from external code.)

getforchild(resourceIdentifier)
internal; called from child to retrieve the value
this implements the recursive search up the tree.

notifyattr(resourceIdentifier, value)
called from parent (for editres protocol,
however it is supposed to work)

attachchild(dobj)
called from parent
connects dobj as a child of 'self'.
an object cannot be the child of more than one parent. (other dataobjects can refer to dobj, but only one can be the parent.)

new class: astore
will store two sets of attributes:
for local and for children
methods: get, put, more or less as above
every data object will have an attribute store

c
reating and inserting an object

creating a new widget means to create a resource file and install it in the widget specific resource file directory, $ANDREWDIR/widgets/

in addition to trying to instantiate an object, esc-tab will look on the XUSERSEARCHPATH for a directory containing a file that has the name of the requested object class. if found, a widget driver is instantiated and given that file for its default resources, and may be overriden by other entries on the path.

Software Developments

Rob Ryan, Technical Staff

Widgets Work Progress on the underpinnings for a new widget set continues. A standard mechanism for argument passing to callback functions and calling between Ness and C++ has been completed. The basic workings of an option menu have been

completed as well. This may one day allow the entire menu system to be implemented under the Andrew Toolkit.

New Developer Tool: Function Locator New to Andrew7.2 is FunctionLocator, a file which provides a list of external functions, along with their source location and the library in which they reside. All information about a particular function is on the same line allowing users to "grep" the function name.

To enable this feature, add the following to your site.h file:

```
setenv FUNCTION_LOC_ENV 1
```

When you set this environment variable, the file $DESTDIR/doc/atk/FunctionLocator will be created and installed at the end of a build. Depending on the system type, the file size ranges from 1.0-2.5 MB.

Patch 6.3.4 A patch is now available to AUIS version 6.3.3 that fixes a potential cause of coredumps for anyone who uses eatmail or has AMS/Messages configured to read mail from a UNIX-style mail spool. The bug is triggered when the "From" line added by sendmail has a local address with a space in it.

The patch fixes three files: ams/libs/ms/cvtold.c, contrib/eatmail/eatmail.c, and atk/atkvers/atkvers.num.
The file is called "patch.634" and is located in the /pub/AUIS/patches/official/ directory on our ftp server, ftp.andrew.cmu.edu. The README in that directory provides further instructions.

After applying the patch, you need only to recompile the following directories under auis-6.3:

atk/atkvers	atkams
ams	contrib/eatmail

Contributed Port: SGI An SGI port to Andrew6.3.3 has been contributed by Bill Cattey <wdc@mit.edu>, a long-time Andrew user at MIT. (Thanks Bill!)

Three ASCII files are all you need to run Andrew on the SGI5.2 system. They are available on our server via anonymous ftp at ftp.andrew.cmu.edu in pub/AUIS/patches/contrib:

SGI-port.6.3.3.README
- additions for your local $ANDREWSRCDIR/config/site.mcr and $ANDREWSRCDIR/config/site.h files

SGI-port.6.3.3.diffs
- context diffs of existing files

SGI-port.6.3.3.newfiles
- additional files to be added to the AUIS version 6.3.3 sources

As the location of the patch implies, this is a contributed feature which is not supported by the Consortium staff.

Contributed Patch: malloc Again, Bill gets credit for this latest contributed patch to "malloc", a version of the memory allocation program written by our own Fred Hansen over ten years ago. The computation of the arena size in addarena is now correct.

See </pub/AUIS/patches/contrib/malloc.ci.auis6.3.diff via anonymous ftp at ftp.andrew.cmu.edu>.

This patch is to be applied to the file $ANDREWSRCDIR/overhead/malloc/malloc.ci in AUIS versions up to and including 6.3. Instructions for applying patches are located on our ftp server in the /patches/official/README file.

Member Mentions

Andrew at MIT The Massachusetts Institute of Technology (MIT) is a long-standing member of the Andrew Consortium.

MIT has been using the Andrew User Interface System (AUIS) since the early 1990s, particularly for word processing on UNIX boxes and file exchange. On a typical day at MIT, over 7000 users access their personal files and various software packages over the Athena Computing Environment, the University's computer network. Over 1000 machines from several manufacturers on that network run Andrew. It is unclear exactly how many of the 1000+ machines use Andrew at any one time, but it surely is a lot!

MIT's Distributed Computing and Networking Services (DCNS) development group in the Information Systems department is responsible for software support on campus. Bill Cattey is the self-described "chief cook and bottle washer for the Andrew User Interface System at MIT". Bill takes care of the EZ document preparation environment and various other useful tools based on Andrew that are provided on Athena. He is also the Consortium's number one proponent of Andrew in Boston...and we quote, "AUIS has pioneered many fine issues in User Interface Toolkit and multi-media application development. I had been writing a multi-media editor myself, but I abandoned it for EZ when I saw the beauty of the AUIS architecture." Bill and the DCNS also provide Andrew Toolkit (ATK) applications to their users.

The University is about to release Andrew6.3.3 to Athena. This latest release will include html support for users who want to more easily create documents for the Web.

Just some of the features that MIT has contributed to the Andrew User Interface System over the years include:

NEOS	applications for classroom-oriented file exchange of EZ docs
insets	header/footer, Postscript and troff
translators	rtf2, 2rtf, and scribetext
support	tabs, and dialog boxes, gnu emacs compatibility enhancements

Just this month, MIT has contributed the following:

* patch for malloc.ci
* port to SGI5.2 system
* patch for Solaris

The future of AUIS at MIT is cloudy. The Athena Computing environment now offers services on PC and Macintosh platforms in student rooms and in offices, in addition to the UNIX workstations. The market for UNIX-only applications is being eaten away by multi-platform services or particular off-the-shelf applications like Word for document preparation, and Lotus Notes for groupware.

The 1000 workstations of the past seemed like a lot, but the network drop per student PC makes that number seem tiny indeed.

Just as CMU is migrating to more and more product bought from companies, and less and less on maintaining in-house developments, so too support for Andrew at MIT may fade. Bill pins his hopes on the Linux community, and on the dream of radical ports of AUIS to the Mac or PC.

News from the Consortium

Susan Straub, System Administrator and AnnMarie Zanger, Assistant Director

The 1995 Survey of Andrew Users has been distributed to as many Andrew users as we could find. We hope that you will take a few minutes out of your busy schedules to fill out the form. It will help us to help you get the most out of Andrew!

You will find the survey on the web at: http://www.cs.cmu.edu/ AUIS/survey

Need a hard copy of the survey? Have questions about it? Contact the Andrew Consortium at (412) 268-6710 or email susan@andrew.cmu.edu.

Conference and Annual Meeting The *1995 Andrew Technical Conference* will be held in Pittsburgh on a Thursday and Friday during the third week of September.

The annual event will showcase the Andrew User Interface System and the WorldWideWeb, both as exclusive entities and as mutually-evolving vehicles, which improve users' ability to process information.

We look forward to a lively two-day mix of technical presentations and invited talks, and as always, we have scheduled the Andrew Consortium Annual Meeting right in the middle. Keep working on those papers and look out for your registration form.

Date: September 21-22, 1995

Place: Carnegie Mellon University
Pittsburgh, Pennsylvania

Sessions: "Andrew as a Tool": September 21, 1995
Annual Meeting - Andrew Consortium: Evening of September 21, 1995
"Using Andrew and the Web": September 22, 1995

Employment Opportunities The Andrew Consortium has two programming positions open. As you know, the Consortium staff enhances and maintains the Andrew User Interface System (Andrew), one of the leading compound document environments. Andrew provides various applications based on these compound documents--word processor, mail/bboard system, help browser, drawing and font editor, directory browser--as well as a variety of embeddable objects--drawings, images, spread sheets, equations, etc.

We welcome your applications as:

* Bug Czar (full-time staff) - Your responsibilities will include fixing bugs, managing bugs database, documenting system changes, testing the system, problem solving, and system development.

* Developer (full-time staff) - You will design and implement new features in existing software, ensure the installability of the system, and communicate with Consortium members.

Please contact our director, Fred Hansen, at 412-268-6788 or via email <wjh@andrew.cmu.edu> if you are interested in any of these positions or if you would like to recommend someone to us.

FACTS ON FILE

Andrew is...

Compound Document Architecture
Word Processor
Drawing Editor
Mail /Bulletin Board Reader, Writer, Manager
Spreadsheet / Table Editor
Widget Set
Application Builder
Scriptable
In C and C++
For X11

...With a consistent, integrated user interface so you can create compound documents containing pictures, words, tables, graphs

...and more.

The ***Andrew User Interface System*** is an integrated set of tools that allow you to create, use, and mail documents and applications containing typographically formatted text and embedded objects. ***AUIS*** or ***"Andrew"*** has three principal components:

The ***Andrew User Environment (AUE)*** is an integrated set of applications beginning with a 'generic object' editor (*ez*), a help system, a system monitoring tool (*console*), an editor-based shell interface (*typescript*), and support for printing multi-media documents.

The ***Andrew Toolkit (ATK)*** is a portable user-interface toolkit. It provides a dynamically-loadable, object-oriented environment wherein objects can be embedded in one-another. Thus, one could edit text that contains not only fonts and styles, but also embedded raster images, spreadsheets, drawing editors, equations, simple animations, etc. These embedded objects could themselves contain other objects, including text. ATK is an open system so programmers can create new objects that can be embedded as easily as those system-defined objects.

The ***Andrew Message System (AMS)*** provides a multi-media interface to mail and bulletin-boards. AMS supports several mail management strategies and implements 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 ttys and low-function personal computers in addition to the high-function workstations.

Andrew-Ready Platforms

Andrew has been used successfully on (at least) these platforms:

IBM---RT AOS 3.4, RT AIX 2.2.1, RS/6000 AIX3.1, PS/2 AIX1.2
SUN---Sun3 3.5, Sun3 4.0, Sun4 4.0, Sun3 4.1, Sun4 4.1, Sun4Mach, Solaris
DEC---Vax Ultrix 3.1, Vax Ultrix 4.2, Vax BSD, DEC MIPS, Pmax Mach
Other---HP, SCO I386, SGI IRIX 4.0, SGI IRIX 5.2, Apollo, Macintosh II, MacMach, 486 Mach, Telmat, Linux

Remote Demo Service

Try out Andrew from your own work station via the Remote Andrew Demo Service. You will need a host machine on the Internet running the X11 window system. You will be able to compose multimedia documents, navigate through the interactive Andrew Tour, and use the Andrew Message System to browse through a few of CMU's four thousand bulletin boards and newsgroups.

To get started, simply run the following command on your machine: finger @atk.cs.cmu.edu (128.2.203.218). The remote demo system will give you further instructions.

Accessing the Source Code and Binaries

Sources and binaries are available online via anonymous ftp from: ftp.andrew.cmu.edu (IP: 128.2.232.154) in pub/AUIS/. See the README file.

Other ftp sites are: ftp.x.org (IP: 192.112.44.100) and its clone sites (see pub/R6untarred/contrib/lib/auis-6.3). On the nationwide AFS file system, Andrew is available in: /afs/cs.cmu.edu/project/atk-ftp.

Binary distribution packages for subsets or all of Andrew are available for the following system types:

RS/6000
Sun 4C Sparcstation
HP 720
DEC Pmax
Linux

Source and binary tapes are also available by mail (see "Services and Offerings-Andrew on Tape" below).

Membership in the Andrew Consortium

The School of Computer Science at Carnegie Mellon University has established the Andrew Consortium to maintain and enhance the Andrew User Interface System (Andrew).

The Consortium offers outside organizations the opportunity to help shape the future of the system. Participate in the development of Andrew, utilize the latest advances by our technical staff, and undertake commercial exploitation with the active cooperation of the developers.

The overall efforts of the Consortium staff are directed toward:

+ increased quality in existing media,
+ enhancements, including a direct manipulation interface builder,
+ embedding of applications such as generic X applications,
+ increased potential for transition from Andrew to successor systems
+ inter-operability with standards such as Posix, Motif, C++, OLE, and SGML.

The Consortium makes its most up-to-date release, currently Andrew 7 in C++, available only to members. The current publicly available Andrew release is 6.3, released via anonymous file transfer over the Internet.

Membership in the Consortium is offered to organizations interested in exploiting Andrew technology within their operations or products. All members support the consortium as it:

+ Enhances, maintains, and distributes the Andrew software base.
+ Conducts an annual meeting to review progress and set priorities.
+ Distributes a quarterly newsletter
+ Monitors the info-andrew mailing list.

We offer memberships at four levels: Participating, Full, Contributing and Associate. You are cordially invited to join us.

Please contact our Director, Fred Hansen <wjh@andrew.cmu.edu> if you would like to discuss the opportunities that the Consortium has to offer.

Consortium Services and Offerings

The following materials are provided by the Andrew Consortium at Carnegie Mellon University for internal educational use only and may not be reproduced for sale or distribution. Please mark your selections below:

ANDREW IN PRINT

_____ **The Andrew User Guide** US$25

This printed and bound guide, offers a comprehensive description of how to use Andrew applications and insets. Chapters cover ez, help, typescript, messages, console, raster, figure, table, and many other facets of the system. (130 pages)

_____ **Selected Technical Papers on Andrew** US$30

This set represents a selection of some of our most handy references on the Andrew system. (250 pages)

A Guide to Andrew (User Tutorial)
Ness: An Extension Language for the Andrew Toolkit
Programming with the Andrew Toolkit
ADEW: The Andrew Development Environment Workbench
Createinset: Generate Source Files for a New Inset
The Andrew Message System

_____ **Andrew Technical Conference Proceedings** US$20

The Andrew Consortium sponsors its own technical conference each year. This volume includes the entire proceedings of the 1994 event, (Application Developer's View of Andrew, The Web Browser, Compound Document Architectures) plus highlights from the 1992 and 1993 conferences. (85 pages)

_____ **Programming Documentation on Andrew** US$95

This package represents a complete set of programming documentation on the Andrew system, sold in hard copy form. (1250 pages)

ANDREW ON TAPE

_____ **Public Source Tape (6.3 in C)** US$150

Want the source code on tape? Here it is, ready to use on your Andrew-compatible platform.

_____ **Private Source Tape (7.2 in C++)**

_____ US$350 for business use _____ Name of designated user/company

_____ US$250 for personal use _____ Name of designated user

Purchase our most up-to-date version of the source in C++ and take advantage of the latest programming advances made by our technical staff. (This tape must be designated for the use of one individual only.)

_____ **Binary Distribution Package (6.3 in C)** US$100

This tape provides you with everything you need to get Andrew up and running on your RS/6000, Sun 4C Sparcstation, HP 720, DEC Pmax or Linux machine. Files include the source code, word processing, documentation, and mail for MIME-compatible messages.

If you are ordering ANDREW ON TAPE, please specify format:

_____ 1/4" streaming tape (150 MB)
_____ 8 mm DAT tape (2.3 GB)
_____ 8 mm HP Iotamat format

US$ _____SUBTOTAL

_____ Add US$20 per item for US express mail delivery
_____ Add US$25 per item for air mail delivery outside the US
_____ Add US$40 per item for express mail delivery outside the US

US$ _____GRAND TOTAL

Name _____
Company _____
Address _____
Phone _____ Fax _____ Email _____

Please place your order by mailing the previous section of this form with your selections marked. Include a check made payable to Carnegie Mellon University and send it to the address overleaf.

Contact Information

Andrew Consortium
School of Computer Science
Carnegie Mellon University
5000 Forbes Avenue
Pittsburgh, PA 15213-3890

Phone: (412) 268-6710
Fax: (412) 268-5571

Email:	subscriptions, info, orders --- info-andrew-request@andrew.cmu.edu bug reports --- info-andrew-bugs@andrew.cmu.edu discussion forum --- info-andrew@andrew.cmu.edu
Newsgroup:	internet.other.info-andrew@andrew.cmu.edu (Andrew format) netnews.comp.soft-sys.andrew@andrew.cmu.edu (ASCII text)
Demo:	finger @atk.cs.cmu.edu
Access:	ftp.andrew.cmu.edu: pub/AUIS/
Web:	http://www.cs.cmu.edu/ AUIS (homepage) http://www.cs.cmu.edu/ AUIS/survey.html (survey) http://www.cs.cmu.edu/ AUIS/cfp.html (conference)

WWW ALERT

Now it's even easier to find us on the web! The abbreviated URL listed above is new this summer. Our old address works, too, and each file is cross-linked to the other so it's pretty hard to miss anything. Look us up for more information about Andrew and the Andrew Consortium.