

# THE **e**LEARNING DEVELOPERS'

Strategies and Techniques for Designers,  
Developers, and Managers of eLearning

**JOURNAL**™

## THIS WEEK — DEVELOPMENT STRATEGIES

### An Overview of Authorware 7 — and Its Evolution

BY JOSEPH GANCI

**W**hen I first started using Authorware in May, 1990, it was almost a brand new product, available only for the Apple Macintosh, sporting a Version 1.0 sticker and delivered on several floppy disks with a few manuals in a faux leather briefcase. I had been programming computers for quite a while by then, having started at the age of 14 in 1974. I had worked on the then ancient TICCIT training system for several years; if you remember that system, you're no spring

chicken! I had used TenCore for a year and I had been a hardcore C and Pascal hound. So you might believe I was well prepared to be an Authorware developer.

Not so. Having been a "scripter" did not necessarily prepare me to be an Authorware developer. In fact, when Authorware was created, it was with the intention that *anyone* should be able to create computer-based training (which we now more often call e-Learning, of course). The whole point to developing in Authorware was to use an icon and flow-line approach, with dialog boxes that would help the developer quickly and easily create an event-driven application

with interactivity and some data tracking. It was the ultimate prototyper's dream; but it would also take you beyond the prototype stage to a finished product.

This summer Macromedia released Version 7 of Authorware — and a very good version it is. Before we talk about this new release, though, more history is in order. Only by seeing what has led up to this version can one get a proper appreciation of Authorware 7's true offerings.

#### Ancient history

Back in 1963, the National Science Foundation provided a grant that ulti-

*Continued on next page*

*It seems as if Authorware has been around almost as long as the e-Learning field itself, and for nearly that long, industry gossip has been that the then-current version of Authorware would be the last. But after thirteen years, Authorware is still going strong. Read this article to learn why you shouldn't bet the farm on Version 7.0 being the end of the line!*

THE **e**LEARNING DEVELOPERS' JOURNAL

**Publisher** David Holcombe

**Editorial Director** Heidi Fisk

**Editor** Bill Brandon

**Copy Editor** Charles Holcombe

**Design Director** Nancy Marland

**The eLearning Guild™ Advisory Board**

Ruth Clark, Conrad Gottfredson, John Hartnett, Bill Horton, Kevin Moore, Eric Parks, Brenda Pfaus, Marc Rosenberg, Allison Rossett

Copyright 2003. **The eLearning Developers' Journal™**. Compilation copyright by The eLearning Guild 2003. All rights reserved. Please contact **The eLearning Guild** for reprint permission.

**The eLearning Developers' Journal** is published weekly by **The eLearning Guild**, 525 College Avenue, Suite 215, Santa Rosa, CA 95404. Phone: 707.566.8990. **The eLearning Guild** is an operating unit of Focuszone Media, Inc., 1030 Beatrice Street, Eagan, MN 55121.

**The Journal** is included as part of **Guild** membership. To join the **Guild** go to [www.eLearningGuild.com](http://www.eLearningGuild.com).

**The eLearning Developers' Journal™** is designed to serve the industry as a catalyst for innovation and as a vehicle for the dissemination of new and practical strategies and techniques for e-Learning designers, developers and managers. The **Journal** is not intended to be the definitive authority. Rather, it is intended to be a medium through which e-Learning practitioners can share their knowledge, expertise and experience with others for the general betterment of the industry.

As in any profession, there are many different perspectives about the best strategies, techniques and tools one can employ to accomplish a specific objective. This **Journal** will share these different perspectives and does not position any one as "the right way," but rather we position each article as "one of the right ways" for accomplishing a goal. We assume that readers will evaluate the merits of each article and use the ideas they contain in a manner appropriate for their specific situation. We encourage discussion and debate about articles and provide an Online Discussion board for each article.

The articles contained in the **Journal** are all written by people who are actively engaged in this profession at one level or another — not by paid journalists or writers. Submissions are always welcome at any time, as are suggestions for articles and future topics. To learn more about how to submit articles and/or ideas, please refer to the directions in the box on page 9 or visit [www.eLearningGuild.com](http://www.eLearningGuild.com).

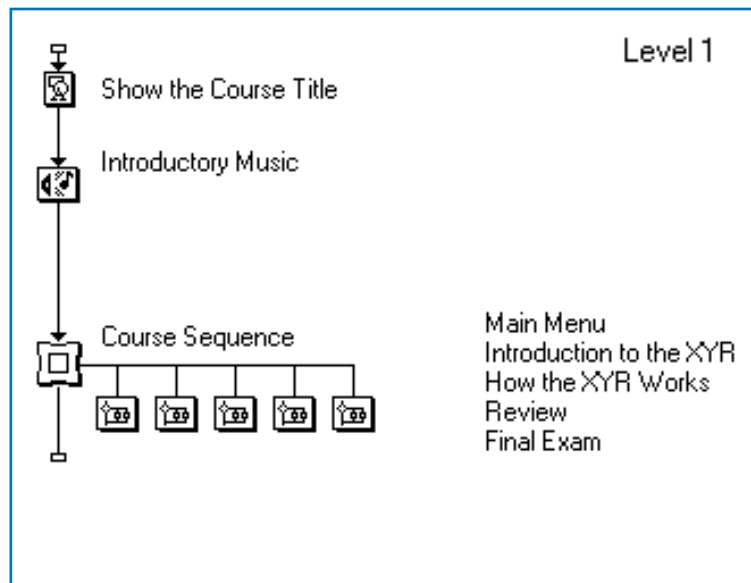
mately spawned two main authoring systems for the creation of computer-based training. One was the aforementioned *TICCIT*, which stood for *Time-Shared Interactive Computer-Controlled Information Television* (whew!). The other, which became much more popular, was called *Plato* (*Programmed Logic for Automatic Teaching Operations*). Both of these products spawned many others, but *Plato* proved to be the more successful in its progeny.

One of those *Plato* descendants was *Authorware*, although the line of descent was a little complicated. The University of Illinois and Control Data Corporation (CDC) developed *Plato*. CDC was fortunate enough to have Dr. Michael Allen as one of its employees. At the time, mainframe and minicomputer-based e-Learning was starting to make way for microcomputers and Dr. Allen was working on a PC version of *Plato*, codenamed *PCD03*.

However, *Plato* was very much script-based and Dr. Allen had a vision for an alternative approach that was more visu-

al. He headed up a team of three people at CDC to create a visual *Plato* product, which he called *Course of Action*. As the story goes, CDC did not express much interest in the product once done, so Michael made an offer for it, and product in hand, left CDC to start a company called *Authorware Inc.*, headquartered in Minneapolis. The product changed its name once slightly to *Best Course of Action*, and then to *Authorware*. In fact, for several versions of *Authorware*, if you were to look at *Authorware's* file type in the Macintosh's resource editor, you would see it was *COA* (for *Course of Action*). Now you know what that meant (and all these years you've been wondering, haven't you?).

The visual approach Dr. Allen had anticipated found a home in *Authorware's* icons and flowline. We'll discuss each of the icons in just a moment. The flowline method allows a developer to drag icons one at a time from an icon palette onto a flowline. (See Figure 1, below.) The developer can place icons above or below other icons, and in some



**FIGURE 1** (above) In *Authorware*, programming is done visually, using icons and a flowline.



**FIGURE 2** (right) *The Authorware icon palette, Version 1.*

instances to the left or right of other icons. This forms relationships between the icons, in much the same way that you would create a decision tree flowchart, so commonly used in business circles.

After you drag an icon to the flowline, you title it with its purpose. For instance, if the icon will play a song, you might give the icon the song's title. This helps to document your work as well. By looking at the flowline, you can get an overview of what the file does and the order in which it does it.

The Windows version had been projected for publication in 1990, but it finally arrived in 1992. It had been a much harder undertaking than first expected.

The price for Version 1 was a cool eight thousand dollars. By that standard, today's price, less than three thousand dollars, is positively a steal. The Version 1 developer toolkit consisted of the icon palette you see in Figure 2 on page 2. In order from top to bottom, the icons included were the *Display* icon, used for showing text and images on the screen, the *Motion* icon, used for moving objects on the screen, and the *Erase* icon, with which you could remove objects at the appropriate time. The *Wait* icon could be used for waiting a certain number of seconds or waiting for the user to click the mouse or press a key.

The *Decision* icon, shaped like a diamond, was very important in that it allowed paging mechanisms to be set up, albeit in a bit of a roundabout way, as Dr. Allen was never a big fan of page turners (good for him!) and I believe he was trying to avoid having developers fall back on these types of lessons too easily.

The *Interaction* icon, shaped on the left like a Display icon and on the right like a Decision icon to form an icon that looked like an arrow with a question mark inside, was and still is arguably the most important. With it, you could set up any of the response types you see in the dialog box in Figure 3, right.

The little equal sign icon was called the *Calculation* icon and with it you could write small scripts, using Authorware's own scripting language (which has remained unnamed to this day). However, the scripting language was very limited for someone used to real programming languages in which you could write your own functions, set up subroutines, and

the like. Again, the point was driven home: this was not a development package for scripters as much as something new: a way for nonscripters to develop powerful e-Learning.

Despite this, no development package can be complete without giving the developer some ability to script. To have every option possible for every situation as a field in a dialog box would result in a plethora of large dialog boxes, unwieldy at best. Authorware's first scripting language may have been limited, but it was necessary to allow for those options not available in dialog boxes.

The *Map* icon was used to group two or more icons into one icon. This was not only a nicety but also a necessity. The flowline could only be as long as the screen resolution allowed in height, so you could quickly run out of room as you strung these icons together on the flowline. This proved an excellent way to force people to organize their code. For instance, once the code for the Introduction to the course was completed, it could be grouped into one Map icon and be titled *Introduction*, so that in essence it became compartmentalized.

The two flag icons, *Start* and *Stop*, could be used to section off parts of the code so that when running the application during testing, only the parts between the start and stop flags would execute.

The last three icons were called the media icons and they were, in order, the *Movie*, *Sound*, and *Video* icons. The *Movie* icon could be used to run a number of types of animation and movie files (eventually it would come to support QuickTime, AVI, and others). The *Sound* icon could import external sounds and play them at various speeds and the *Video* icon was used to connect and play external video devices, primarily video-disc players.

## Time marches on, new versions appear

In 1992, Authorware Inc. merged with Macromind Paracomp to form Macromedia, Inc. and Bud Colligan, at the time CEO of Authorware Inc. (having arrived there from Apple), was chosen to head up the new company. The headquarters moved to San Francisco, where Macromind Paracomp was already well established.

## Version 2

A year later, the next major release of Authorware was brought to market. It added some nifty new features. The most important of these was *libraries*, which for the first time allowed developers to place icons in separate library files which could then be shared between Authorware files and developers. This meant that a company logo, for instance, could be placed in a library file and then linked to hundreds of Authorware lesson files. If the library file icon was updated to contain a new logo, that change could be permutated to all the lesson files. This was a huge timesaver! Sounds, images, and other media could now be more easily updated.

In addition, for the first time text could be scrolled on the screen. While this is not necessarily a boon for design, it could be useful when long reference materials that were not necessarily crucial to lesson plans needed to be made available.

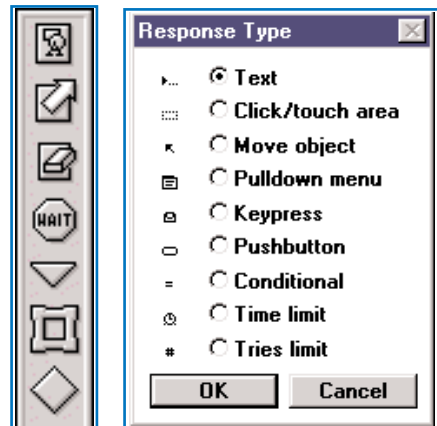


FIGURE 3 (above) The Authorware Response Types, Version 1.



FIGURE 4 (left) The new Authorware icon palette in Version 3.

Among other new features, the hot object response should at least be mentioned here. This was another cool element, in that this meant that authors no longer had to keep screen elements that users needed to click from moving on the screen. This means you could now enhance an exercise in which you show the parts of a machine in pieces. Since Version 1, you could allow students to click each part and have that part animate to its proper position in the machine. However, starting with this version, that part, though moved to a new location on the screen, could still be clicked. At that point, you might choose to have the part move out of the machine back to its original location or perform any other event you wish.

### Version 3

Before Version 3 was released in 1995, I attended the Macromedia conference. During the Authorware Sneak Peeks session, the engineers showed the new icon palette. On it were two new icons that had all us all abuzz. (See Figure 4 on page 3.) When someone asked the inevitable question, "What are those two new icons?" the answer was, "Yes, interesting, aren't they?"

Eventually, we came to learn that the new icons were called the *Framework* icon (which looks like a frame on the icon palette — the sixth icon from the top of the palette) and the *Navigate* icon (which is right above the Framework and looks like a triangle) and they were used not only to easily create paging mechanisms, but also many other options,

such as automated searches for students, roundtrip navigation which allowed a form of subroutine previously not available, and more. This didn't mean that we were now all forced to create page-turners (though, alas, this did become easier) but it freed us up to create almost any design approach in a faster way.

The new six-color palette at the bottom of the icon palette could now be used to color icons, which allowed developers to set up team development a little more easily ("If I color icons red, don't touch them!"). Other features were added, including a spell checker, the ability to set up text styles, a button editor to create custom buttons, a trace window to assist in debugging applications, Rich Text Format importing, and more.

Oh, and Version 3 was now being delivered on CD-ROM.

By now, the World Wide Web was fully coming into its own. No longer considered a cute fad, investors and others were jumping onto the WWW bandwagon. Even companies like Microsoft and America Online were seeing the Web as an inevitable, ever-growing phenomenon. This fact was not lost on Macromedia, which not too long thereafter started positioning itself more and more as an Internet company.

### Version 3.5

A year later, in 1996, an important dot version was released. The most important new feature in this release was the ability for Authorware files to be displayed in Web browsers using a plug-in,

as Macromedia Director (another Macromedia product) was already doing at the time. Authorware also started using *Xtras*, which were external files that could be used to extend Authorware's abilities more easily. With them, third party companies could create new transitions and small applications that could be used in Authorware, Director, or both.

### Version 4

The first three versions of Authorware had sometimes been criticized by competitors for not allowing content to be linked externally, meaning that image files, for instance, could not be kept in an external folder and be used at run time. In 1997, the next major version of Authorware was released. It was obvious that the engineers had been hard at work. Version 4 brought an end to the linking limitation. Content could now be imported or kept external on an individual basis. There were, and still are to this day, advantages and disadvantages to both these methods, but now authors had a choice for almost every media file they were given.

The scripting language also got a major boost with the advent of *linear lists* and *property lists*, which already had been a part of Lingo, Director's scripting language. The same benefits of lists could now be had by Authorware developers, principally the ability to organize large amounts of data inside of the development package itself.

Using linear and property lists meant that developers who read in data from databases or text files had an excellent place to store that data in Authorware. A list could hold up to 30,000 locations, and each location could hold up to 30,000 characters, which meant that one list could hold almost a gigabyte of data, as long as computer memory permitted.

For instance, a list of student names or customer records could be read and sorted, along with each student's past history and contact information.

It didn't stop there, though. As each list could have up to 10 dimensions, meaning that a location in a list could be another list, and each location in that list could be another list, and so on, up to 10 permutations, a multidimensional list could in reality hold more data than you

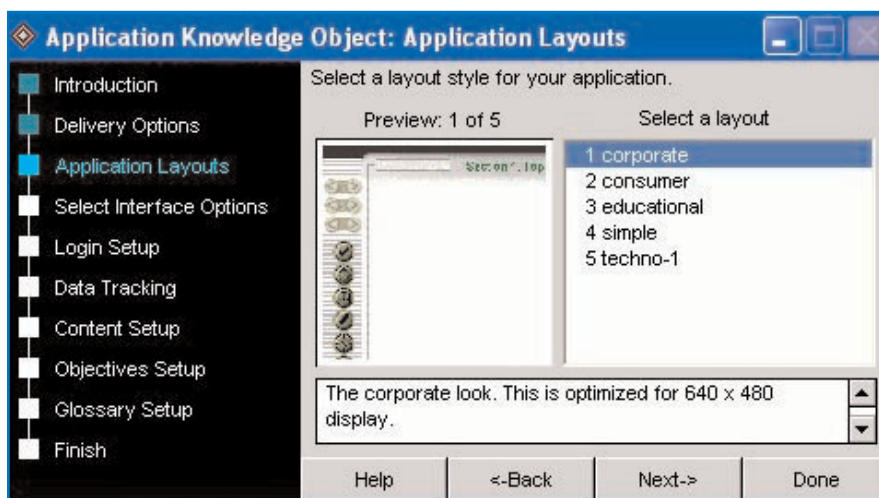


FIGURE 5 The wizard interface allows novice authors to set up a full application.



could probably ever throw at it. In fact, to this day, most advanced authors haven't used more than four dimensions (30,000 x 30,000 x 30,000 x 30,000 characters).

This added to Authorware's ability to work with live data, such as today's student or patient data, at run time, making it much more dynamic. You no longer had to use "canned" data.

Other new features included the ability to link to Flash files (it was already possible to link to Director files) and to interact with ActiveX controls.

### Version 5

Promptly on schedule, Version 5 was released in 1998. With this release, Authorware no longer provided a separate Macintosh version for developers. Up until Version 5, you could develop lessons on either the Macintosh or in Windows and deliver to students on both platforms. Now, you could develop lessons only in Windows, though you could still deliver them to students on both platforms. This continues to be true today. This allowed for much better integration between Authorware and Windows, once Windows had, for all

---

**About 80% of PowerPoint features can currently be converted to Authorware icons automatically. Even if a PowerPoint file uses features that are not possible to convert automatically, the converter will still bring Authorware developers much closer to the final conversion, saving many hours of time.**

---

intents and purposes, won the operating system war.

A major part of this release was the inclusion of *Knowledge Objects*. These wizard interfaces would allow new developers to create new applications and quizzes more easily. Version 5 also allowed experienced developers to develop new Knowledge Objects for novice Authorware users. Organizations that were creating large projects with many lesson files could use high-end developers to create Knowledge Objects to be used by lower paid novice and intermediate level developers.

For example, the wizard interface shown in Figure 5, on page 4, allowed novices to set up much of the work for a full application. Once done, the wizard would then automatically create the structure and place it on the flowline for the developer.

Anti-aliased text was also introduced in this version, along with the use of the alpha channel in those image files that would support it. Another timesaving feature, batch changes, was added as well.

Scripting got a further boost because the Authorware engineers started to open up more icon properties to

San Francisco, California • November 12 – 14, 2003 Pre-Conference Workshops November 11, 2003

Register Today! Early Registration Discount ends September 26.

# eLearning Producer™

## CONFERENCE & EXPO 2003

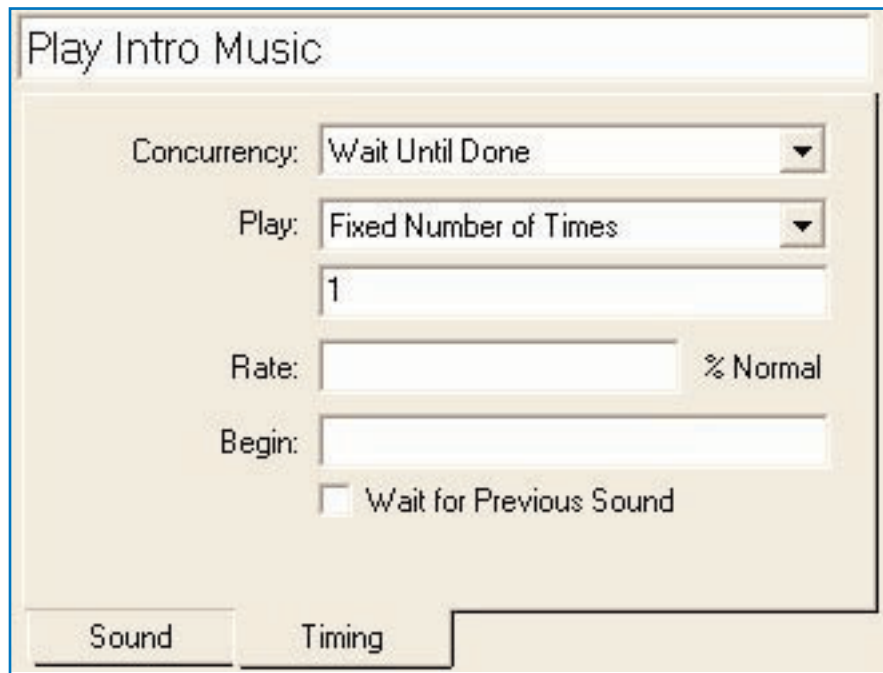
*The Essential Conference for e-Learning  
Designers, Developers, and Managers*



The program is divided into three core practice areas that mirror the three main roles that e-Learning producers are engaged in – Designer | Developer | Manager. The Program Committee, comprised of volunteer Guild Members, has identified the most critical topics that address these areas and those topics will be the focus of all our General Sessions, Think Tank Sessions, Concurrent Sessions and optional Pre-Conference Workshops.

Every session is designed to allow time to hear from the experts, debate and discuss with your peers and explore topics to their fullest. In addition to the structured time you will spend learning in the sessions, you will also have ample opportunity to network with the other participants in morning discussion groups, over lunch, and during social activities. The size and format of the conference will help you develop important professional connections during the conference that you can continue in the months and years to follow.

**Concurrent Sessions, Workshops,  
Think Tank Sessions online now!**  
Visit [www.eLearningGuild.com](http://www.eLearningGuild.com)



**FIGURE 6** Dialog boxes allow setting options in order to change icon properties.

scripters. Recall that almost all of the icons had associated dialog boxes where developers could set options, such as in the dialog box shown in Figure 6, above, for the Sound icon.

Each of these options, and some that cannot be seen in any dialog box, are examples of properties. In previous versions, an option set in any dialog box (such as the *Concurrency* setting above) could not be changed on the fly from within the Authorware file at run time. Changes that began in this version meant that developers could do that very thing with a simple script. No longer were they constrained by the choice they made while setting up the file. If the sound should *Wait Until Done* in one case but the same sound play *Concurrently* in another case, the same sound icon could now be used in both cases just by making a change in a script.

The Calculation icon itself got its first major upgrade in interim Version 5.2, released a year later. It now sported its own tool bar, shown in Figure 7 below, with several new features, such as multiple undo levels.

## Version 6

It would be 2001 before Version 6 was made available to developers. A lot of work led up to this release.

Until this version, Authorware developers would *package* files to make them ready for students. This created executable files that users could run just like any other application on their systems. Packaging was never a fun part of the development process, especially after Xtras had been introduced in Version 3.5, because the proper Xtras had to be included with the packaged files, and you were left to your own devices to remember to include all the proper Xtras. Since Version 4 the same held true for external content files.

Something had to be done to simplify the process and so the One Button Publisher feature was born. By using the OBP, developers were freed from much of the fuss and bother of publishing their student files for both local and Web delivery.

*Commands* were another new feature of AW 6. Acting somewhat like Knowledge Objects, but in a freer form,

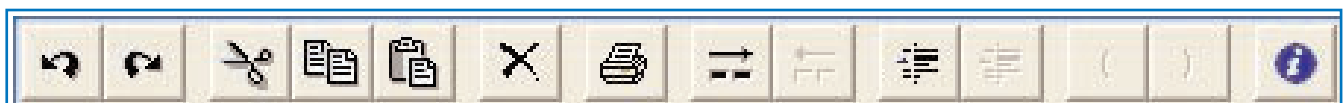
Commands could be created by developers both in Authorware and in Borland Delphi to extend Authorware's abilities. Commands could be shared with other developers, further easing the team oriented approach to developing large-scale projects. In fact, some key features of this release were written as Commands, including the One Button Publisher.

Here's an example of how powerful the Commands feature proved to be. To practice my programming skills in Borland Delphi and because I was on the Authorware beta team, I decided to write a Command in Delphi for the AW 6 release that would dive through an Authorware file and make a list of useful information about every media asset found — images, video clips, and audio clips. I never thought it would prove all that useful as a Command, but shortly after the release of Version 6, a client of ours asked us to upgrade a large set of lessons that had been developed in Authorware, and one of the first things we had to do was to compile a list of media assets! Using this Command, work that might have taken many days was accomplished in less than a minute. That's the power of a well-written Command!

Another Command that was included in Version 6 was an RTF Objects Editor, which now made it easy for developers to create and use RTF objects. Rich Text Format objects give designers and developers a good deal of power, especially to update text content and attributes (font size and shape, for instance) after lessons have been delivered.

Improvements were made to Authorware's interface. Performance tracking, always a hallmark of Authorware, was extended in this version. This allowed developers to integrate their Authorware files with Learning Management Systems with a degree of ease not possible previously.

More properties and further extensions to Authorware's scripting language were also introduced. While Authorware continued to remain relatively easy to use for nonscribers, hardcore program-



**FIGURE 7** The new tool bar in Version 5.

mers felt more and more at home in it, because scripts could now be written that allowed unprecedented control over how Authorware files would behave at run time.

### Version 6.5

In 2002, Version 6.5 was released. Far from being just an interim dot release, this proved a major boon for those of us who had long missed the ability to create our own true script functions in Authorware. Script functions were now allowed in Authorware, which allowed developers to cut down further on the number of icons that were normally needed in previous versions. Ask any programmer about the ability to write functions and he or she will say that the job would not be possible without this capability. While Authorware has always allowed workarounds to the lack of this feature, giving developers this ability suddenly freed them to an unprecedented degree.

The other big new feature of this ver-

sion was the Accessibility Kit, a new Command that allowed developers who were creating new lesson files to make them easier to use for students with disabilities. Using the Kit, developers could use a text-to-speech feature to read screen text out loud for those who are vision impaired, set up overall keyboard shortcuts, and more. It also included a tutorial on how to make lessons more accessible from every angle.

Even more properties were opened up in this release and the One Button Publisher was further improved.

### At last: Version 7 arrives

Now Version 7 has come to market. Several changes and additions are included with this release that makes it a must-have for Authorware developers.

### The user interface

The user interface got its first major overhaul, making it closer to the look and feel of Macromedia's other products. Those using Flash, DreamWeaver,

and other Macromedia products will be able to come up to speed more quickly in Authorware now.

The new Property Inspector makes it easier to change icon options. In past versions, once you opened an icon dialog box to make changes to the icon's options, it would have to be closed before you could continue to other parts of your file. However, now the options appear in the Property Inspector at the bottom of the screen. This leaves you free to move about to other parts of your file. Every time you click an icon, the Property Inspector will show the options for that icon.

The toolbar is now movable, as opposed to past versions where it was always fixed to the left side of the screen. (See Figure 8 below) You can even move it outside of the Authorware application window — it is completely independent.

On the right side of the screen is a special docking section where you can keep several features organized togeth-

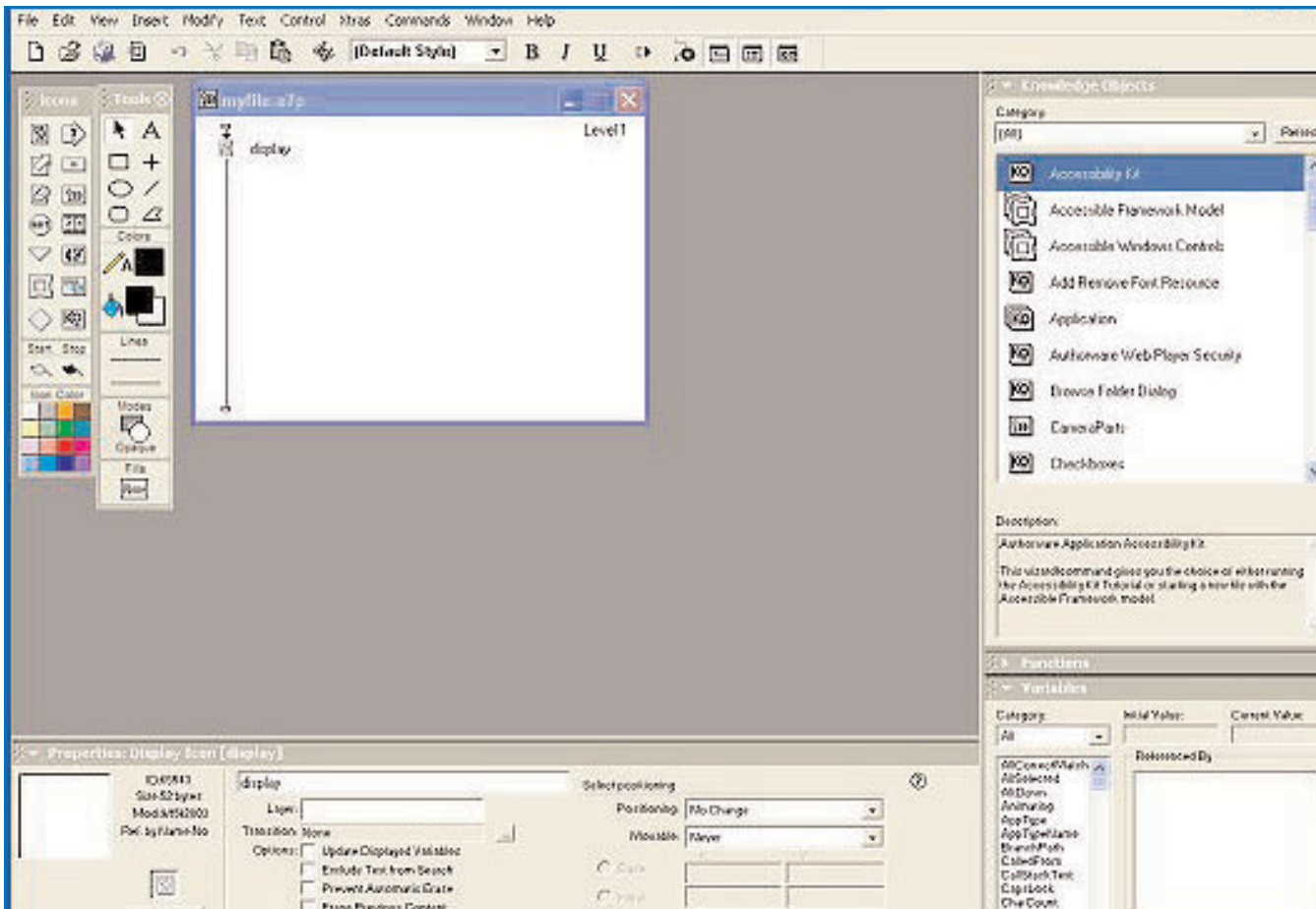


FIGURE 8 The user interface has been improved in Version 7 and now more closely matches other Macromedia products.



er. This leaves the screen less cluttered than in the past.

## **The PowerPoint Importer**

This is the feature that business owners and managers will probably find most appealing. For the first time, Microsoft PowerPoint files can be converted to Authorware files. This means that the PowerPoint pages become Authorware Map icons attached to a Framework icon. This icon structure can then be further enhanced inside Authorware. Many instructional designers use PowerPoint to create their storyboards. In addition, company presentations or other PowerPoint files that had been created for other purposes can now be converted quickly. Previously, images and text would have to be copied and pasted one by one, a very laborious and time-consuming process.

About 80% of PowerPoint features can currently be converted to Authorware icons automatically. Even if a PowerPoint file uses features that are not possible to convert automatically, the converter will still bring Authorware developers much closer to the final conversion, saving many hours of time.

### **SIDEBAR** *Authorware 7.0 top features checklist*

1. The user interface has been modernized.
2. PowerPoint files can now be converted to Authorware files.
3. Learning Management Systems can now be accessed much more easily through dialog boxes.
4. Creating accessible lessons has become even easier.
5. Any section of a DVD video can now be used in Authorware and displayed at any size.
6. JavaScript 1.5 scripts can now be written in Authorware.
7. Almost all icon properties have been opened to Authorware scripters.
8. Variables can now hold half a megabyte of data each.
9. The Help system has been streamlined.
10. You can now deliver to Mac OS X systems.

## **Learning management system (LMS) commands**

Talking to SCORM compliant Learning Management Systems has become a much easier process with the inclusion of three LMS commands:

- Authorware Learning Object Content Packager
- Authorware Learning Object Metadata Editor
- LMS Emulator

Using these commands means that developers can now set up communications with LMSs from within Authorware with simple wizards that hold their hands step-by-step. Anyone who has struggled with LMS connections in the past will appreciate these new commands greatly.

## **Accessibility improvements**

It's much easier now to use screen readers, develop accessible user interfaces, synchronize captions with video frames, and more. When creating new applications in Authorware 7, those with disabilities can now be served to unprecedented levels, helping to meet the American Disabilities Act, Section 508 requirements.

## **DVD video use**

The old Video icon, used in the early days to talk to videodisc players, but unused by the vast majority of developers in many years, has now been converted to a DVD icon, which lets developers use a dialog box to show DVD video on the screen in any size while retaining all of a DVD's interactivity, such as clickable menu options. This is a great boon for those organizations that have developed a good deal of video; now they can deliver their Authorware applications on the DVD itself (or alternatively, on their hard drives or through a Web browser) and access the high quality video resident on the DVD discs.

## **JavaScript Use**

Calculation icons can now accept scripts written in either the traditional Authorware scripting language itself or in JavaScript 1.5. This is a fantastic option for all those who already know how to write in that language, and also because a plethora of JavaScript scripts can already be found all over the Web. In fact, JavaScript is a native language of

Web browsers and is widespread in use so this makes it easier for other programmers to feel comfortable when coming to Authorware.

## **Properties**

Almost all icon options are now available to the scripter to help create truly dynamic files for students. Dynamic files are also highly maintainable, which cuts down on the amount of revision that normally would have to be done to files when the content has to change for external reasons — such as when a company process that is being taught has undergone modifications.

## **Other Improvements**

*Variables* are a feature common to every programming language. Each variable can hold one data item, whether it's a simple number or a long text passage. In past versions, variables in Authorware have always been limited to holding 32,000 characters of data, which at one time may have seemed enormous but has felt rather constraining in the last few years. Version 7 increases the amount of data a variable can hold by sixteen times, to half a megabyte.

The Help system is now a compiled Help file. The last several versions of Authorware used HTML-based non-compiled Help. While this may not seem a big deal, it meant that hundreds of small HTML files were being used, which greatly lengthened the installation of Authorware as well as taking up much more room on the hard drive. Help runs more smoothly and faster now, which means more users can take advantage of it.

Mac OS X delivery is now possible with this version as well, which will make Macintosh users very happy. This includes a stand-alone runtime player, publishing tool, and a Web player.

Overall, it is apparent that the engineers at Macromedia have worked very hard to keep Authorware a product that is easy to use by nonprogrammers while at the same time allowing programmers to be much more at ease in the product. This is not an easy balance, and yet it is pulled off more cleanly in Authorware than in any other product I've seen. It is not perfect, but it's getting closer to perfection with every release!





## AUTHOR CONTACT

Joseph Ganci is President of Dazzle Technologies Corp., located in Sterling, VA, just outside Washington, D.C. He has been involved in every aspect of multimedia and learning development with an emphasis in development since 1983. Joe has taught classes and seminars at leading universities and in many government facilities, commercial firms, and educational institutions, and has spoken at several industry conferences. He is also a published author, having written several books and articles about e-Learning and Authorware, and is widely considered a guru for his expertise in e-Learning development. Joe has consulted worldwide on a wide range of multimedia and learning applications for many organizations. Joe received his Bachelor of Science degree in Computer Science from Brigham Young University. Contact Joe at [jganci@dazzletech.com](mailto:jganci@dazzletech.com).

**NOTE:** Joe Ganci will be teaching a one-day workshop, "From Soup to Nuts: A Practical Guide to e-Learning Project Management" on November 11, 2003 at the upcoming *eLearning Producer Conference & Expo*. For more information on this workshop, visit <http://www.elearningguild.com/pbuild/linkbuilder.cfm?selection=doc.401>. Joe will also be leading concurrent sessions on the new features in Authorware (Session 205, 1:15 pm on November 12) and on selecting tools and getting them to work together (Session 705, 2:40 pm on November 13). See <http://www.elearningguild.com/pbuild/linkbuilder.cfm?selection=doc.425> for details on these sessions.

**Additional information on the topics covered in this article is also listed in the Guild Resource Directory.**

**Do you have an interesting story to tell?**

**Get It Published in...**

**THE eLEARNING  
DEVELOPERS'  
JOURNAL**

If you have an idea for an article, send a plain-text email to our editor, Bill Brandon, at [bbrandon@eLearningGuild.com](mailto:bbrandon@eLearningGuild.com)

**Please refer to [www.eLearningGuild.com](http://www.eLearningGuild.com) for more details.**



**The eLearning Guild™** is a Community of Practice for designers, developers, and managers of e-Learning. Through this member-driven community, we provide high-quality learning opportunities, networking services, resources, and publications. Community members represent a diverse group of instructional designers, content developers, web developers, project managers, contractors, consultants, and managers and directors of training and learning services — all of whom share a common interest in e-Learning design, development, and management.

## The eLearning Developers' Journal™

The Guild publishes the only online "e-Journal" in the e-Learning industry that is focused on delivering real world "how to make it happen in your organization" information. The Journal is published weekly and features articles written by both industry experts and members who work every day in environments just like yours. As an active member, you will have unlimited access to the Journal archive.

## People Connecting With People

The Guild provides a variety of online member networking tools including an Information Exchange and a Job Board. These services enable members to discuss topics of importance, to ask others to help them find information they need, and to provide leads to other members.

**Become a member today! Join online at [www.eLearningGuild.com](http://www.eLearningGuild.com).**

## THANK YOU TO THESE GUILD ENTERPRISE SPONSORS



Because the most dramatic learning happens through first-hand experience, everything we do is focused on creating experiences — engaging, challenging and yes, fun, experiences — that people will take back to work and use every day to improve their performances.

**[www.alleninteractions.com](http://www.alleninteractions.com)  
Contact: Jackie McMillan  
[jmcmillan@alleni.com](mailto:jmcmillan@alleni.com)  
800.204.2635**



**Cyclone Interactive** is an interactive media and web development firm creating online, CD and presentation solutions for a wide range of clients and industries.

**[www.cycloneinteractive.com](http://www.cycloneinteractive.com)  
Contact: Earl Dimaculangan  
[earl@cycloneinteractive.com](mailto:earl@cycloneinteractive.com)  
617.350.8834**

To learn about **Guild** sponsorship opportunities, please contact David Holcombe at [dh@eLearningGuild.com](mailto:dh@eLearningGuild.com) or call 707.566.8990.