IN THIS ARTICLE

Authoring Systems

Smarter, Faster

HERE'S HOW PACIFIC BELL USED NEW TECHNOLOGY TO CHANGE THE WAY IT DEVELOPED CBT. THE NEW TOOLS SAVED MONEY AND TIME—AND ENHANCED LEARNING.

BY DAN B. ZIAGOS

CBT Development

ANY TRAINER OR performance specialist would like to develop CBT courses faster and better. Not surprisingly, Pacific Bell found that technology was the key to improving technology-based training.

In 1994, Pacific Bell's Education and Training Department embarked on an initiative to change how it produced its computer-based training. The goal was to reduce the course-development cycle by 20 to 40 percent and still present information in a self-paced, interactive environment using multimedia to enhance learning and retention. PBET hoped that a shorter cycle time would lower the development costs of its training products, accelerate training delivery, and save Pacific Bell money.

For more than 10 years, PBET has been developing CBT. As technology progressed with better hardware and software tools, the multimedia elements within CBT have become more complex. The approaches we've been using for CBT development and delivery at Pacific Bell have also evolved, but in incremental steps.

Typically, we use a five-phase approach: analysis, design, development, implementation, and follow-up evaluation—spending 20 percent of the time on analysis, 20 percent on design, 40 percent on development, 15 percent on implementation, and five percent on evaluation.

The idea of the new initiative was to shorten the design and development phases in particular by changing team members' responsibilities and by using models and authoring-shell templates.

First, some definitions

Authoring-shell template. This is a software tool used to develop interactive lessons, reviews, and evaluations. It features navigation capabilities and such pre-programmed functions as a glossary, help screens, a comments section, and "placeholders" to mark content.

Authoring systems. This software application is used to create the authoring-shell templates. It has the capabilities to test users on training content, judge their answers, and branch to a specific location, based on a user's entry. It also integrates such multimedia elements as graphics, animation, audio, and digital video. Most authoring systems include menus or icons from which authors can choose various programming options.

Models. These are small "pieces of logic" that can be plugged into a template for a specific function. For example, a multiple-choice model can be plugged into a lesson template to create an interactive question-and-answer test.

Models are a bit like knitting kits. The overall kit would be the authoring system; both include instructions and patterns. A sweater-making kit might have different patterns—such as pullover, turtleneck, and cardigan. A sweater pattern is analagous to an authoring-shell template in that each pattern and template have a pre-designed format. In the case of a template, the format could be an interactive lesson, a review, or an evaluation. The models are like a sweater's pockets or embroidered design. They're "accessories," or extras, integrated into the final product.

A team effort

In the past, course development was usually done by a PBET team made up of a course developer responsible for a specific curriculum, a multimedia-training consultant (an expert in the design and development of CBT), and a graphic artist. Other PBET teams might have additional members—such as a media director, a project manager, or an instructional designer who served as both the course developer and multimedia-training consultant.

The approach focused on the team members and their tasks. The course developer created paper-based course material (with the help of a subject matter expert), produced trainee and facilitator guides, and conducted development and field tests.

The multimedia-training consultant reviewed the course material and redesigned it for CBT delivery. He or she also identified the graphics requirements, created CBT, and integrated the graphics. The graphic artist created the graphics.

That mode of operation was successful in that it produced high-quality CBT and met training objectives. But it was labor-intensive in terms of hours—in our experience, four to 10 times more intensive than development for instructor-led training. Consequently, the department rolled out fewer CBT courses each year than instructor-led courses.

Yet, cost-benefit analyses showed that CBT delivery can be a cost-

effective and learning-effective training medium. Even though CBT-development costs tend to be higher than development costs for instructor-led training, the savings in CBT-delivery costs more than make up for the difference.

Such savings include

- the elimination of expenses for travel and classroom setup
- reduced costs for printing and distributing materials
- less class-preparation time
- less training time (25 to 40 percent less compared with noninteractive media)
- fewer productivity losses due to less time spent in class.

CBT lets trainees take only the course modules appropriate for their jobs, and trainees can skip a topic if they demonstrate the required knowledge or skills.

Not surprisingly, we decided to develop more CBT, based on the potential savings and the rising number of courses appropriate for CBT delivery. We wanted high-quality CBT to serve as performance-support tools. And we wanted to make the tools easily accessible, consistent, timely, efficient, and user-friendly.

A new approach

PBET considered outsourcing CBT development but decided instead to try to make a significant change in the development process by enhancing the skills of internal course developers in CBT development.

The new tasks of the course developer would be to

- create course material in text files formatted for authoring-shell templates, with the help of an SME
- specify the requirements for graphics and customized models
- create CBT by transferring course material from text files to templates and by integrating graphics
- create trainee and facilitator guides
- conduct development and field tests.

Using templates gives course developers greater responsibility and flexibility in designing their training products

The multimedia-training consultant would build custom models and maintain a models library. He or she would also provide internal hotline support to course developers. The graphic artist would create the graphics.

Though the new approach involves more tasks, the overall process is shorter. The course developer spends more time involved, but the multimedia-training consultant spends less time, mainly because there's no need to rekey paper-based course material. Authoring time is less because of the templates and models. The overall reduction in development time is 20 to 50 percent. (See the table.)

TIME SPENT

Team Role	Old Approach	New Approach
course developer	60 hours	200 hours
multimedia-training consultant	300 hours	60 hours
graphic artist	40 hours	40 hours
Total =	400 hours	300 hours

The new mode of operation also involves more internal training resources directly with CBT development, particularly in authoring. We were careful to purchase an authoring system and templates that matched course developers' skills and experience. Most of the products are menu- or icon-driven, enabling first-time CBT authors to create courses, without having the hard-coding knowledge of experienced programmers. The use of templates to support the authoring-software tools gives course developers greater responsibility and flexibility in designing and upgrading their training products.

Barriers and bugaboos

The course developers, multimedia-training consultants, and graphic artists received one week of training on the authoring templates and models. Course developers received an additional week of training on the authoring system, which served as the engine for the templates and models. Despite the training, several barriers remained.

One, some course developers did not have the appropriate skills—such as knowing how to use a graphical-user-interface operating system. Or, they didn't understand basic programming skills.

In addition, we thought that some employees might reject the new technology, or that they would be turned off by a lack of consistency among the CBT products or their new look and feel. One glitch was that the new training lacked audio capability because employees' hardware didn't include sound cards.

Another problem was that most course developers were working on their first CBT project. Though they were skilled developers of instructor-led training and knew accelerated-learning techniques, they had received no formal training in CBT design and development. They did receive some coaching, provided by their team leader and the multimedia-training consultants.

At first, the interaction designs more closely resembled designs produced for self-paced, paper-based training than CBT. For example, remediation consisted only of corrective feedback. Typically, corrective feedback looks like this: "Sorry, that's not correct. The correct answer is A." An improved CBT design would give this feedback: "A is the correct answer because you need to...."

The first designs also didn't provide analogies with the corrective feedback, entry-specific remediation (a screen showing explanatory feedback to several questions), or branching based on performance.

For example, with branching, if a trainee gets fewer than four correct

answers in a five-question quiz, a review screen appears so he or she can retake the missed questions.

In addition, course developers were slow to follow the CBT practice of keeping learning events short and concise.

At different times, team members realized that the templates weren't fancy wordprocessors. But early in the individual projects, teams recognized that they had to pay attention to detail and that troubleshooting would be complex. Initially, they experienced frustration at not being able to solve problems quickly, even those created by what they deemed "little mistakes." But by continuing to use the templates and models, the course developers increased their confidence and performance.

Success at last

Eventually, we overcame the barriers. The course developers learned the necessary skills for becoming efficient CBT authors. And they learned how to use the templates and models, with the support of the multimedia-training consultants and graphic artists. What's more, trainees didn't object to differences in the look and feel of CBT tools or the lack of audio.

Our criteria for success was a 20 to 40 percent reduction in the development ratio (development hours to training hours produced). The baseline was our former mode of operation in which the mutimedia-training consultant alone would author a CBT course using an authoring system but not templates and models. The CBT courses created during the new initiative measured a 20 to 50 percent reduction in the development ratio.

Smoothing the glitches

At the outset, team members determined that training would be delivered to end users at their workstations. Due to heavy work loads, most employees found it difficult to leave their workstations for training. But it was also difficult for them to receive training at their workstations because of interruptions. So, desktop training in 20- to 30-minute lessons worked best.

Employees received the CBT courses via floppy disks, which they loaded onto their hard drives. When they finished a course, they returned the floppy disks to PBET for course credit and an update of their training records.

The criteria for success was a 20 to 40 percent reduction in development hours to training hours

But that process proved cumbersome to administer. Trying to establish clear procedures for recording course credit, for charging tuition, and for permitting user access was tedious, labor-intensive, and sometimes inaccurate. And it was necessary to add version control.

For example, PBET rolled out a CBT course in January, 1995. After three months, it was necessary to make some revisions. So, we made the modifications and rolled out a new version in June. But we found that we still had to make new disks. Because some of the old disks were still in use, there was confusion about the different versions, despite dates on the disk labels.

To make administration easier, PBET purchased a commercial software application, renaming it the "Pacific Bell Learning Manager." PBLM is a network-based training system that lets employees receive training materials at their computers. It can customize curricula for individual learners, track their progress, and generate training records. PBLM also allows simultaneous access by multiple users, storing CBT courses on a fileserver.

For the second initiative, PBET downloaded (to a network hard drive) a compressed CBT file—an electronic file condensed by the temporary removal of redundant information, up to 50 percent smaller than the original. But the network's bandwidth limitations made it difficult to transmit large-sized multimedia elements. So, PBET decided to download one CBT lesson at a time, when trainees requested it. That was an easy chore for the network, compared with accessing multimedia courses from a fileserver in real time from a remote-site computer.

Once it reached a trainee's hard drive, the compressed CBT lesson decompressed automatically, ready for the trainee to begin the lesson. In a decompressed file, the redundant information is restored. PBLM eliminated the needs for manual disk duplication and for distribution, billing, and course-credit recordkeeping. It also resulted in cost savings for PBET.

The results taught us that it's crucial to seek constantly for smarter ways of doing business. We must find better ways to reach our goals, increase our clients' performance, and improve our business processes and products. Currently, PBET is investigating the World Wide Web as a possible applications environment, looking at Web-based software tools for developing CBT and new modes for delivering multimedia elements via the Web. ■

Dan B. Ziagos *is a multimedia-training consultant, Education and Training Department, Pacific Bell, 2600 Camino Ramon, San Ramon, CA 94583. Phone 510/823-5716; fax 510/830-0971; e-mail dbziago@PacBell.com.*