LITERACITY: A MULTIMEDIA ADULT LITERACY PACKAGE COMBINING NASA TECHNOLOGY, RECURSIVE ID THEORY, AND AUTHENTIC INSTRUCTION THEORY

Jerry and Dee Anna Willis, Clare Walsh, Elizabeth Stephens, Timothy Murphy, and Jerry Price
Center for Information Technology in Education
College of Education, University of Houston
Houston, TX 77204

William Stevens
Technology Utilization Office
Lockheed Engineering & Sciences Company
NASA, Lyndon B. Johnson Space Center, Houston, TX

Kevin Jackson
National Institute of Corrections
Department of Justice, Washington, D. C.

James A. Villareal and Bob Way
Advanced Software Architectures, Software Technology Branch
Information Systems Directorate
NASA, Lyndon B. Johnson Space Center, Houston, TX

ABSTRACT

An important part of NASA's mission involves the secondary application of its technologies in the public and private sectors. One current application under development is LiteraCity, a simulation-based instructional package for adults who do not have functional reading skills. Using fuzzy logic routines and other technologies developed by NASA's Information Systems Directorate and hypermedia sound, graphics, and animation technologies the project attempts to overcome the limited impact of adult literacy assessment and instruction by involving the adult in an interactive simulation of real-life literacy activities. The project uses a recursive instructional development model and authentic instruction theory.

This paper describes one component of a project to design, develop, and produce a series of computer-based, multimedia instructional packages. The packages are being developed for use in adult literacy programs, particularly in correctional education centers. They use the concepts of authentic instruction and authentic assessment to guide development. All the packages to be developed are instructional simulations. The first is a simulation of "finding a friend a job."

THE DESIGN PROCESS

To develop the instructional package we used a heavily modified version of a traditional instructional development (ID) procedure. We adapted the Four D instructional development model of Thiagarajan, Semmel, and Semmel [1], but it is very similar to models developed by Dick and Carey [2] and many others. This model has four stages: (1) Define, (2) Design, (3) Develop, and (4) Disseminate. Although this model does provide a framework within which to organize and manage the process of developing instructional materials, it has a decidedly behavioral and linear flavor. Because the instructional packages we are developing are based on cognitive models of learning and teaching, the relatively poor "fit" between the ID model we had adopted and the process we were actually using quickly became obvious. Traditional ID models seem most suitable when the goal is to develop traditional, text-focused tutorials based on relatively standard behavioral learning theory. A project that strays from tutorials or from behavioral learning theory may find many aspects of the traditional ID models become barriers rather than facilitators to smooth, effective development of materials. In this project, the book, *Learning With Interactive Multimedia: Developing and Using Multimedia Tools in Education*, by Sueann Ambron and Kristina Hooper [3] was a major inspiration for modifying and transforming the ID model we were using.

In the following sections, each of the stages in the Four D model is summarized, with extensions and revisions related to the development of multimedia materials based on cognitive instructional theories. Most of the extensions and revisions reflect two major conceptual shifts in the ID model we used: recursion and reflection. While the 4 D Model is linear, we used it recursively which reflects one aspect of a more cognitive model of learning as well as of instructional development. The steps in the Four D model are described below in sequence that implies they are stages or phases in the ID process which must be completed one after the other. As we used the model, the 4 Ds were not steps to be completed one after the other as if they were layers of bricks in a wall. They were, instead, tasks to be completed. Unlike each layer of bricks in a wall, the tasks in the process may be addressed many times as the project progresses. Design and Development is thus a recursive rather than a linear process.

The other major change in the Four D model involves reflection in action. Schon's work on reflective practice [4] emphasizes the need to think about and revise practice based on careful observation and analysis of what is happening in the practice environment. That approach, when applied to ID, places a heavy emphasis on obtaining feedback from students and instructors in the environment where the material will ultimately be used. In many ways, this represents an expansion of the formative evaluation procedures included in many behavioral ID models. The ID model we use in this project is thus a Recursive, Reflective Design and Development Model (R2D2).

THE DEFINITION TASK

In the traditional ID model this task includes the preliminary steps required to develop any type of instructional material. The purpose of Definition subtasks is the identification and definition of instructional requirements. The five subtasks in this stage include (1) front-end analysis, (2) learner analysis, (3) task analysis, (4) concept analysis, and (5) specifying instructional objectives.

Front End Analysis

The initial step in developing instructional material is an evaluation of the need. This process generally involves a specification of the need and an evaluation of existing materials to determine if needs are already being met. The front-end analysis for this project indicated some computer-supported instructional materials were available for adult literacy programs, but almost all of them were drill and practice or tutorial packages designed from a distinctly behavioral perspective.

Learner Analysis

This step traditionally involves developing a clear understanding of the target students. In this case, the primary users are adult literacy students. We treated learner analysis as an on-going process rather than a preliminary phase of development. We worked with a number of students at the Houston Read Commission, but the format was not a traditional learner analysis that involves assessment of learner skills. Instead, the students became part of the development team. We asked for input on interesting themes for the simulation, discussed topics that should be in the simulation, and received feedback on visual "look and feel" issues as well as instructional strategies the students preferred. Instructors at the Houston Read Commission also gave input on what their students needed, what they preferred, and ways the simulation could be made more interesting and relevant.

Task and Concept Analysis

After the need has been established and relevant student preferences, characteristics, and needs defined, the next step in a traditional ID model is to define the major skills to be acquired by the students and to analyze those skills. In traditional behaviorally oriented projects the focus is on breaking things down into subunits or components that can be taught separately. In reading, this approach leads to a discrete skills model of instruction that targets things like short vowel sounds and consonant digraphs. The result of this model in adult literacy programs is often relatively boring exercises that attempt to remediate weak underlying skills on the assumption that when the student develops those cognitive "muscles" reading will improve because reading is really a collection of skills. This model has not served adult literacy well and it may be one reason for the high dropout rate in many programs as well as the lack of enthusiasm shown by many participants who truly want to learn to read. In addition, many adults who do not have a functional level of reading ability have already completed and/or failed at many such exercises during their school years and have significant negative associations with them.

Using the R2D2 Model we approached task and concept analysis from a somewhat different perspective: authentic assessment and instruction. The primary goal was to deliver instruction and provide assessment directly related to authentic tasks an adult in this culture completes by reading. Those tasks are the focal point of the project and both instruction and assessment focuses on them. Thus, during instruction the task of the reader will be to accomplish a common task such as finding and applying for a job, locating and renting an apartment, using social services, or buying groceries. Assessment focuses on aspects of the reader's approach to the task that limit or prevent success. That is not to say that no discrete skills analysis or skills instruction is conducted or used. It simply means that it is framed within the context of how it relates to a particular task that is real or authentic for the student.

When task and concept analysis is considered within the framework of authentic assessment and authentic instruction, the analysis becomes less a job to be done once and for all at the beginning of the development process and more a continuous aspect of ID. For the project described in this paper the task was "Finding a Friend a Job" which involved selecting appropriate jobs, obtaining and completing the application, and submitting the application to the potential employer. The instructional product was built around the task. Actual instruction and help learning to read is available to the student when, and only when, the student requests it. The instruction and help given is determined by the student who selects it in order to complete the authentic task at hand.

Specifying Instructional Objectives

This is normally the final step in the first stage of a traditional ID model. It involves converting the results of the analysis of tasks and concepts into a set of objectives. They in turn provide the basis for designing the instructional package and developing evaluation and assessment strategies. As with task and concept analysis, specifying instructional objectives changes somewhat when viewed from the perspective of the R2D2 Model. In behavioral theory, specific instructional objectives such as "When the reader is faced with a paragraph of directions on how to drive from home to a potential employer's office, he or she draws a correct map based on those directions" are needed to guide the design process. In the R2D2 Model, which is based on authentic instruction and assessment, specific objectives evolve naturally out of the creation of a realistic instructional environment that represents a real world literacy task. Objectives keep designers focused and help ensure that the instruction leads to learning that transfers to the real world. It is not as important to write specific objectives first when the design involves creating an instructional version of some aspect of the real world. Concepts such as information landscapes, user support, and user interface become much more important than the development of a long list of specific instructional objectives.

THE DESIGN TASK

The focus of this task is the design of prototype instructional material. Two subtasks in most traditional ID models, Media Selection and Format Selection, are completed along with a third, Selection of a Development Environment. Two other traditional subtasks, Initial Design and Evaluation Strategy, are folded into other tasks rather than treated separately.

Media and Format Selection

The media we selected was interactive multimedia and the format selected was simulation. Because of the type of computer and media resources available, or affordable, in most adult literacy programs, we selected a single-screen simulation that can be run on a Macintosh or IBM computer with a CD-ROM drive and a sound system. The package puts the reader in the primary role in a simulation of an interesting and typical real world activity - finding a job. During the prototype development stage, the entire package, including sound effects, music, and speech files were stored in files on the computer's hard disk drive. The final version will be distributed on a CD-ROM that includes the computer program, including graphics, in track 1 with all the sound in CD-Audio format on tracks 2 through 99.

The "Finding a Friend a Job" scenario used in the first instructional package requires the reader to select a friend (Figure 1.), select a job from those advertised that fits the friend's needs and abilities (Figure 2.), and correctly complete a three part job application that serves as a model for the friend (The second part is shown in Figure 3.) . If the job selected matches the needs and talents of the friend, the simulation ends with a call from the friend who says "I got the job .... and I could not have done it without your help." If the job selected is

Figure 1.

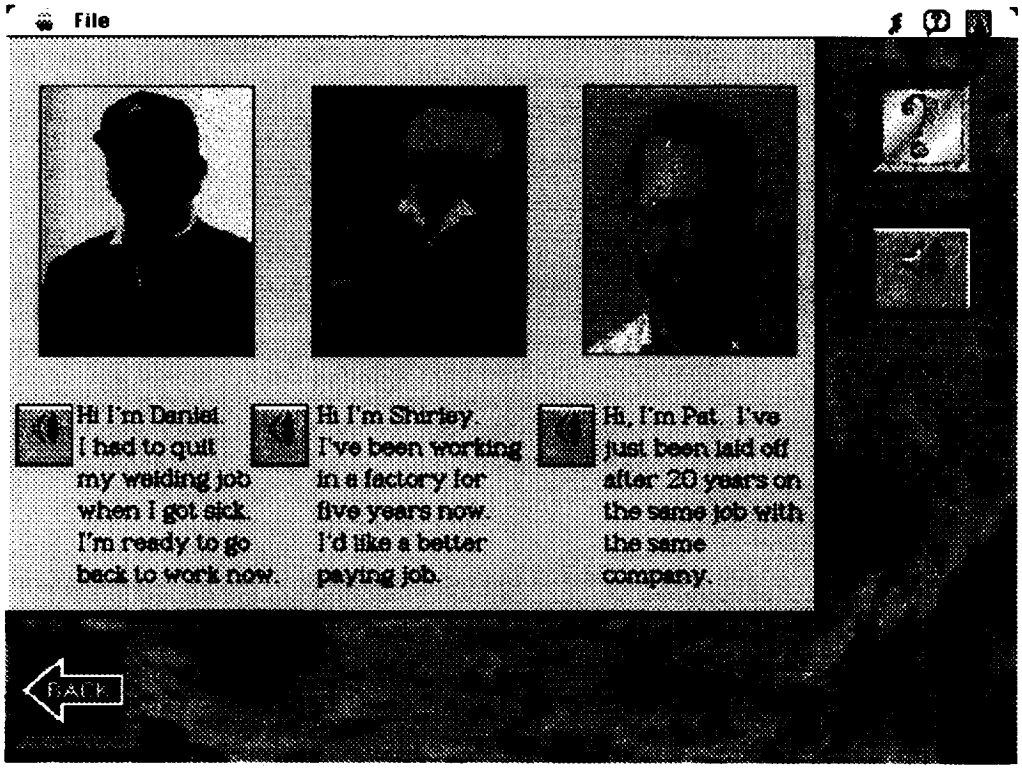
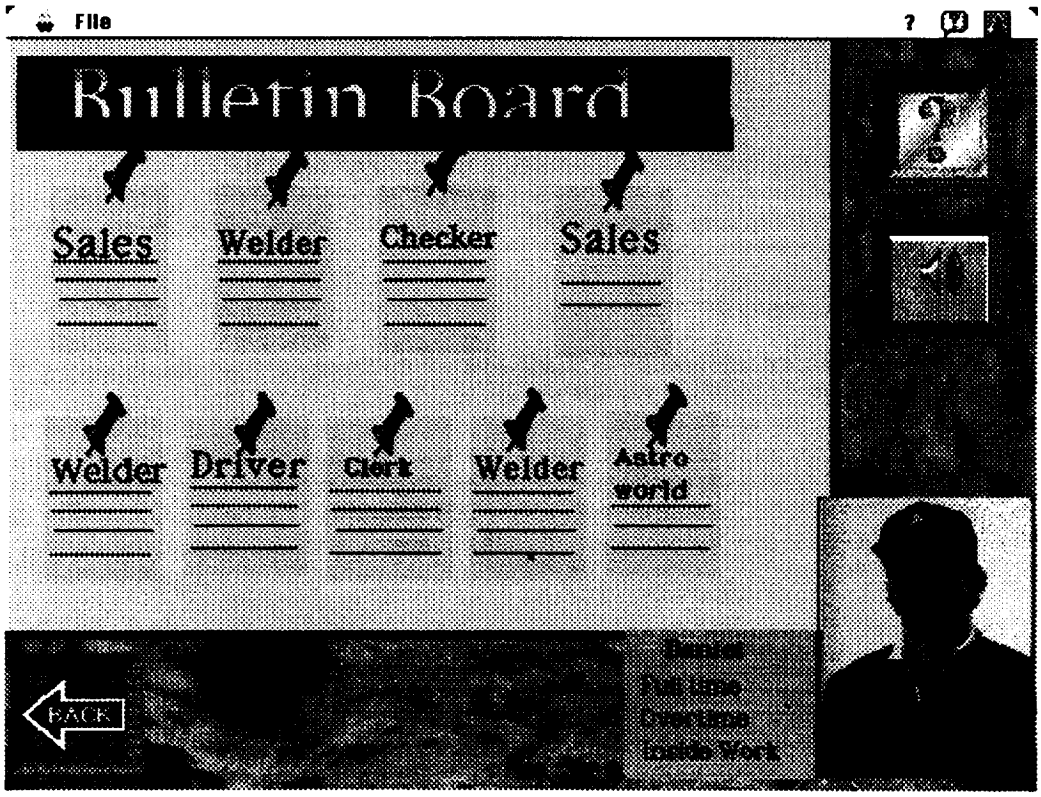


Figure 2.



inappropriate the friend calls to tell the reader there was a problem and asks for more help. The simulation can then begin again.

Throughout the simulation, the reader can switch from simulation mode to help mode by clicking a button. When the reader does not know a word or phrase, he or she can ask for assistance. Readers click the problem word or words and then choose which of five different reading strategies they think would be most helpful. The five strategies, which are represented by 5 icons at the bottom of the help screens, are phonics, syntax, context, pronounce for the student, and read and explain. Figure 4. illustrates one step on phonic analysis of the word *duties*. For each strategy selected readers apply it with the guidance of spoken directions. Throughout the simulation, readers can ask for help by clicking a question mark button and replay verbal directions and explanations by clicking a speaker button on the top right of the screen. They can replay the voice version of printed directions, explanations, and information by clicking small speaker icons displayed beside the text. Spoken help, directions, and assistance is available in either Spanish or English.

Selecting a Development Environment

The core of the development environment was Authorware Professional on the Macintosh (APM) supplemented by C language subroutines as well as sound and graphics software including Photoshop, SoundEdit, SoundWave, and Canvas. Graphics, music clips, and sound effects were used from a number of clip collections including Killer Tracks music clips and Wraptures backgrounds. A major advantage of APM as the development platform is its ability to support a wide range of utility and support software and hardware. APM has provisions for integrating sound including music clips and voice from many sources, still images and video from CD-ROM and laser discs, and graphics from many paint and draw programs.

Authorware Professional for the Macintosh (APM) is widely recognized as a powerful development system. It is used by a number of NASA subcontractors, including Rockwell, and by Boeing. APM has a number of advantages over developing instructional packages in standard programming languages such as C. Perhaps the most important advantage for this project is the ease of learning factor. Experienced instructional designers can be "up and running" in Authorware in less than a week while estimates of the time it takes someone to become a proficient independent C programmer are as high as two to three years of full-time work and study. In addition, products developed on APM can be transferred to the Authorware Professional version for Windows (APW). APM's ease of learning, and ease of use, facilitates an important aspect of the development team's work pattern. In traditional ID models, each person takes a specialist's role. For example, a graphic artist may create illustrations, an instructional designer creates sequences or frames of instruction, and a content expert either creates or approves of the content included in the package.

In the R2D2 Model as used in this project, the team was composed of "specialists" who were primarily responsible for different aspects of the program such as graphics or sound. One member of the team, however, served as the designated programmer. The roles were, not, however, watertight compartments. Instead, almost all the team members worked some on all aspects of development. The content specialist, for example, wrote and revised some of the Authorware code and redesigned some of the icons. The sound specialist wrote some of the script and designed several of the help and instructional strategies. APM's ease of use made it possible for every team member to program. The interactive nature of APM made design recursion possible. Without a development environment like APM, reflection and recursion would be less useful, or discouragingly useless, because it is too difficult to make changes and adjustments to the program once it is "set in code." For example, instead of storyboarding an instructional sequence and then turning it over to the programmer to be written, the team could quickly answer many "what if" questions. What if we change the location of the icons, what if we add an explanatory screen between two existing screens, what if we change the layers a reader moves through to get help with an unknown word? In APM it is easy to set "Start" and "Stop" flags at points in the program, make changes in that section, and then run only that section to see how it will appear to students. The same feature makes it very easy to get student and teacher input and suggestions on different components as the program is evolving. For example, the team created four different "look and feel" designs and asked sixty adult literacy students to select the one they preferred. The same group listened to eight different musical styles and selected the one the liked the best. Such on-going and recursive feedback-revise cycles are much easier to complete using APM than regular programming languages.

Another prime advantage of APM is that its underlying language is C. This allows NASA'a fuzzy logic , also written in C, to be easily interfaced with the created simulation and to analyze data produced when the

Figure 3.

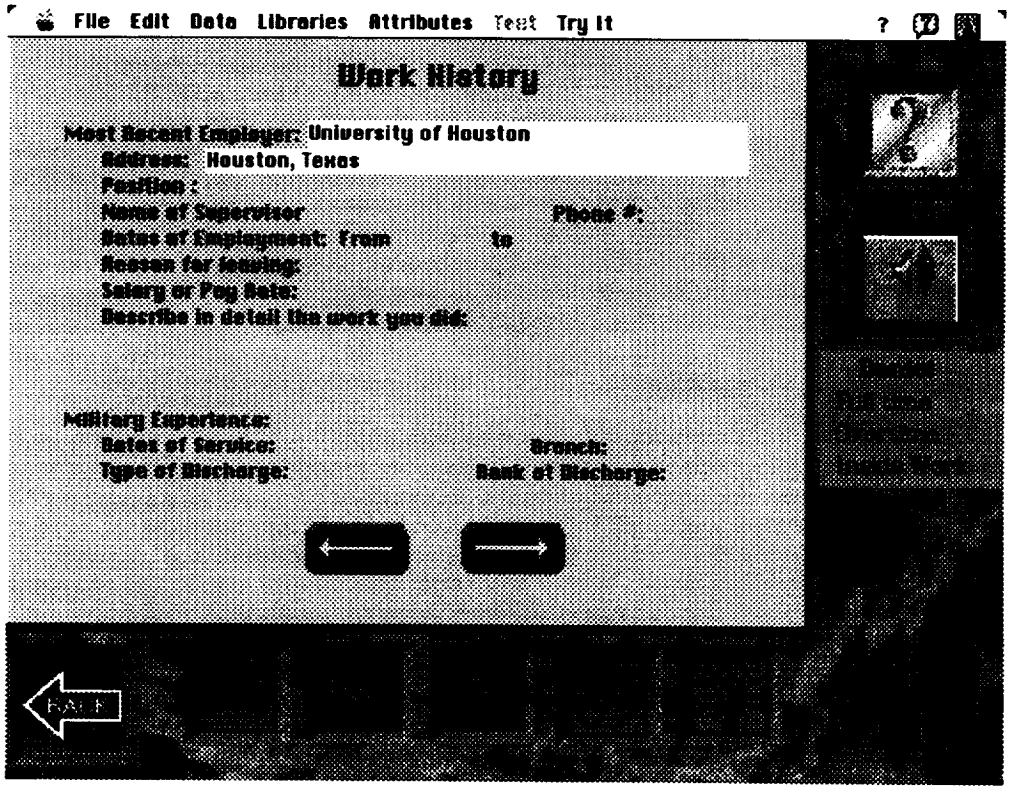
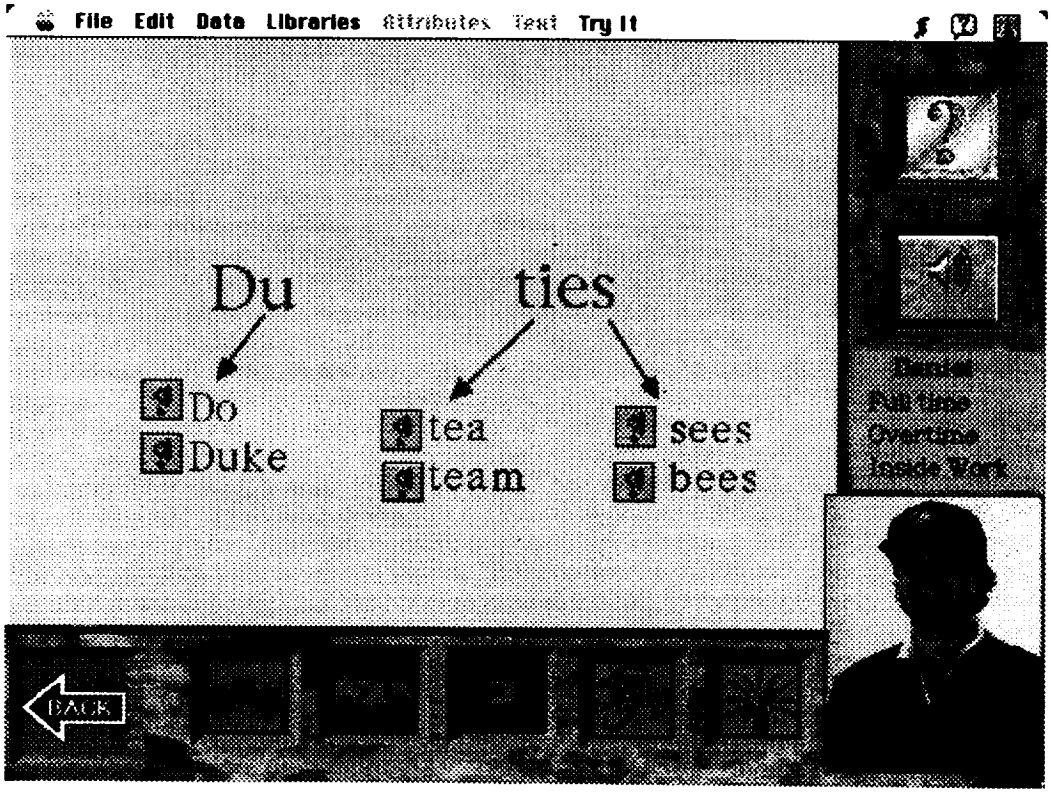


Figure 4.



simulation is run. While the major emphasis of LiteraCity is placing control in the hands of the learner the data gathered, once analyzed and reduced to more manageable forms, can be of help to instructors guiding learners.

APM is not a perfect development environment, but it has much to offer when the ID model is recursive and reflective. An educational product is as much an artistic creation as it is a technical creation. We are all aware of technically correct educational materials, manuals, and documentation that numb the mind and discourage their use. This project aims to create a product that appeals to the user, one that has the snap and crackle that comes from well placed and well designed graphics, a product that has an appealing premise creatively executed. A traditional, top down, linear model of development is unlikely to produce such a product. What is needed is a development model that is the equivalent to the "management by walking around" movement in corporate leadership. That is, the designers and developers need to be able to "walk around" the program and try out possible alternative designs and arrangements with regular and ongoing feedback from potential users and consumers. The approach is easily implemented with Authorware. Changes, even major changes, are relatively easy to make while your program is running. And changes, once made, can be run immediately to view the effect from the perspective of the student who will be using the package. This advantage, which avoids the designer-to-programmer-to-software-to-designer cycle that has doomed so many instructional development projects, allows developers to "play with" many different alternatives in much the same way a musician can play with changes in a musical composition. This non-linear, real time approach supports improvisation, which leads to exciting designs. Eric Holsinger, in his article titled "Nonlinear systems enhance video editing" [5] makes the same point about electronic video editing. "Unlike standard video-editing systems, an editor can quickly cut and paste sections of the presentation together in any order. A user can save and compare different edits of the presentation easily, whereas starting a new version on a traditional off-line system requires you to start the edit from scratch each time."

Initial Design and Evaluation Strategy

The final two steps in the traditional Design stage are Initial Design and construction of an Evaluation Strategy. Because this project involves a product based on authentic instruction and authentic assessment theory there is less concern about a separate 'test" of success. The simulation has a built-in assessment component and there is much less need for an external evaluation strategy such as pre- and posttests. The focus was thus on Initial Design. We approached the task of creating an instructional simulation by first developing an overall concept. Then we worked concurrently on three aspects of the general format and design:

1. Surface design - screen layout, typography, language, graphics, illustrations, sound.
2. Interface design - look and feel, user interaction, help, support, navigation, metaphors.
3. Scenario - sequence of simulation, options/choices, results.

We approached these three components by defining one path through the simulation that included every type of interaction, every type of help, and every type of outcome and feedback. We then completed all the work on that path so that the simulation ran from start to finish as planned - if one particular path was selected. At this point we have completed the surface design, interface design, and scenario for a single path. After students and teachers critique this path and revisions based on those critiques are completed, we will use the Authorware components in this path to build all the other paths. This reduces the effort required to develop the full simulation. One path serves as a template for the other paths through the simulation. In addition, many components in the Finding a Friend a Job simulation will be reused in the next simulation in the LiteraCity series.

THE DEVELOPMENT TASK

As the Design Tasks are completed, the Development Tasks become more and more important and relevant. An initial prototype is collaboratively developed and evaluated. Collaborative development involves the entire design team as well as consultants such as front line practitioners with recognized expertise and experience. As components of the prototype were developed they were subjected to two types of evaluation: expert appraisal and developmental tryouts. Both provide feedback that will be used to revise the prototype. In this project expert appraisal of the package was obtained from adult literacy practitioners and correctional education specialists. Developmental testing was carried out in Houston-area adult literacy programs.

We currently have a working prototype developed and before the program is ready for distribution the develop-get feedback-revise cycle will be completed many times for each component and several times for the complete prototype. Because Authorware is a flexible development environment it will be easy to make changes based on the feedback from both expert appraisal and developmental tryouts.

DISSEMINATION

Once a package has been revised based on feedback from expert appraisals and developmental tryouts, the final product is subjected to a summative evaluation. We have not reached this stage yet. The summative evaluation will involve using the package with a fresh group of students in an environment like the one where the material is to be used. Qualitative and quantitative data gathered from this evaluation will be included in the final teacher's manual included with the package.

The three Dissemination subtasks are Final Packaging, Diffusion, and Adoption. Final packaging involves creating and producing any necessary print materials (such as student guides and teacher's manuals), creating a master for the CD-ROM or laser disc and pressing a quantity of CD-ROM for initial use. If we are able to attract a commercial partner before final packaging, some of the work in this phase will be handed off to the partner. The commercial partner will also carry the primary responsibility for diffusion and adoption.

THE DESIGN AND DEVELOPMENT PROCESS

For this project a design team was organized. A wide range of specialists were required including programmers, instructional design specialists, content specialists and instructors, computer graphic artists, and project management leaders. The development of an effective team is not an easy task. There are, in fact, probably many more failures than successes in this field. The difficulty of the task is illustrated by Mark Heyer's article in the Microsoft Press book, *CD-ROM: The New Papyrus*. In his chapter Heyer, who has worked with Sony and Group W Westinghouse on CD-ROM and laser disc projects, describes what is hopefully a worst case scenario:

The conception, design, and production of visual/computer interactive programming is the newest challenge facing the creative community. . . .

In the videodisc industry, some organizations understand this problem [getting designers, visual producers, and programmers to work together] and have begun to create teams. In most cases, though, members of the three groups are separated in space and time. A few companies have even mandated that programmers will not talk to designers or video producers -- a clear failure mode.

Generally the process follows a path something like this: First a design document is generated and argued about for four to six months. So far it's a paper war. At best, 10 percent of the people who are judging and modifying the paper concept will actually understand the nature of interactive video.

Next the video is produced. Whether the content is motion or stills, this step normally consumes 80 percent of the total project time and budget. . . .

Then, after the visual material is mastered on a video or data tape, the control software is needed. In the case of videodiscs, most of which use external computer control, the disc is pressed and then the control software is written to conform with what has been put on the discs. Sadly, in many cases the programmer isn't even hired until after the disc is done. At that point the opportunity to make changes has passed. . . .

CD-ROM has an extra burden to consider, since the control software has to be pressed on the disc along with the visuals. Everything has to be perfect the moment the master is made. In reality, many producers will bear the expense of pressing discs with just the visuals as an aid to programming. This works about as well as any batch processing scheme, but it still doesn't allow for many visual change cycles, or for creative design input during programming. The computer programmer works with a fully interactive computer programming language and attempts to control a videodisc, which can be done in many ways, but the read-only pressed videodisc is in fact fixed and unchangeable.

The challenge of understanding these problems, and to some extent working around them, has occupied a whole generation of videodisc producers. We are just now in possession of design tools and methods which dramatically cut the development time and costs for videodiscs. The same techniques and equipment will work for CD-ROM [6].

Heyer paints a fairly accurate picture of many development efforts. Fortunately, he also offers some very useful suggestions. He suggests, for example, that design tools and systems must be developed and used that are relatively inexpensive and easy to use. Perhaps the most important point he made relative to this project relates to what he calls Interactive Editing. "For creative artists and designers, whether visual or computer, the ability to make unlimited small changes is an absolute benefit." Heyer suggests using a videodisc emulator so that "designers, visual artists, and programmers are working together on program creation in real time. Hundreds of changes can be made in a day, and each change is instantly demonstrable. Design and development time are cut to a fraction of what they used to be." Just such a recursive, real-time development process is essential in a project that depends on creativity for success. Heyer proposes that the design process begin with a crude storyboard that may be done on paper or on the computer using rough video and low fidelity graphics. Team members can work on the rough storyboard that may be done on paper or on the computer using rough video and low fidelity graphics. Team members can work on the rough storyboard, which is really a rough prototype, and experiment with a wide range of design options and alternatives. As the prototype matures, "real video and graphics are substituted for storyboard images, the code is finalized, and the project is finally sent off for mastering." Heyer sees three major advantages in this approach. Risk is reduced because fundamental problems can be discovered early. Because design and programming occurs at the same time, there is less likelihood that the team will commit to a design that cannot be executed. A second advantage is the encouragement of experimentation and revision which will enhance the quality of the final product. Finally the process of development becomes understandable and "each person knows exactly what the others are doing."

In this project we developed a storyboard (prototype) in Authorware Professional using "placeholder" graphics, photographs, sound and instructional sequences. Then, as the project progressed, placeholder materials were replaced with what we thought were final components. This was accomplished in two phases. First, a fully functioning path was developed through the entire simulation. Then the appropriate graphics, sound, and images were created and added to the program. That path was then evaluated by literacy teachers, literacy students, and experts in relevant areas such as graphics, instructional design, and assessment. Finally, the pattern and format that emerged from work on the path was extended to the other paths through the program. We are currently in that phase of the project: "filling in" the remaining paths through the simulation using the format, graphics, and structure already developed and critiqued by teachers and students.

A design and development approach that is recursive and allows for changes and revisions in "real time" calls for a collaborative team with a wide range of skills and expertise. It also calls for a team that can tolerate almost continuous change, adjustment, and fine tuning as well as the frustration that is probably inevitable when some work is abandoned or replaced by new material. The major drawback to traditional linear models of instructional development may be that they eliminate many opportunities for fine tuning and artistic enhancement. The major drawback to recursive models may be that everything is so fluid that team members have difficulty seeing progress and feeling good about their accomplishments to date. If everything can be changed at any stage of development, nothing is ever truly "finished" until the end of the project. Many team members need the feeling of closure that comes from finishing different stages of a long, complex project and teams that use a recursive model need to attend to those needs. [At the end of a long hard work session the "simpled-out" silliness that often occurs probably serves to alleviate the frustrations that have developed as well as build a sense of camaraderie.] Recursive approaches, like linear models, can be taken to extremes that are both frustrating and nonproductive. Whistler, for example, had great difficulty deciding when a painting was truly "finished." On several occasions he actually went to the homes of patrons who had purchased a painting and made small changes to it!

Non-linear, recursive development models must also deal with the fact that time, a factor in all projects, is linear. Almost all development projects have tight timelines, and recursive approaches call for nimble but energetic project management, particularly when the work of many different team members feeds into, and is required for, the work of others. Common problems in projects that use a linear development model -- assignments that are not completed on time, work that does not "fit" with work done by other team members, concepts and design formats that are changed by one group or individual but not communicated to others, and the "just a few more changes and it will be ready" phenomenon - can happen more often in recursive models and multiply the possibilities for problems, delays, and conflicts. Effective time management, long and short range project planning, successful team building, and strong overall project management are critical when a non-linear, recursive approach is used.

REFERENCES

1. Thiagarajan, S., Semmel, D., and Semmel, M. (1974). *Instructional development for training teachers of exceptional children: A sourcebook.* Reston, VA: Council for Exceptional Children.
2. Dick, W., and Carey, L. (1985). *The systematic design of instruction (2nd ed.).* Glenview, IL: Scott, Foresman.
3. Ambron, S., and Hooper, K. (1990). *Learning With Interactive Muldimedia: Developing and Using Multimedia Tools in Education.* Redmond, WA: Microsoft Press.
4. Schon, D. (1987). *Educating the reflective practitioner.* San Francisco: Jossey Bass.
5. Holsinger, E. (1992). Nonlinear systems enhance video editing, *MacWeek*, 10, 26.
6. Heyer, M. (1987). The creative challenge of CD ROM. In S. Lambert and S. Ropiequet (Eds.) *CD ROM The new papyrus.* 347-358.