ED 393 410 IR 017 746

AUTHOR	Capell, Peter
TITLE	The Multimedia Factory.
PUB DATE	93
NOTE	7p.; In: Verbo-Visual Literacy: Understanding and Applying New Educational Communication Media Technologies. Selected Readings from the Symposium of the International Visual Literacy Association (Delphi, Greece, June 25-29, 1993); see IR 017 742.
PUB TYPE	Reports - Descriptive (141) -- Speeches/Conference Papers (150)
EDRS PRICE	MF01/PC01 Plus Postage.
DESCRIPTORS	*Computer Assisted Instruction; Computer Software Development; Educational Technology; *Instructional Design; Instructional Materials; Lesson Plans; *Multimedia Instruction
IDENTIFIERS	*Automated Tutoring; Multimedia Technology; *Piano Instruction

ABSTRACT

The history of instructional multimedia systems development includes many difficulties--authorial uncertainty about how to order concepts, the misplaced focus of so-called "intelligent tutors," the sometimes disconcerting proliferation of products, and financial constraints. Against this background, this paper discusses the project which created Piano Tutor, a multimedia workstation that teaches basic piano playing. In its teaching mode, it can guide students through a dynamically changing lesson path. In its practicing mode, it provides guidance and graphical representations of playing technique without interruptions. Piano Tutor's software building blocks are listed, along with processes and experiences that led to its automated curriculum analyzer, its accompanying videodisc, and multimedia event schedulers. It is hoped that with existing technologies and those debuting in Piano Tutor, multimedia can be created in systematic production-line style rather than ad hoc, "reinventing the wheel" each time. (BEW)

THE MULTIMEDIA FACTORY

by
Peter Capell

Abstract

Currently, multimedia systems suffer from a common problem: complexity. Nowadays these systems can incorporate a number of elements: intelligent diagnostics, computer graphics, digitized sound, digital or analog video, music, and so on. Most systems are created *ad hoc* such that the development method is discarded and forgotten soon after the system is created. However, with the use of existing technologies as well as those created in the development of the Piano Tutor, multimedia system development can be given a significant boost towards becoming a *production line* process.

Introduction

One problem in developing instructional multimedia systems is in structuring content. Most developers have the same types of questions: How do I order the ideas I want to teach? What would happen if I put *concept y* before *concept x?* Oftentimes a large curriculum can be daunting and the order and sequence of ideas is not obvious, or even linear--maybe the student should choose. The Piano Tutor project went down these paths and produced software to cope with the complexity of the tasks involved--from automated curriculum design assistance to computer-screen graphics development. The Piano Tutor's tasks were non-trivial because the curriculum consists of one hundred lessons which are executed according to the student's accomplishment of as many skills that are considered as prerequisites to and objectives of the set of lessons.

The *byproducts* of our development effort exist now as fully functional systems. It is my belief that the toolset is currently advanced enough to permit us to create the *multimedia factory*. We understand the elements required to engineer our systems, however, the process is still relatively undefined. We are on the brink of being able to systematically construct what Wenger (1987), described as knowledge communication systems.

Yes, we certainly have the constraints of limited resources and deadlines for projects to weigh against any heady development efforts, however I do not believe that it is resources and deadlines that are the actual constraints limiting the vision of multimedia systems. Instead, I choose to blame the process. Here is why:

In the 1980's there began in earnest the development of intelligent tutors. These systems could make simple deductions about the state of the learner and adjust teaching strategies or the curriculum, or both, to fit the system's best reasoning about the student. Unfortunately, the focus throughout the period, such as Meno Tutor, Buggy, Proust, and the Algebra Tutor (Sleeman & Brown 1982; Wenger, 1987), often seemed not so much upon the learner as much as the pompous question of "Where's the intelligence?" My view as an educator and non-computer scientist was something like: ". . .it's just a computer program. It's as intelligent as its programmer." Unfortunately, the question really was about proving something about *artificial intelligence* more than making excellent learning environments.

All of this is not to deni ite the fantastic achievements of those systems. John Anderson's LISP Tutor exceeds the efficacy of classroom instruction by a large margin (Anderson & Reiser, 1985). So far, the Piano Tutor has been able to move its limited sample of students through one year's worth of piano instruction (*one year's worth* as judged by two professors of music on the project) in less than a third of that time.

In addition to the success of these systems, there was the development of a new idea--using a computer as the modeling environment for specific learning paradigms and theoretical examination. Computers became laboratories for the evaluation and testing of ideas in cognitive science.

Beside these interesting and profound developments in the intelligent tutoring community, was the continuing practical application of computers and video to conduct training. The emergence of affordable computer video technology created a whole new paradigm in video and computer-based instruction. For people *in the trenches* doing *real* training and instruction, computer video provided an exciting new platform upon which to build simulations, surrogate travel, surrogate worlds, and so on. We saw the Aspen Project, and *Discovision* and we learned all about levels 0-4, struggling mightily to make sure we understood them. Specifically, I can remember holding to a mental model that level one was *less than* level four until I met Ed Schwartz at the University of Delaware and saw a beautifully level one application of videodisc in the music series developed there. There was a lot of excitement in this period where new doors were opening to machine-based education at unprecedented rates.

In fact the rate of *door opening* has not really settled down since, and this has turned out to be somewhat of a deficit. Endemic to the information age is the profusion of options. Now we have DVI, CD-I, videodisc, digitized voice, and MIDI. We have multiple possible implementation platforms with multiple configurations per platform. We have a variety of new and improved authoring tools as well. And while there is no question that the tools have improved, there is still a question of what these tools are building. Most often the systems being built are either *exploratory* or *prototypical*--intended either as research projects of proofs of *cutting edge* technologies. Either that, or we have a mix of applications that are not really stretching technology at all--point of purchase applications and buyer assistance guides that are little more than databases with fancy graphics. These systems are useful, however we need now to consider bringing what is now cutting edge into the world of the commonplace, and I believe we are currently on the verge of being able to do so.

There are many important obstacles to this notion having nothing to do with available technologies, not the least of which is money. The Piano Tutor cost $200,000 per year for three years, not including three years before the project was funded which were used for foundation research, planning, and the effort to obtain grant money. A bigger obstacle however is perception. I am hoping that we are moving towards a time when these environments will proliferate on their own, based on a common understanding of their value. In other words, it will be understood how to build them, and they will be built because their value is taken for granted.

In order for this wide ranging *understanding* to happen, the proper mix of elements must take place between the diagnostic power of the intelligent tutor in addition to multimedia and training-type systems. And their needs to be a relatively standard process for their development. There needs to be a multimedia factory, consisting of the tools and methods to build these complicated systems quickly with replicable success.

On the Piano Tutor Project, I believe we developed a few useful tools and made some discoveries that could prove useful in building the multimedia factory. Before I describe those, I will provide background on what the Piano Tutor is.

Piano Tutor

The Piano Tutor is a multimedia workstation that teaches basic piano playing. The system employs real-time music technology, expert tutoring, and videodisc. In developing the system, a curriculum analysis system was developed to model the curriculum in order to test the integrity of its instructional design.

The Piano Tutor operates in two modes; as a teacher and as a silent observer/evaluator during practice. In its Teaching mode, Piano Tutor guides students through a dynamically changing lesson path with appropriate interruptions for suggestions and corrections. In its Practicing mode, Piano Tutor provides guidance without interruptions, providing students with graphic evaluations of their playing, among other capabilities. Students may work with a teacher or by themselves, receiving expert piano instruction at a pace set by their own ability.

Piano is composed of a computer connected to a velocity sensitive, piano (MIDI) keyboard, MIDI synthesizer, and videodisc controller. The computer serves as the controller for the workstation and as a music score display. The videodisc is used to provide detailed visual and verbal instructions, the keyboard is the student's input device, and the synthesizer allows the student to hear what he or she has played.

In order to evaluate the student's performance, turn music score pages at the appropriate times, and accompany a student's playing, the system recognizes and evaluates keyboard input. The MIDI interface reports the notes played, but the heart of the system is the pattern-matcher. The matcher allows the system to identify where the student *is* in a score, even though he may be playing at varying rates of speed and playing many wrong notes. The information gathered by the pattern-matcher is used to evaluate the student's performance, accompany the student, and coordinate the display of the score (*turn pages* on the computer screen).

Although the practicing and teaching systems are separate components, each has similar software building blocks:

- pattern-matcher--matches music performance with music score in real time,
- digital recorder--records keyboard performances in digital form,
- playback--plays recorded keyboard performances,
- performance evaluator--conducts evaluation of student performance,
- score display systems--handles all score displays,
- intelligent decision maker--determines the appropriate lessons to execute relative to immediate student input from the keyboard and student performance history,
- videodisc controller--handles video-disc interactions,
- accompanist--provides accompaniment for any prerecorded score.

The system provides direct instruction in its teaching-mode and passive criticism in the practicing-mode. In the teaching mode, Piano Tutor actively teaches new concepts, evaluates the student's performance, and directs the student's progress. The practicing system provides passive visual criticism to guide the student in evaluating and correcting errors without interruption during play.

In teaching mode, Piano Tutor instructs beginning piano students in basic performance skills. The current curriculum provides approximately one year's worth of piano instruction (Dannenberg, Sanchez, Joseph, Saul, Joseph, & Capell, 1990).

The Curriculum Analyzer

Instructional design for an automated, non-sequential lesson planner proved to be a solvable but non-trivial exercise (Capell & Dannenberg, in press). Piano Tutor's strategy in coping with this problem comes directly from the most basic principles of instructional systems design (ISD), (Dick & Carey, 1985). Two of the most basic ideas of ISD are: (a) Students are evaluated according to observable skills, and (b) students must fulfill the requirements of less complicated skills before moving to a new skill. This means that any lesson taught by the Piano Tutor has prerequisite skills and an objective.

Observable skills are distinguished carefully in ISD to avoid the uncertainties of ad hoc curriculum writers who specify their objectives with comments such as: "The student will understand how to read at the first grade level." The ISK model is an inherent challenge to these kinds of vague goal statements requiring detailed information such as: "What does 'understand how to read at the first grade level mean?'" The instructional systems design expert will always want to see an operational definition for any goal statement. In our example, the ISD specialist will ask: "What must the student *do* in order to demonstrate the ability to read at the first-grade level?" Will the student have to read "War and Peace"? Or will she only be required to read a few sentences? ISD is all about specifying and verifying learner behaviors as evidence of their comprehension, and then adapting instruction to adjust to weaknesses.

In one sense, it is easier to implement a model with ISD's rigor in a computer system because people rarely have the fortitude to be as exacting as they need to be. One instructional breakthrough for us was the development of an analysis model that helped us to determine whether its prerequisite skills were always met in a network of one hundred lessons. The problem is that Piano Tutor scores skills, not lessons. So by obtaining certain skills, one gains access to new lessons. It is not easy to figure out in one's head however, whether or not there may be *holes* in the curriculum, where for example a lesson teaches a skill that is not used by any other lesson in the curriculum, or where a lesson is *cutoff* from the rest of the curriculum because its prerequisites can never be met.

The curriculum analyzer turned out to be central to the development effort, permitting us to create lessons and skills, simulate them, and determine whether our curriculum was complete unto itself. It also showed the utility of the systems approach to instructional design in a very formalized way with practical results. Further, any system of like complexity will have to cope with this problem. We believe that our method is sound, complete, and replicable for systems of high complexity with large number of lessons.

Video

The next most difficult problem in my area of the Tutor was the production of a videodisc to accompany the 100-lesson set. We wanted only one videodisc, so there would be no need to change discs in the middle of curriculum. Since the system would be executing lessons with many possible orderings, the student could potentially wind up changing discs again and again as the system would require video stored on either of the two discs. And so our effort began to try to put as much as possible onto only one disc. Hopefully this need will soon be oviated forever with DVI, CDI, or some other variant, but in our experience coping with this issue we discovered some useful videodisc production techniques that will have implications no matter what the video storage medium.

We learned to script our lessons individually and to time them. We had the scripts in electronic form, and so these were entered into a database. Perhaps the most time-intensive aspect of video shooting is changing camera angles. Having each script in the database, we encoded each paragraph with a marker as to its angle. In other words, some paragraphs would be marked *overhead*, others *left side*, and so on. This enabled us to run simple queries to the database to extract all the scripts with a side angle or all the scripts with an overhead shot for example, enabling us to do the action and reading for each *script cluster* leaving the camera in position. This saved a lot of time. Given the modular nature of the system and the sheer number of lessons, had we shot in script-by-script fashion moving the camera in order accordingly, we would easily have spent weeks accomplishing what took only three days otherwise.

Additionally, digitized voice saved us enormous amounts of videodisc real-estate because it permitted us to repurpose portions of the disc just by silencing the audio portion of the disc and playing the digitized voice synchronized to the same video as needed. Without these efficiencies and methods, our effort would certainly have been lost.

Mixing Multimedia Elements

Although there are now many multimedia event schedulers on the market, this was not so when we were developing the tutor. A member of our team wrote a multimedia timeline program that permitted us to integrate video, digitized audio, MIDI, and screen graphics. The system provides the user with a timeline onto which events are placed, so that at time zero, a video presentation would begin, then a graphic on the screen appears with accompanying voice over, and so on. Thank goodness, these systems are now widely available as commercial products such as Authorware™.

So What?

This is the question that researchers in any field should be asking at every moment. The importance of any of the elements described above is not contained in any one of them, but in all of them as a system. Currently there is an explosion going on in multimedia, especially in the use of digital video. Systems, such as "From Alice to Ocean" (Dannenberg, et al., 1990), provide users with beautiful graphics and an interesting interface to navigate, but we need not stop here. In the future these beautifully crafted systems will incorporate user diagnosis and intelligent feedback. We now have the elements in place to bring about what the early attempts at programmed instruction could not--a truly gripping learning experience using the best of modern technology.

The Multimedia Factory

And so what is the multimedia factory? A factory is a cohesive set of operations that are dedicated to the creation and assembly of a finished product. In other words, lots of efficient operations cooperating to produce some sort of widget.

Why a factory? As multimedia aficionados, we all want these efforts to succeed. It is an exciting technology that we know in our hearts is going to sweep us into a future of wealth and happiness--or at least we hope. Our big obstacles are creation and development costs--costs in resources, time, and money. My argument is that we are still spending far too much time on the inefficiencies of learning curve, not because of our misunderstanding of new technology, nor because we are valiant explorers, but because of inefficiencies of process.

On one hand we could say we need a factory to keep up with the tide of events. All of us have seen technologies that have seemed to take inordinate amounts of time to develop--consider how long videodisc has been around. And so we need the factory because of the obstacles to progress that still haunt our efforts--cost, time, and complexity. We need the factory to help us to quickly create modules that can be assembled into larger components.

If systems go the route of greater complexity as I predict, we will need the factory if only to keep up with the tide. But if things do not take off as I expect, and we want to *create* the future, we will need to have the factory to convince our funding sources that we can produce systems with broad functionality for relatively low cost.

Conclusion

The reason for my excitement at this moment in time is that I believe the world of multimedia and knowledge systems of this kind are at a point where they will be *manufactured* as products instead of being explorations in and of themselves for academic ends. There really has been a lot discovered in the development of these video and computer-based interactive systems over the past ten years, and I believe it is now time for advances in their creation to take hold.

And so the questions that will press the multimedia community in coming years will be the issues of defining process. How are these systems created? Which parts are gruesomely time consuming? How can we automate or standardize these? What will be our general development strategy?

As a developer in this field, I sincerely hope that we will be able to capture our lessons learned so that the process of multimedia development will become standardized enough that we can consider more radical extensions including advanced intelligence instructional systems, and so on. I would like to see stunning interactive systems with the ability to provide instruction in complex domains so that students of the next generation cannot pull themselves away. We are now at the point of departure.

References

Addison-Wesley (1992, July). *From Alice to ocean.* (Based on *Tracks* Davidson, R. Pantheon Books).

Anderson, J. R., & Reiser, B. J. (1985). The lisp tutor. *BYTE*, 159-175.

Capell, P., & Dannenberg, R. (in press). Instructional design and intelligent tutoring: Theory and the precision of design. *Journal of Artificial Intelligence in Education*, *4*(1).

Dannenberg, R., Sanchez, M., Joseph, A., Saul, R., Joseph, R., & Capell, P. (1990). An expert system for teaching piano to novices. *ICMC Glasgow Proceedings* (pp. 20-23). San Francisco, CA.

Dick, W., & Carey, L. (1985). *The systematic design of instruction.* London: Scott, Foresman.

Sleeman, D., & Brown, J. S. (Eds.). (1982). *Intelligent tutoring systems.* New York: Academic Press.

Wenger, E., (1987). *Artificial intelligence and tutoring systems.* Mountain View, CA: Morgan & Kaufman.