Rule Base and Digital Video Technologies Applied to Training Simulations

Advanced Learning Technologies Project

Michael G. Christel
Scott M. Stevens

Abstract

The Advanced Learning Technologies Project developed a digital video course on code inspections from 1987 to 1990. The essence of this course is an environment in which a student participates in a code inspection as a contributing reviewer of the code.

The student chooses an inspection role, and later assumes all the responsibilities of that role while performing in a code inspection simulation. The student is an active participant in the code inspection, and his or her contributions affect the course of the inspection dialogue and ultimately the success of the inspection. In addition, the role the student takes in the inspection is not predetermined but is selected by the student. To provide this flexibility, a rule base is used to control the code inspection simulation.

To participate effectively in the code inspection, the student needs to recognize and react to the other reviewers' comments and their emotional states. The importance of group process issues necessitates that the inspection simulation be presented as realistically as possible while still preserving the flexibility of dynamic role selection and active participation. The code inspection course makes use of digital video for dynamic scene creation in addressing this requirement.

These techniques are applicable beyond the code inspection course to other instructional simulations. The synergistic effects of using digital multimedia to improve visual fidelity along with rule bases for behavioral modeling and dynamic scene creation can significantly improve the utility of low-cost simulators and part-task trainers.

Software Engineering Institute
Carnegie Mellon University
Pittsburgh, PA 15213

1 INTRODUCTION

Computer-based simulations are increasingly being recognized as a valuable approach to education and training. Congressional testimony has stressed the need for expanded use of simulation as a way to reduce costs and to improve large-scale military training [USSCAS90]. The establishment of the Defense Modeling and Simulation Office in 1991 reflects the enhanced attention given to simulation by the Department of Defense. Computer simulations are gaining popularity for instructional use in several diverse areas, from military training [Saunders89, White90] to science instruction [Weir90, StanSmith86] to management education [Muller-Malek88, Hassett86].

A simulation is an imitation or artificial duplication of a process or operation. Depending on the simulation, various levels of fidelity may be attained. In [McBride90], current families of simulators are classified into two basic categories; this paper shall retain that classification. The first category consists of high-fidelity, high-cost simulations represented by the wide variety of full motion flight trainers and shipboard system trainers. These simulators often employ elaborate equipment mockups to support a high degree of instrument and control functionality, along with sophisticated computer-generated imagery. It is noted in [McBride90] that because of their expense and complexity, these simulators are usually located in fixed facilities, and they involve considerable logistics overhead to ensure maximum effective use by the training population. They also require a large amount of technical and administrative support to keep them running.

The second simulator category includes part-task trainers and systems that are usually more limited in scope and capability. This paper will concentrate on this category, and will refer to such systems as low-end simulators. This category is represented by maintenance and procedural trainers. These systems in the past have been characterized by hardware mockups with a mix of operative and inoperative dials and controls, along with low-resolution display devices. As [McBride90] notes, simulators in this category now have a wide range of fidelity and cost, and innovative adaptations of commercial arcade games have appeared recently in trainers for aircraft recognition, tank gunnery, and rifle marksmanship.

Low-end simulators offer many advantages over the actual process or operation occurring in the real world. They offer a unique opportu nity to present consistent and replicable situations. Events that may occur only rarely in reality can be presented at any time in a simulation. A simulation can also present a wide variety of experiences within relatively short practice periods. Events with hazardous or costly consequences in reality can be presented in the simulation without such adverse effects. A simulation is less expensive to use than the actual equipment. It is also potentially transportable, allowing relocation of the simulator to take advantage of student demographics. These and other advantages, which are presented in [Smith86] with references, have led to the increasing use of such simulations for education and training.

Emerging computer technologies yield dramatic new opportunities for simulation. Two such technologies are the use of multimedia, specifically digital audio and video, and the use of rule bases for behavioral modeling and the dynamic presentation of multimedia information. Advantages that such technologies offer to low-end simulators include higher output fidelity at low cost, and better simulation fidelity through dynamic interactions rather than preset scenarios.

The Advanced Learning Technologies (ALT) Project at the SEI defined a new paradigm for the use of such technologies in imple menting a simulation of a group process activity. This paradigm will be presented in the next section. The code inspection course created by the ALT Project will then be discussed, as the inspection simulation within this course illustrates an application of this paradigm.

This paradigm has implications beyond group process simulations for instructional purposes. Low-end simulators have frequently used interactive analog videodisc systems (for example, an F16 maintenance trainer [Saunders89] and a marksmanship trainer [White90]). The use of digital multimedia as prescribed by the paradigm can lead to significant improvements in such interactive video-based simulators.

Likewise, the paradigm can lead to improvements in the design of embedded training, a critical area considering that as software intensive systems become more complex, the need for better instruction, in the operation, maintenance, diagnostics, and repair of such systems increases. The use of multimedia data complemented by intelligent rule base support allows the training to be integrated into the actual system. Section 4 of this paper discusses in further detail these and other points concerning the implications of the rule base and multimedia paradigm.

2
SIMULATION SUPPORT VIA DIGITAL MULTIMEDIA AND RULE BASES

The Advanced Learning Technologies Project began in 1987 with the following goals:

Demonstrate the applicability of technologically advanced instructional media to software engineering problems.

Promulgate effective software engineering methods by providing a technology-based course.

Raise the awareness of educators and trainers concerning technology-intensive software engineering education.

In support of these goals, the project developed a code inspection course using interactive digital video; this course is the subject of Section 3.1. The basis for this course is a code inspection simulation. With such a simulation, the student using the course can gain practical experience in group dynamics and inspection techniques while receiving the advantages of low-end simulators presented earlier. Without this simulation, the course would have offered little benefit over traditional educational media. Thus the simulation played a central role in the project, and one of the first project activities was the evolution of a paradigm for developing group process simulations.

The paradigm integrates four related principles often encountered in the instructional simulation literature:

- enhancing aural and visual output fidelity
- using fine-grained output components for improved modeling
- enriching simulation fidelity through behavioral modeling
- enriching the learning experience through intelligent tracking of student progress and control over data presentation

These four facets of the paradigm, when considered individually, have each received past attention with respect to low-end simulation. For example, videodisc courses composed of several multiple minute sequences have long been criticized as being "interrupted video," containing little simulation capabilities because the grain size of the output is too large. However, the four facets taken together have a synergistic effect on the utility of low-end simulations. The importance of this paradigm is not so much in identifying the various approaches to instructional simulation as it is in defining how they can work in concert with each other. This section discusses the different facets of the paradigm, followed by Section 3, which discusses the paradigm's application to the code inspection course.

2.1 Output Fidelity Issues

For a simulation to communicate its operations or results, it must address output mechanisms. In addition, the simulation program must have access to input channels to initiate action based on the desires of an external user or group of users. Further, if the simulation is interactive, it must have access to input and output during the course of execution so that the user community can dynamically change parameters and see the effects of those changes. Ideally, a training simulator will be able to provide exactly the same look and feel as the real-world experience. Thus a simulation, especially an interactive simulation, has to take into account input and output concerns.

When simulating a group process such as a negotiation or a technical review, output fidelity is very important. Information from members in the group is conveyed not only through the content of spoken messages, but also through the volume and tone used in such messages as well as various nonverbal cues. For example, a group member slumped in a chair with eyes looking downward conveys boredom, while another person waving fists in the air indicates aggression. Thus, for group process instruction, the representative capabilities of the audio and video output can contribute significantly to the simulation fidelity.

This hypothesis has been addressed in part in past interface research. For example, one study found that the use of natural human speech rather than computer synthesized speech promotes better recall, comprehension, and satisfaction by the user [Waterworth85]. It is likely that digitized human speech retains such benefits over synthesized speech. Evidence on the effectiveness of simulations incorporating video can be found throughout the literature on multimedia instruction (for example, [Bosco86, Kenworthy87]). Through a meta-analysis of 47 studies comparing interactive videodisc instruction to conventional instruction involving military training, higher education, or industrial training, Fletcher concludes that interactive videodisc instruction is more effective (and also less costly) than conventional instruction [Fletcher90]. However, these examinations of video in instructional simulations fail to prove whether it is the increased output fidelity provided by the video that leads to the instructional effectiveness [Fletcher90, page S-3]:

> There was little in the reviewed studies to indicate how interactive videodisc instruction achieves its success. The studies examined in this review did little to indicate which features of interactive videodisc instruction contribute to the observed increases in student achievement. Additionally, there are many outcomes-such as speed of response, accuracy of response, attitude toward the subject matter, insight, transfer, and retention-that may alone or in some combination become the objectives of instruction. How different designs for interactive videodisc instruction contribute to accomplishing these different outcomes was rarely addressed by the reviewed studies and remains a proper topic for future research.

To directly address the role of visual fidelity in instructional simulations, a formal experiment was conducted on a group process simulation. The simulation used for this research was the code inspection course described in Section 3.1. The experiment measured the effects of changing output characteristics of the simulation on recall performance and opinions toward code inspections. Complete details of this research are presented in [Christe191]; the experiment will only be briefly summarized here. Results are reported at the 0.05 level of significance.

The experiment used 64 college juniors, seniors, and graduate students as subjects. One group of subjects was presented some information from the code inspection course as motion video with audio. The remainder received this information as a "slide show" of images, still synchronized to the audio but appearing at no faster than one video image every four seconds.

The images used for the no motion video treatment group formed a subset of the images presented to the motion video treatment group. For example, in instances where the data to be displayed was stored as full motion, 30-images-per-second video, the no motion video treatment group would only be shown every 128th image, with each image being displayed for slightly over four seconds until the next image in the sequence was ready to be displayed. The motion video treatment group would view every image of that data, with each image being displayed for one-thirtieth of a second. Thus, for a 20-second video, the no motion video treatment group would see five images, each displayed for approximately four seconds, which would be perceived as a slide show of still images with accompanying synchronized audio. The motion video treatment group would see 600 images, each displayed for one-thirtieth of a second, perceiving this as motion video with synchronized audio.

The audio was exactly the same for both treatment groups. To visually distinguish the two treatments, consider the scene shown in Figure 1. One group was presented only still sequence animations of the three pictured personae. The other group saw motion video presentations of these personae.

The subjects receiving the motion video interfaces remembered significantly more material concerning the personae's code inspec tion discourses than the subjects receiving only the stills and audio. In addition, subjects receiving motion video significantly shifted their views of code inspection concepts toward being more active than did the subjects receiving no motion video.

Two methods of navigating through the code inspection world were also tested, one a direct manipulation point-and-click map, and the other a surrogate travel interface where the subject "walks" through the space and into the desired room. These are contrasted in Figure 2 and Figure 3. Half of the subjects used the point-and-click map, and half used the surrogate travel interface.

Both groups liked, or disliked, the interfaces equally and used them in equivalent fashions to navigate the world. However, sub jects with the surrogate travel interface came away from the experience with significantly more positive opinions about code inspections. While the surrogate travel interface was more cumbersome and slower, and it clearly has nothing directly to do with inspections, its subjects were more completely brought into the "virtual world" of the simulation. That is, the overall simulation fidelity was increased.

Figure 1
Example of Video Output for Code Inspection Simulation

Figure 2
Floor Plan Image for Graphic Menu Navigation

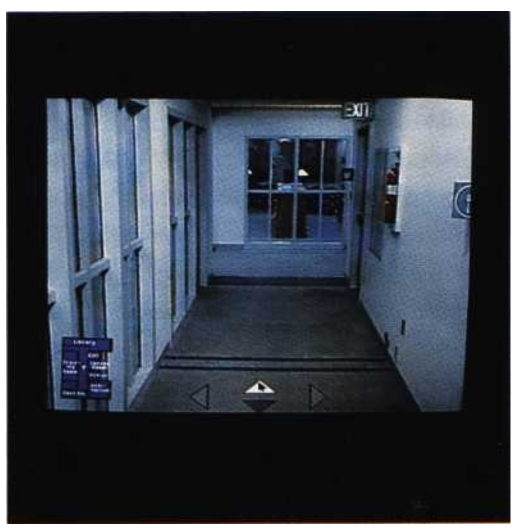


Figure 3
One Still Image from Surrogate Travel Navigation

The instructional objectives for the code inspection course included recognizing and remembering the contributions and emotional states of the personae, and appreciating the group dynamics involved with code inspections. Thus recall performance and opinions toward code inspections were both valid measures for determining instructional effectiveness. This research provides evidence that for the code inspection simulation, the increased fidelity provided by the motion video and a more realistic virtual world (surrogate travel interface) resulted in an improved course.

2.2 Granularity Issues

The first facet of the paradigm establishes the importance of output fidelity for instructional simulations of group process. One suggested means of improving fidelity is the use of video and digitized speech, as opposed to using graphical animations or cartoon images and synthesized audio. The interactivity of the student with the group process, and hence the simulation fidelity, can be further enhanced if such output data is organized into reconfigurable elements rather than monolithic chunks. Such reconfiguration is best supported by digital video, and the organization is improved through the use of descriptive header information.

2.2.1 Advantages of Dynamic Presentation

Consider a hypothetical simulation of a conflict between two parties, used to teach the student mediating skills, where the output data consists of several discourses. A discourse in this context is a single person speaking without interruption on one or more topics. If the output data consists of only two discourses, one for each party, there is very little feedback that can be given to the student concerning the student's mediating actions. The student can see the two discourses played in sequence, or may see only one. This results in only four permutations if at least one discourse is played, as shown in Figure 4. If, on the other hand, each of these two discourses is broken up into smaller, meaningful pieces, then there are numerous possibilities to alter the information presented by the simulation according to the student's mediating actions. The discourses might be partitioned into expressions of emotion, a position statement, several arguments supporting that position, and a conclusion. Depending on the student's effectiveness as a mediator, emotional outbursts can be controlled, tangential points suppressed, positions stated first by both parties before arguments are presented, and so on. The finer output elements provides the student with greater interactivity in learning about mediation.

The benefits of such interactivity have been reported in multimedia educational research [Bosco86, Kenworthy87, Fletcher90], although, as mentioned before, these studies fall short in proving the exact causes behind such ascribed benefits. The authors believe that the use of dynamically reconfigurable audio and video components enhances the output fidelity of a group process simulation, which in turn leads to better achievement of the simulation's instructional objectives.

The manner in which the video is stored takes on added importance when considering the effect of output data granularity. In the past, motion video presentation capabilities were added to a microcomputer through the use of an analog videodisc player. In recent years, however, digital video computer systems such as Intel's Digital Video Interactive (DVI®), Philips' and Sony's Compact Disc-Interactive (CD-I), and Commodore's Dynamic Total Vision (CDTV) have been introduced [Robinson90]. These systems offer the potential for greater flexibility in that digital video and audio can be manipulated like other computer data rather than having to be treated as a distinct analog medium. To maximize reconfigurability, the audio and video should be stored in digital form. This prescription, along with a more detailed comparison of the capabilities of analog video and digital video, will be discussed in the following paragraphs. This material provides insight into how analog video was used in low-end simulators in the past, and it prepares the reader for the constraints that were placed on the implementation of the digital video simulation of a code inspection.

Figure 4
Effects of Granularity on Conflict Simulation

2.2.2 The Capabilities of Analog and Digital Video Systems

The incorporation of video information into computer simulations in the past has been performed through the use of a peripheral video disc player, which allows the accurate, quick retrieval of analog video images. In the constant angular velocity NTSC format, a laser videodisc can hold 54,000 single frames of video, which is equivalent to 30 minutes of motion video. There are also from 2 to 4 audio tracks of 30 minutes each. Unlike videotape, the constant angular velocity videodisc is a random access device, and any video frame from this set of 54,000 can be displayed within approximately one second. The high-quality video and the large storage capacity of the videodisc is complemented by the powerful capabilities of the controlling computer, which makes available rich visuals for computer applications. However, such a platform is of limited usefulness for simulations because a videodisc image cannot be changed without additional hardware options and even then only in a limited fashion.

Group process simulations should be able to piece together audio and video segments during program execution much like text pro cessors can piece together text at runtime. The ability to create scenes dynamically in supporting such instructional simulations is predicated upon the accessibility of video information down to the pixel level, just as text processors can access information down to the individual character level. An advantage that digital video systems offers to simulations over videodisc is that the atomic unit becomes the bit rather than the frame, allowing these bits to be manipulated by the host. The biggest disadvantage with digital video is its size (for example, a single high-quality video channel would require 200 million bits per second [Watkinson90]), so commercial digital video delivery systems necessarily incorporate compression techniques.

An example will help clarify the added flexibility provided by digital video. Consider the image of Figure 1 (on page 9) and the requirement that each of the 3 pictured reviewers must be able to look in 5 directions: right, left, down, straight ahead, and off in the distance. Consider the additional requirement that each gaze is to be accompanied by either a very passive, emotionally neutral, or very aggressive countenance. Thus, there are 15 looks for each of the 3 reviewers. Finally, consider that these 3 reviewers to be displayed are chosen from one of 3 sets made up from a domain of 4 reviewers: moderator, producer, and recorder; reader, producer, and recorder; or reader, producer, and moderator.

With an analog system, the scenes cannot be pieced together; so to have the 3 reviewers looking straight ahead and all being emotionally neutral, we need to utilize a videodisc frame. To have the producer being very aggressive but everyone else the same and all looking forward, we need another videodisc frame. For just the 15 looks described here, we would need 15 times 15 times 15, or 1475 frames for the "look" combinations, times 3 for the 3 combinations of reviewers. Thus, assuming we wanted only one set of still images of the reviewers in these poses of emotion and look direction, we would need 4425 stills. Should we want a half second of motion for each look combination, we would exhaust the storage capability of an analog videodisc.

With a digital video system like DVI, each reviewer image can be stored separately. Each reviewer takes up about one-sixth of the screen display as shown in Figure 1, so storing the reviewer image takes roughly one-sixth the space of a full screen image. To store the 15 looks described above requires 15 reviewer images. To create any combination of looks on the screen merely involves piecing together 3 reviewers and then displaying them. Thus, to create any combination of these looks for the conference room visual, we only need 15 images per reviewer, or 60 images.

Digital video offers other unique presentation capabilities not possible with strictly analog video systems. For example, DVI technology can present a seamless 360 degree view. When the user wishes to see this full circle panorama, DVI technology decompresses the rectangular patches of video that constitute the full view and holds them in memory. Depending on the direction the user indicates, the system digitally combines parts of two adjacent frames to make a display frame. No matter which direction the user selects, a smooth view is always presented. Other unique DVI capabilities include warping video images over wireframe models. For example, a DVI flight simulator was developed that textures realistic images onto wireframe models of buildings and terrain. The armchair user maneuvers a World War II Spitfire airplane through this simulated countryside, complete with engine sounds from a real Spitfire [Ripley89]. A final interesting digital video capability concerns saving images in a very high resolution form which permits very detailed zooms of sections of the image. The user can display the complete image, or selectively choose a section of the image to display at full screen size. As an example, a battlefield scene can be viewed on the display, or a tank from that battlefield can be zoomed to completely fill the display screen, simulating the effects of binoculars.

2.2.3 Multimedia Building Block Organization

With the addition of manipulable digital audio and video, a training simulation can have enhanced fidelity which in turn can lead to better achievement of the simulation's instructional objectives. However, reducing the size of the audio and video data elements increases the need for data organization. Choosing indexing schemes that adequately describe the audio and video data, along with choosing an appropriate level of granularity, allows this data to be used as building blocks for increasing the output fidelity of instructional simulations.

For example, it is possible to access digital video down to the bit level. However, trying to find a specific bit in a video scene (for example, the display of the state of a piece of equipment attached to a jet on a runway) might be of too fine granularity for a particular application. For a course on equipment troubleshooting, this information might be relevant. For a course on target identification, information about the particular state of a piece of equipment is probably unimportant. Even if the level of granularity is appropriate, the video information is unlikely to be retrievable without the addition of indexing schemes, such as header information identifying which still image in a motion video segment contains displays of that piece of equipment and the portion of that still that shows the state information. While text processors take advantage of the ease with which text can be searched, instructional simulations cannot search through large blocks of video without additional information to aid in the search. Organization and classification of the audio and video data in meaningful ways for the simulation will improve the access and presentation of such data.

The descriptions of a multimedia information database, being closely tied with the information, should logically remain with the information data rather than be implemented in an application working on the information. By following this recommendation, the database can then be more easily reused by other applications, since the descriptions will be in a reusable form tied with the data and not the application.

Capitalizing on the redundancy in the information is one way of improving information access while also optimizing storage requirements. A cluster of related information, such as a motion video sequence, can have its common attributes described in a *scene header.* Individual components in that information, such as still frames within the motion video sequence, can have their unique characteristics described in a *frame header.* The scene header and frame header are stored with their associated data. This is illustrated in Figure 5.

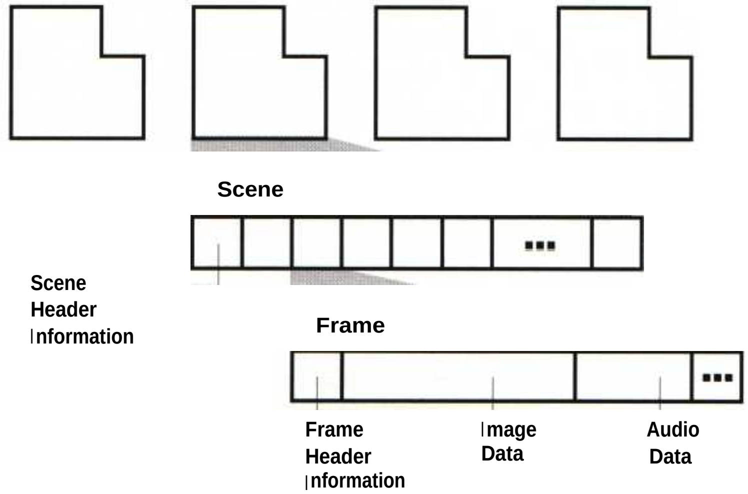


Figure 5
Multimedia Header Information

The scene header consists of traditional file control block information such as the name of the scene, the size (in bytes) of the scene, and the creation date and last modification date of the scene. The type of media is stored in the scene header (for example, audio, or video with synchronized audio). Media-specific descriptors can be stored in the scene header as well; for example, for an audio scene, the sample rate, filtering, attenuation, and compression algorithm can be identified. For a video scene, the scene header might include the frame rate, position, hue, saturation, and compression algorithm.

In addition to physical representations of the data, the scene header can be extended to include abstract representations of the data. For example, a scene header for a motion video sequence might include the actors or key objects used in the sequence and the setting of the sequence (for instance, an F-16 shown on a particular runway).

The frame header differentiates important information contained in the frames of a scene. For some information, this might include absolute three-dimensional location, field of view, and depth of field information. An example in which this information would be useful would be a set of images constituting an aerial image of a city. Abstract representations of the frame can be included along with such objective information. Using the same example, abstract descriptions might include the functional descriptions of buildings and the streets shown in a particular frame.

The use of such scene headers and frame headers can simplify the task of locating information in a fine-grained multimedia database supporting an instructional simulation. The fidelity of the simulation depends on both the material in the information database and the ease with which it can be dynamically combined into a presentation. Material presented at a constant rate such as motion video and audio can form realistic interfaces. In the past, video especially could not be easily modified dynamically for use as a building block in constructing images for simulations. The new capabilities provided by digital video allow motion video to be used in this way.

2.3 Behavioral Modeling Issues

Thus far, the facets of the paradigm for creating group process simulations have discussed only the importance of output fidelity and its achievement through the use of fine-grained digital audio and video components organized via header information. Any improvements in output fidelity will have only limited benefits unless the simulation fidelity is sufficiently rich to be able to fully utilize that output fidelity. This section will discuss how behavioral modeling can be used to improve the fidelity of a group process simulation.

Note that the paradigm developed by the Advanced Learning Technologies Project does not encompass every aspect of developing group process simulations. This paradigm does not try to identify and incorporate all the ways in which the fidelity of a group process simulation can be improved, but rather focuses on behavioral modeling and other techniques used in intelligent tutoring systems.

The simulation of group process necessarily involves the simulation of the interrelationships between entities of the group. The approach prescribed here for group process simulation is to model all of the entities (personae) involved in the group, even those whose roles will be assumed by the student. This permits the student to take any role desired with the system simulating the others. The same simulation can then be used to give the student realistic experiences in each of the different roles in the group process.

The behavior of entities involved in group dynamics is likely to be non-algorithmic. As such, the behavioral modeling is best achieved through using a rule base. The rule base models generic attributes such as defensive and aggressive behavior as well as how much the attributes change depending on the current and historical events. All entities modeled contain values for relevant attributes, which define the entity's "personality." The rule base determines what actions the entities take based on their personality, their role in the group, rules governing their behavior, current interactions from either the student or other entities, and a global history of interactions.

This behavioral modeling approach is insensitive to the number of entities modeled, how many students interact with the system, and which roles the students take. In this approach it is possible for the system to simulate all entities with the student taking a passive role, watching the simulation unfold. It is just as easy for more than one student to take different roles and interact with the simulation. One advantage of this approach is that it overcomes the "missing man" problem associated with simulators involving two or more people. If only one person is present, then such simulators cannot be run. If, however, the other participants in the simulated experience are modeled by the system, then a single student can interact with the system whenever convenient without having to invest additional people's time.

2.4 Student Feedback Issues

Along with using the rule base for modeling the behavior of the entities in the group, the rule base can also be used to support the tracking of student progress during the group process simulation. Traditionally, the assessment of student performance using simulators has been the task of the training staff. In many situations, a lack of sufficient staff to monitor the students or inconsistencies between instructors results in non-standard assessment techniques, inappropriate remediation, or lack of any tutoring at all [Maguire90]. (Remediation is corrective guidance and instruction in response to student shortcomings.) The use of the rule base for tracking student progress is a consistent, cost-effective, and convenient alternative to requiring instructor monitoring.

The rule base should keep track of situations that develop during the group process simulation in which the student performs ei ther poorly or admirably. In addition, the rule base should keep track of overall performance by the student. For example, the student might consistently fail to address a certain need within the group such as prompting a withdrawn group member to contribute. Such tendencies should be noted.

Unlike many intelligent tutoring system strategies that interrupt the student's work to present remediation and advice, the prescrip tion here is for the positive and negative feedback on student performance to be given following the completion of the group process simulation. A more invasive technique in which feedback is interspersed with the simulation would not only decrease the fidelity of the simulation, but also destroy the student's experience of participating in a dynamic group process. In real-life meetings, for example, an expert tutor does not interrupt the meeting to tell you how you are doing. Likewise, in a simulation of such a meeting, the remediation and advice should be formulated based on the student's performance during the meeting, but be given following the end of the meeting simulation.

This does not imply that there should be no feedback given to the student during their interaction with a simulation. Rather, that feedback given during the course of the group process should be natural feedback in agreement with what would be the natural consequences of the same action in a real experience. For example, if the student neglects to motivate a withdrawn group member into contributing, a traditional intelligent tutoring approach would notify the student via a computer-initiated message that some intervention should be done on his or her part to motivate the withdrawn group member. Regardless of whether this message is text, audio, or video, it is unlikely that this parallels what would occur in the real world given the same scenario. Instead, more natural feedback would be for the simulation to visually depict that the group member is not involved, for example by showing that group member looking out the window, and for the modeled behavior of that persona to become less attentive to the current events.

3

CODE INSPECTION COURSE

The multifaceted paradigm discussed in the previous section was used in the design and development of a digital video course on code inspections from 1987 to 1990. The concepts of behaviorally modeled simulation and dynamic scene creation were first tested through this course. Following an overview of the course, subsequent sections will focus on these concepts by further detailing the course's implementation.

3.1 Course Overview

The code inspection course runs on an Intel Pro750™ 80386-based computer incorporating a Digital Video Interactive (DVI) board set. This machine also has as a peripheral a standard compact disc read only memory (CD-ROM) drive. Most of the digital data for the course is stored on a CD-ROM, with the remainder of the data residing on the Pro750's hard disk. The course contains 450 video objects distributed between 20 minutes of full screen, full motion (30 frames per second) video with audio, 100 minutes of quarter- and sixth-screen video (at either 30 or 15 frames per second, with both perceived by the student as full motion) with audio, 10 hours of AM-quality audio decomposed into 4500 files, 2000 still images, and 5 megabytes of computer data.

The content area for the course is software code inspection. Inspections are a formal review process defined first at IBM for the purpose of finding defects in code [Fagan76]. They have been used effectively to reduce the cost of software development and to improve software quality in numerous institutions, including AT&T and IBM [Ackerman89, Fowler86, Fagan76]. A discussion of inspections in contrast with other software review practices can be found in [IEEE89]. An endorsement for the use of inspections along with implementation advice is provided in [Humphrey89].

Code inspections typically are one to three hours long, and are conducted by reviewers who have well-defined roles and responsibilities. An inspection usually has three to five reviewers, and at least the following roles are present (a person may be responsible for more than one role) [Fagan76]:

a moderator, who is responsible for verifying entry and exit criteria for the code inspection, ensuring that everyone is prepared and contributing, and maintaining the focus of the inspection on the code

a reader, who is responsible for informing the other reviewers of the current focus of discussion, pacing the meeting, and introducing and summarizing the next piece of code to be discussed

a recorder, who is responsible for writing down defects found during the inspection, classifying those errors according to predefined categories, and noting action items to take care of after the review completes

a producer, who is the writer of the code being inspected, responsible for answering any specific questions about the code and objectively presenting information about the code

The course creates a virtual world in the form of a software development company in which the student learns about and participates in code inspections. The student takes the role of a software engineer who has just joined this software development company. The student can access various "rooms" in the company, including an auditorium, training room, library, office, and conference room.

The student controls where to walk in the company's building, deciding which rooms to visit, with each room serving a specific purpose. The auditorium introduces the student to the importance of software quality and to the company environment in which the student will learn about code inspections. The training room gives background information on related topics such as the roles in an inspection and other types of software technical reviews. The multi-media library provides textual resources and short video segments of what to do and what to avoid during inspections. The office provides the student with tools for preparing for the inspection, such as a source level debugger and hypertext tool for examining the code. Within the conference room, the student takes part in a code inspection simulation, which is the subject of Section 3.2. In each room, the student interacts with appropriate simulated personnel, such as a trainer in the training room, a librarian in the library, and his or her secretary in the office. A comprehensive overview of the code inspection course is provided in [Stevens89], which describes further the contents of these rooms and the overall instructional philosophy inherent in the course's design.

The Advanced Learning Technologies (ALT) Project began development of the course by first prototyping an inspection simulation using a digital speech board from Texas Instruments and a digital still image board from AT&T. When a beta test version of DVI became available in December 1988, development began on the DVI version of the code inspection course, complete with motion video. This course initially ran using version 1.05 of the DVI Software Libraries, and migrated to versions 2.00, 2.01, 2.06, 2.08, 2.10, and 2.12, the version of the DVI Software Libraries used at the time of this article.

The software for the code inspection course contains approximately 50,000 lines of commented C code, a few hundred lines of assembly code, and roughly 8,000 lines of OPS83, a programming language for developing expert systems [Forgy89]. The OPS83 code implements approximately 200 complex rules and associated procedures for the code inspection simulation, which is the focus of Section 3.2.

3.2 Code Inspection Simulation

The primary instructional component within the inspection course is the simulation of a code inspection in which the student has a fully active role. Early on in the course, the student chooses to be either the moderator, reader, or recorder for an inspection. The student then assumes all responsibilities for that role, as well as the general responsibilities of a reviewer, which include the discovery of errors. In the conference room, the student interacts with the other three reviewers while inspecting an Ada code artifact. These three reviewers are simulated by the computer during the code inspection through use of behavioral modeling discussed in Section 2.3, and are referred to as "personae." Note that the set of simulated participants is dependent upon the role selected by the student. The student retains control over when he or she wishes to communicate to the others, who to talk to, the contents of what is to be said, and the tone of the comment.

In agreement with the focus on output fidelity expressed in Section 2.1, the personae are represented with motion video and digitized speech, rather than some combination of text, synthesized speech, and graphic animations. Because of the storage limitation of the delivery platform (680 Mb for one CD-ROM), and the large storage requirements for motion video, however, the personae are not exclusively represented by motion video but also are presented in a slide show format, that is, animated still images with audio.

Many instructional simulations lay out all of the possible interactions of a student with the system and then create a traditional program to handle those interactions, which are often grouped into scenarios. In following the facets of the paradigm expressed in Section 2.2, the inspection simulation was designed so that the student could potentially change the course of conversation with every comment he or she makes. Rather than utilizing scenarios of multiple minute conversations, individual discourses are used as the building blocks of the simulation, and the choice of which discourse to play is based in part on the student's interaction. Thus, there are far too many branches to handle with traditional programming techniques. A forward-chaining, rule-based expert system was developed to run the inspection simulation.

The rule base was developed from analyses of taped code inspections, analyses of normal conversations, and a review of the group process literature, including the work of Bales [Bales5l, Bales70]. The resulting expert system is used to control the topic of conversation, determine who should speak next, model certain "personality" factors of the three personae, track the student's progress, and change the behavior of the three persons based on the context of the inspection. These activities of the rule base are in agreement with the goals expressed in Sections 2.3 and 2.4.

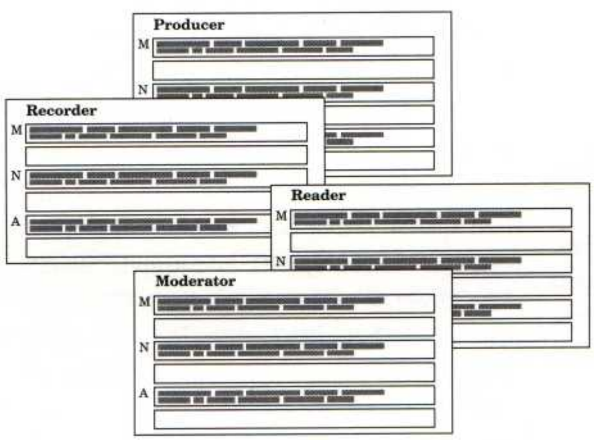
The expert system decides what should be presented next, and by which reviewer. It then selects a discourse from a four-dimensional multimedia database, illustrated in Figure 6.

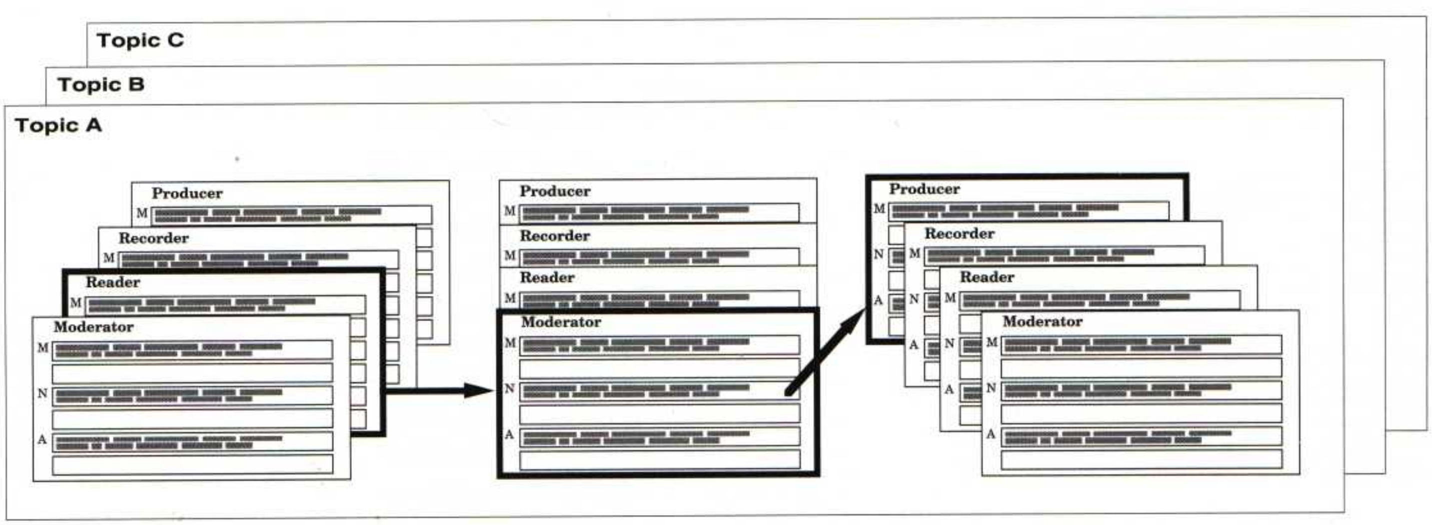
The first dimension is the topic (context space) under discussion in the inspection. The second dimension is the speaker (that is, one of the reviewers). The third dimension is specificity within the topic, which is related to the temporal aspect of conversations. (As people speak on a topic, they tend to get more specific and build on what was said earlier in that topic's discussion.) Finally, the fourth dimension is affect or emotion. (Something can be said passively, neutrally, aggressively, or defensively.) Within each dimension data types include: audio, motion video with audio, and still images. Different data types are integrated together and concatenated to create the audio and video output of the personae.

Figure 6
Four-Dimensional Discourse Structure of an Inspection Simulation

Individual cards represent the *speaker,* with areas within the cards indicating *affect* (for example, mild, neutral, and aggressive).

A stack of cards represents one temporal level of a topic, with each rectangular region below representing a topic.





These dimensions are important for learning about inspections. For example, an effective moderator performs tasks related to each of these dimensions:

- Make sure that all relevant topics are optimally covered and irrelevant topics are minimally discussed.
- Check that everyone gets a chance to speak and that everyone does contribute.
- See that a topic is either resolved or tabled as an action item if discussion continues for a lengthy period.
- Maintain everyone's emotional levels, and make sure that situations like the producer becoming too defensive do not happen.

Therefore, these four dimensions used by the simulation each have a purpose and are important for providing a realistic experience that can aid in learning about inspections.

The expert system looks at several factors in determining what should be selected next from this four-dimensional space (topic, speaker, specificity within topic, and tone). It bases these decisions in part on the attributes of the personae, including:

- predisposition to ramble, talkativeness, aggressiveness, and defensiveness factors of the reviewers at a given point in the inspection
- the current state of the conversation
- the short-term history of the inspection
- the long-term history of the inspection

The expert system also contains rules that update the factors of the reviewers during the course of the inspection.

Along with storing personality attributes for each of the four reviewers, the state of the conversation is saved in the expert system's database. The state includes the current topic of discussion, the length of focus on this topic, whether it has been resolved or not, each participant's spoken stance on the topic, the current speaker, the person focused on in the current comment, and the person addressed in the current comment. For example, if the moderator tells the reader that the code is poorly written, the producer is the person focused on, and the reader is the person addressed. The producer may be emotionally affected by this comment despite not being the person addressed.

The student is an active participant in the inspection; he or she can be addressed by the other personae and can in turn address them with comments. A reasonable solution to the problem of how the student speaks to the other reviewers was found in menu-based natural language interfaces [Hendrickson88]. Harry Tennant of Texas Instruments did much of the pioneering work in this area [Tennant83]. The student communicates with the other reviewers by constructing sentences from sentence fragments, using the interface pictured in Figure 7.

Each sentence fragment is located in a separate menu. For example, one menu might contain variable names; and the other menu, the phrases "is a good name," "is a bad name," "is not needed," and "is initialized incorrectly." By selecting phrases from these menus, the student can construct a sentence that says "The variable `temp' is not needed." The student has the flexibility to create a number of meaningful sentences for any given context, while the syntax and domain is restricted enough to be manageable by the inspection course program. The content layout of the phrases in the menus is based in part on past inspections analyses with the IBM Federal Systems Division [Letovsky88].

Figure 7
Student's Input Interface During Inspection Simulation

The student's view of the inspection is presented in Figure 1 (on page 9). When the rule base decides that one of the three pictured personae should speak, that discourse is presented via digitized speech, and accompanied by either a series of still images or synchronized motion video of that reviewer. Such a presentation of the personae speaking to each other and to the student continues until either the student indicates his or her wish to speak, or one of the personae asks the student a direct question. At that point, the student enters the communication interface shown in Figure 7. When the student finishes constructing the sentence he or she wishes to say, the student returns to the interface of the three personae, and the inspection resumes with either one of the personae or the student making the next comment.

The code inspection simulation implements the concepts presented in Section 2; as such, it possesses the benefits discussed in that section. The use of discourses as building blocks allows conversations to be constructed dynamically, with meaningful modifications to this construction based on the student's contributions. The discourses are organized via header information according to speaker, topic, specificity level, and tone, as well as direction of gaze and camera angle. This organization supports easier retrieval of the discourses from the thousands stored on the CD-ROM for the code inspection course. Modeling the behavior of the code inspectors allows the student to receive experience in more than one role, and allows the student to use the course in isolation. The code inspection simulation provides a controlled environment in which to test the student's reactions to different situations. These reactions are tracked dynamically during program execution so that the student receives detailed feedback on his or her inspection performance. The next section will discuss further the presentation schemes employed by the code inspection simulation.

3.3 Dynamic Multimedia Presentation

Through the use of digital audio, digital video, and the partitioning of the output data into many discourses rather than a few scenario conversations, the code inspection simulation has the capability to tailor the output to the student. This section will present a few examples illustrating how the role of the student, the interaction history within the inspection, and even personal attributes such as the student's name are used by the simulation program to present a more realistic experience of the code inspection to the student.

The capabilities of digital video were critical in allowing the student to participate in different roles within the inspection. Depending on the chosen role, the student either saw the reader, producer, and recorder; moderator, producer, and recorder; or moderator, producer, and reader. Digital video allows the display to be created dynamically from smaller pieces, so that a scene such as Figure 1 can be composed from the five pieces illustrated in Figure S. Visually, the student's selection of a different role now only involves replacing the piece corresponding to that role with the image of the reviewer currently not shown in the figure. The advantages of this jigsawing of images over analog video approaches have been discussed in Subsection 2.2.2.

Digital audio can be manipulated during runtime, and this capability is utilized during the code inspection simulation. The audio is personalized whenever appropriate by saying the name of the reviewer who is addressed. For example, suppose that Diane is the name of the moderator being simulated by the system. A comment to the moderator stating "Do you think this is an error? I don't think so," can now become "Do you think this is an error, Diane? I don't think so." Such piecing together of audio is trivial when the data is all stored digitally. The audio comment cannot be stored with the name embedded, however, because the student may decide to take the role being addressed. Suppose in this example, the student decides to be the moderator, and the student's name is stored in the audio database. Then the student's name, and not Diane's, should be included in the audio directed to the moderator. Approximately two hundred names are included in the audio database for such purposes.

The selection of still images to accompany an audio discourse is very important for the fidelity of the simulation. For example, it would be inappropriate to display the producer with a hostile facial expression and arms banging on the table if the accompanying audio is passively stating in a pleasant tone: "That's a good point. I fully agree with you and appreciate your support." After consulting with directors of visual media and reviewing the cinematic literature, the members of the ALT Project developed a system of rules named "Hitchcock" to support the intelligent presentation of still images.

Hitchcock is used to improve the choices of still images to animate during the playback of an audio discourse. It works on a database of more than a thousand images. Much as the discourses for the code inspection simulation are organized according to a four-dimensional structure, this image database is also organized according to specific attributes. Of primary importance is the reviewer or set of reviewers included in a still image. Other attributes include the direction of gaze of the pictured reviewer(s), the emotion shown, the gesticulation used, and the camera angle used in recording the still image. Remember that the inspection simulation is presented as if the student is sitting at a table with the other reviewers, and the student sees the scene from his or her point of view, as illustrated in Figure 1. Hitchcock is composed of rules such as selecting stills containing a majority of looks toward a given reviewer when the audio is specifically addressed to that reviewer, and selecting stills for the speaker conveying the emotional tone of the spoken discourse.

Cinematic principles were employed, where possible, to enhance the visual cues of the group dynamics taking place during the inspection. For example, if the student has not been contributing at all and is being dominated by the other reviewers, stills recorded with the camera pointing up at the reviewers will be presented to the student to strengthen the apparent dominance of the other inspectors over the student. Likewise, if the student is the dominating force and the other reviewers have become meek and submissive, stills created with the camera pointing downward will be used to support the student being in a position of dominance.

The standard view given to the student during the inspection simulation shows the three other reviewers participating in the in spection. This view is given in Figure 1 (on page 9). Hitchcock rules specify occasions when close-up stills of a single reviewer or of two reviewers should be used instead of the standard view. One such rule states that if a reviewer has not been shown with a close-up for quite some time, and that reviewer has just

Figure 8
Pieces of the Inspection Scene

had a sudden shift of emotion from passive to aggressive with the current dialogue, then show a close-up of that reviewer to enhance the dramatic effect.

The current code inspection course is limited to playing back one motion video file at a time, or else to playing back a single audio file with the accompanying cycling of a small set of still images. More powerful multimedia architectures are under development (for example, the work being done at Carnegie Mellon University's Information Technology Center [Marks91]). As these new architectures become available, such constraints will be made obsolete so that more than one reviewer will be able to move simultaneously, and multiple audio and video files will be able to play at once. This will allow not only the simulation of the speaker of a discourse, but also the simulation of the reactions of the other reviewers to that discourse as it is being spoken.

4 IMPLICATIONS

Improving visual fidelity has long been a driving force in simulator development. For example, it is a driving force behind the evolution of flight simulators, as discussed in [Nash91]. Greater visual realism, however, comes at a great cost, with high-end simulators costing several million dollars. The paradigm described in Section 2 postulates the use of fine-grained digital multimedia data as a means of achieving higher simulation output fidelity without an exponential increase in cost. This enables the development of instructional simulations for problems that were previously deadlocked due to budgetary constraints and output fidelity requirements. The inclusion of the rule base in the paradigm for behavioral modeling and student tracking addresses the overall fidelity of the simulation, which also needs to be improved to utilize fully a rich set of output capabilities.

The paradigm has direct relevance on the development of other group process low-end simulations. Simulations of negotiations, in terviews, command and control center communications, and other human interactions can be improved by incorporating the techniques established here. The paradigm has bearing on analog videodisc simulations and embedded training. These relationships will be explored through the use of a few examples.

Consider first the interactive video simulation for advanced marksmanship training discussed in [White90]. A portion of this train ing system is concerned with presenting scenarios of criminals, terrorists, and other law enforcement situations to the student. As described in Section 2.2, the interactivity of this simulation could be increased if the scenarios were broken up into smaller components. The smaller components could then be recombined in different ways depending upon the student's actions. The simulation fidelity could be further enhanced through the use of behavioral modeling of the actors in the scene. For example, the criminals could react differently depending on what commands are said and actions taken by the student. The increased simulation fidelity offered by this behavioral modeling works best when the visual fidelity can be increased as well; for example, if the criminals are modeled, then their various behaviors should be illustrated in the display. This likely involves the capability to be able to reconfigure scenes, one of the advantages that digital video offers over analog video systems.

Next consider an electronics troubleshooting instructional simulation that is currently under development [Betac90]. During image production a camera crew filmed meters, probes, other diagnostic tools, and the test equipment (that is, the equipment that the student will be troubleshooting). The simulation provides the student with the ability to pick up the test equipment and probes and place them on a workbench. After opening the faulty equipment, the student must place the probes at the appropriate locations and set all of the equipment (dials, buttons, etc.) as it would be in the real-life case.

In this example the application of the facets of the paradigm can create a high-fidelity simulation. By representing each of the diag nostic tools and the components of the test equipment as separate digital video images, an accurate, manipulable image of the workbench can be dynamically created. By modeling the dynamic behavior of the equipment, the appropriate signals can be given to the student when a probe is attached to the equipment being tested. These signals are the audio and visual feedback that the student would receive if performing the same operation in the real world, with the more invasive feedback of remediation and advice presented after the student has completed his or her attempts at troubleshooting.

In this example it would be difficult to render all of the tools and pieces of equipment as graphical objects, and it also would be less realistic than using actual video images. The output fidelity is handled adequately through the use of a set of digital multimedia building blocks. The video images are critical since visual discrimination of parts and locations for signal injections and readings are difficult but crucial to success. The simulation fidelity is improved through the behavioral modeling of the equipment, allowing the probe to be placed in erroneous locations and still have appropriate feedback played back to the student. The input fidelity in this case is less important. Using a real probe rather than a computer input device like a keyboard or mouse would only add some practice in the manipulation of probes and knobs, where the focus of the simulation is on teaching diagnostic skills.

As a final example, consider the forward observer trainer described in [McBride90], in which the student must learn to identify and locate objects in a wide variety of terrains. Visual discrimination is critical to the task, and digital video is used to dynamically add and move objects in the terrain being observed by the student. The student performs the observations in the simulation from a fixed position, and has tools such as a periscope or binoculars available for use. Similarly, in performing the real- world task, the person doing the targeting is looking through a periscope or binoculars from a fixed position. This is an ideal situation for texture mapping and digital video, which offer greater realism than graphics rendering techniques. Behavioral models could be used for the targets to enhance the realism of their movements. For example, if artillery fire is descending on an adversary's position, the movement of the adversary target objects could be modified to avoid the destination location of artillery fire and seek out the source of this fire. The appropriateness and precision of the targets' actions can be based on a history of the scenario and behaviorally modeled tactics and doctrine, accounting for such factors as crew fatigue. The adversary behavior thus becomes much more realistic and complex than if the adversary simply responded to environmental factors.

Embedded training is a concept whereby the training is integrated into the actual system, bringing the power of training technology to the system user rather than requiring the user to travel to a training site often removed from the system environment. Low-end simulators have the potential for delivering embedded training in that they run on small, relatively inexpensive platforms readily integrated into other systems. The advantages of embedded training include removing training travel costs, enabling training on the actual equipment in the functional context of the job, and allowing the student to access the training whenever necessary since it is located with the system.

The simplest training might only be in the form of a paperless technical manual that accompanies a piece of equipment. The manual might consist of a CD-ROM filled with hundreds of thousands of text pages, audio, and video information. The techniques of Subsection 2.2.3 could be used to help organize this data.

More complete training would include instructional simulations showing how to use, maintain, diagnose, and repair the equipment. Digital multimedia building blocks could be used for output realism, behavioral modeling employed for simulation fidelity, and intelligent tutoring techniques implemented so that the embedded trainer could track student progress without the need for external trainer support. Moreover, incremental changes to the rule base can supply expert system support for diagnostic tasks in active maintenance situations.

5 CONCLUSION

The Advanced Learning Technologies Project developed a code inspection simulation for use in a digital video computer course on the topic. In creating this simulation, a paradigm was defined for developing group process instructional simulations. This paradigm can be used to bring a new level of fidelity to low-end instructional simulators for group process, tactics, and doctrine.

The implications of this paradigm extend beyond group process to those problems for which high-end simulators are impractical and low-end simulators have been inadequate. High-end simulators are often impractical because of their high cost, fixed facilities, more limited throughput, and high technical and administrative support requirements. Low-end simulators offer the advantages of low-cost, portable systems, but traditionally have offered inadequate visual display and modeling capabilities.

The application of this paradigm to the code inspection simulation has shown that the use of fine-grained digital video and audio components offers the potential to vastly improve the output fidelity of low-end simulation. The emergence of DVI and other digital technologies provide powerful display capabilities in personal computer packages at little more than the price of a standard personal computer. The delivery hardware for the code inspection course produced by the Advanced Learning Technologies Project costs less than 7000 dollars, in units of one. With the incorporation of the behavioral modeling and intelligent tutoring capabilities discussed in this paper, such low-end simulations now have the potential to address those problems for which low-end simulations were formerly inadequate and high-end simulations, impractical.

As the need for training in the field and on the job increases, the demand for portability in simulators will also increase. Likewise, as training budgets and staffs are diminished, the desire for stand-alone training and embedded tutoring and evaluation capabilities in simulators will increase. As these trends continue, they will direct further attention on this paradigm, which is important for indicating improvements which can be made in low-end simulators through dynamic scene creation using digital multimedia and rule bases for behavioral modeling.

6 REFERENCES

Ackerman89 Ackerman, A. F., Buchwald, L. S., and Lewski, F. H. "Software Inspections: An Effective Verification Process." *IEEE Software, 6,* 3 (May 1989), 31-36.

Bales5l Bales, R. F. *Interaction Process Analysis: A Method for the Study of Small Groups.* Cambridge, Massachusetts: Addison-Wesley, 1951.

Bales70 Bales, R. F. *Personality and Interpersonal Behavior.* New York: Holt, Rinehart, & Winston, 1970.

Betac90 Betac Corporation. *Demonstration of Electronics Troubleshooting DVI Application.* Demonstrated at the 1990 CD-ROM Expo, Boston.

Bosco86 Bosco, J. "An Analysis of Evaluations of Interactive Video." *Educational Technology, 26,* 5 (May 1986), 7-17.

Christe191 Christel, M. G. *A Comparative Evaluation of Digital Video Interactive Interfaces in the Delivery of a Code Inspection Course.* PhD Thesis, Georgia Institute of Technology, Atlanta, Georgia, 1991.

Fagan76 Fagan, M. E. "Design and Code Inspections to Reduce Errors in Program Development." *IBM Systems Journal, 15,* 3 (1976), 182- 211.

Fletcher90 Fletcher, J. D. *Effectiveness and Cost of Interactive Videodisc Instruction in Defense Training and Education.* IDA PAPER P-2372, Institute for Defense Analyses, Alexandria, Virginia, July 1990.

Forgy89 Forgy, C. L. *The OPS83 User's Manual, System Version 3.0.* Pittsburgh, Pennsylvania: Production Systems Technologies Inc., 1989.

Fowler86 Fowler, P. J. "In-Process Inspections of Workproducts at AT&T." *AT&T Technical Journal, 65,* 2 (March 1986), 102-112.

Hassett86 Hassett, J. and Dukes, S. "The New Employee Trainer: A Floppy Disk." *Psychology Today, 20,* 9 (September 1986), 30-36.

Hendrickson88 Hendrickson, J. J. and Williams, R. D. *The Effect of Input Device on User Performance with a Menu-Based Natural Language Interface.* Technical Document 1224, Defense Technical Information Center, Alexandria, Virginia, January 1988.

Humphrey89 Humphrey, W. S. *Managing the Software Process.* Reading, Massachusetts: Addison-Wesley, 1989.

IEEE89 IEEE. *IEEE Standard for Software Reviews and Audits.* ANSI/ IEEE Std. 1028-1988, IEEE, New York, June 1989.

Kenworthy87 Kenworthy, N.W. "Evaluating the Effectiveness of Generic Interactive Videodisc Skill Performance Job Aids." *Optical Information Systems,* 7,4 (July 1987), 232-234.

Letovsky88 Letovsky, S., Pinto, J., Lampert, R., and Soloway, E. "A Cognitive Analysis of a Code Inspection." *Empirical Studies of Programming,* G. Olson, S. Sheppard, & E. Soloway, eds. Norwood, New Jersey: Ablex Publishers, 1988, 231-247.

Maguire90 Maguire, J. A., Chen, T. T., and Barbee, D. "Applying Fault-Isolation Techniques to Analysis of Student Performance in an Intelligent Tutoring System." *Proceedings of the Eighth Annual Conference on Technology and Innovations in Training and Education.* Colorado Springs, Colorado: American Defense Preparedness Association, March 1990, 220-231.

Marks91 Marks, A., Crumley, P., Infortuna, A., and Pasieka, M. "Continuous Time Media Systems: The Next Generation." *Proceedings of the Continuous Time Media Workshop.* Pittsburgh, Pennsylvania: Carnegie Mellon University, June 1991.

McBride90 McBride, W.G., Jr. and Brown, S.F. "Using Digital Video Interactive (DVI) Technology in an Intelligent Tutoring System for Artillery Forward Observers." *Proceedings of the Eighth Annual Conference on Technology and Innovations in Training and Education.* Colorado Springs, Colorado: American Defense Preparedness Association, March 1990, 260-273.

Muller-Malek88 Muller-Malek, H., Matthys, D., and Romaen, H. "Visual Interactive Simulation, A Powerful Tool for Management Education and Decision-Making." *Proceedings of the 1988 Summer Computer Simulation Conference Held in Seattle, WA.* San Diego, California: SCS, July 1988, 554-559.

Nash91 Nash, T. "Visual Systems for Flight Simulators." *Military Simulation & Training,* 3 (March 1991), 8-19.

Ripley89 Ripley, G. D. "DVI - A Digital Multimedia Technology." *Communications of the ACM,* 32, 7 (July 1989), 811-822.

Robinson90 Robinson, P. "The Four Multimedia Gospels." *Byte,* 15, 2 (February 1990), 203-212.

Saunders89 Saunders, J. "The Use of New Technologies in the Design and Development of F-16 Maintenance Trainers." *Proceedings of the IEEE National Aerospace and Electronics Conference.* IEEE, May 1989.

Smith86a Smith, P. "Low Cost Simulations." *Educational Technology,* 26, 6 (June 1986), 35-39.

Smith86b Smith, S., Jones, L., and Waugh, M. "Production and Evaluation of Interactive Videodisc Lessons in Laboratory Instruction." *Journal of Computer-Based Instruction,* 13, 4 (Autumn 1986), 117-121.

Stevens89 Stevens, S. M. "Intelligent Interactive Video Simulation of a Code Inspection." *Communications of the ACM,* 32, 7 (July 1989), 832- 843.

Tennant83 Tennant, H. R., Ross, K.M., and Thompson, C. W. "Usable Natural Language Interfaces Through Menu-Based Natural Language Understanding." *CHI Proceedings,* ACM, 1983, 154-160.

USSCAS90 United States Senate Committee on Armed Services. *National Defense Authorization Act for Fiscal Year* 1991. Senate Report 101-384, United States Senate, Washington, DC, 1990.

Waterworth85 Waterworth, J. A. and Thomas, C. M. "Why is Synthetic Speech Harder to Remember Than Natural Speech?" *CHI Proceedings,* ACM, 1985, 201-206.

Watkinson90 Watkinson, J. *The Art of Digital Video.* Boston, Massachusetts: Focal Press, 1990.

Weir90 Weir, H. "The Evolution of Computer Simulations in Physics Teaching." *Proceedings of the Conference on Computers in Physics Instruction Held in Raleigh, NC (August* 1988). Redwood City, California: AIP, APS, NSF, IBM et al., 1990.

White90 White, C. R. and Carson, J. L. "Evaluation of an Advanced Marksmanship Trainer: Cost Analysis." *Proceedings of the Eighth Annual Conference on Technology and Innovations in Training and Education.* Colorado Springs, Colorado: American Defense Preparedness Association, March 1990, 162-172.

About the Authors

Michael Christel is a member of the technical staff working on the Advanced Video Techniques for Imaging and the Requirements Engineering Projects.

Scott Stevens is a senior member of the technical staff at the SEI and adjunct senior research scientist at the Carnegie Mellon Center for Development of Educational Computing.