Journal of Computing in Higher Education
Fall 2004, Vol. 16(1), 30-?.

Enhancing Computer Science Education with a Wireless Intelligent Simulation Environment

Diane J. Cook, Manfred Huber,
Ramesh Yerraballi, and Lawrence B. Holder
Department of Computer Science and Engineering
University of Texas at Arlington

ABSTRACT

THE GOAL OF THIS PROJECT is to develop a unique simulation environment that can be used to increase students' interest and expertise in Computer Science curricula. Hands-on experience with physical or simulated equipment is an essential ingredient for learning, but many approaches to training develop a separate piece of equipment or software for each topic area. We describe the development of a simulation environment that provides a foundation for cross-disciplinary exercises. The Wireless Intelligent agent Simulation Environment (WISE), which combines activities from virtual and physical versions of the Wumpus World game, provides a dynamic learning environment that can enhance a number of Computer Science courses.

In this paper, we describe the WISE environment design. We also describe steps for integrating WISE into Computer Science curricula. As a demonstration of the effectiveness of the tool, we describe its use in the artificial intelligence, multimedia, and wireless networks courses at the University of Texas at Arlington. (*Keywords: Simulation, intelligent agents, distributed environment, multimedia computing, wireless networks, interactive games*)

INTRODUCTION

THERE EXISTS a need for a change in course work in areas such as mobile computing, artificial intelligence, multimedia, wireless networks, robotics, and distributed systems that provides crucial hands-on experience in a collaborative environment. Many studies indicate the value of computer-based multimedia and simulation tools to learning (Clark, 1997; Graham & Trick, 1998; Reinhardt, 1995). This is perhaps no more important than in Computer Science curricula, where students perform a majority of their work in front of a computer screen. In response, we are developing an environment in which students can electronically access all class material. We introduce a tool in this environment that simulates and tests the taught concepts in a dynamic, competitive, and complex simulation environment.

A number of simulation environments are currently in use at different universities to support research and teaching. In Artificial Intelligence, for example, the TileWorld system (Pollack & Ringuette, 1990) supports simulation of agent design in a dynamic two-dimensional grid world. The Truckworld simulator developed at the University of Washington (Hanks et al., 1993) simulates a delivery scenario to test planned delivery schedules and responses to unforeseen situations. The Intelligent Home Project Java-based simulator at the University of Massachusetts (Vincent et al., 2001) allows students to create intelligent agents and study their interactions. Similarly, the Michigan Intelligent Coordination Experiment (MICE) testbed (Montgomery & Durfee, 1990) focuses on multiagent interaction in a timed environment. A number of tools to simulate aspects of multimedia computing, robotics, and wireless networks are also now commonly available.

While these environments present valuable teaching or research tools, their application within the curriculum is generally limited to a single topic. In contrast, the tool we are developing addresses a wide range of topics from different parts of the Computer Science curriculum. The goal here is not only to provide an environment within which to implement assignments for a particular course but also to increase students' interest and expertise in Computer Science curriculum through hands-on experience in a unique interactive game environment.

In our earlier work we reported the results of using a Wumpus World game simulator as a method of enhancing artificial intelligence curriculum (Holder & Cook, 2001). Here, we introduce an interactive environment representing an extension of this effort, the Wireless Intelligent agent Simulation Environment (WISE). The WISE environment allows software agents to play competitively or cooperatively with human and robot agents in a virtual or real-world model of the Wumpus World computer game, a grid-based adversarial game. We describe integration of the tool into diverse Computer Science courses including artificial intelligence, multimedia, and wireless network classes. Our research goal is thus developmental, focused on the creation of an interactive distributed environment for enhancing learning in Computer Science courses. Our assessment of the results will be both quantitative, through measured responses to the environment, and qualitative, through interviews and personal feedback.

WISE SIMULATION ENVIRONMENT

INSTEAD OF DEALING WITH EACH TOPIC area in isolation, many real-world technology tasks require blending of expertise in multiple areas including wireless communication, decision making, and cooperation between human and software agents. Our Wireless Intelligent agent Simulation Environment, called WISE, represents a distributed wireless decision making environment that allows cooperation or competition between autonomous mobile agents, which can be humans, software agents, or robots.

THE WUMPUS WORLD GAME

The basis for the WISE environment is the Wumpus World computer game. The game's agent explores a square grid world of arbitrary size while avoiding a creature known as the wumpus. Other elements

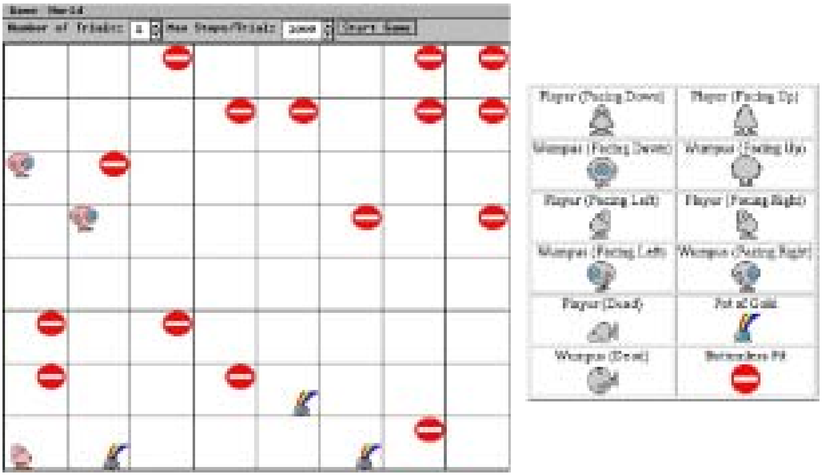

Figure 1. Sample Wumpus World game

of the world consist of obstacles, bottomless pits (which do not affect the wumpus), and bars of gold. The agent starts in the lower left cell of the world with a single arrow in his arsenal. The objective of the game is to collect the gold bars, return to the initial grid location and exit the world. If the agent steps into a square containing a wumpus or a pit, he immediately loses the game. **Figure 1** shows an example Wumpus World game. Our implementation of the game follows the description given by Russell and Norvig (2003).

At the start of each turn, the agent can perceive his surroundings by means of five basic percepts and two extended percepts. A *stench* percept indicates that a wumpus, which emits a foul odor, is in one or more of the four squares adjacent to the player's location. A *breeze* indicates that a pit is in one or more of the four adjacent squares, and a *glitter* indicates that a gold bar is at the agent's current location. A *bump* is sensed whenever the agent attempts to move into a wall or an obstacle. If an arrow hits a wumpus, the player will sense a woeful *scream*, indicating the wumpus has been defeated. One extended percept is a natural language hint: an English sentence that provides information about the Wumpus World. The second extended percept is a color image, showing the contents of the cell one step away from the agent along his current orientation.

Figure 2. Wise server interface

The player may perform any of the following actions: turn left, turn right, go forward, grab some gold, shoot an arrow straight ahead, and climb out of the world. The player can only climb out of the world from the start location. The player receives a score of -1 for each executed move. Gold bars are worth 1000 points, but only if the player successfully carries the gold out of the world. If he is defeated, the agent loses 1000 points and any gold that he is carrying at the time.

The simulation may be set up to execute multiple trials. Each trial is complete when all the agents in the world are either defeated or victorious, or exceed the maximum allotted time for the trial. The agent may adopt a new strategy based on experience obtained from previous trials.

WISE IMPLEMENTATION

The WISE simulator represents an extension of the Wumpus World game. Instead of modeling a virtual grid world, WISE models an actual physical environment. In our design, we use the second floor of our classroom building. The simulator supports multiple independent agents, each with its own separate logic and possibly separate hardware platform. WISE employs a client-server architecture that separates the simulation code from the agent design. To support platform independence, a Java-based environment was selected. The WISE code can be downloaded from http://ailab.uta.edu/wise.

Upon start-up, agent clients connect to the server as shown in **Figure 2.** The client requests a specific type of *avatar*, or character that executes the moves for the client in the game. The server sends percepts to each client and receives the client's requested action. The server correspondingly updates its model based upon the chosen action. The server also maintains a score for each client, and prints game statistics when the desired number of trials is completed.

Figure 3 shows the WISE architecture. An agent makes decisions about how to move through the environment, the avatar physically executes the moves, and the server keeps track of the resulting game state. There are three types of avatars: software, human, and robot. The software avatar is a virtual entity that exists only in the simulator, while the human and robot avatars move around the physical environment. Figure 3 shows the wearable device we designed for the human avatar as well as our robot avatar. The decision-making abilities for each agent can be performed by a software application or by a human who directly controls the corresponding agent.

Because WISE models an existing physical environment, agents can receive the basic percept information directly from sensing the physical environment or by receiving the percept information from the server, as shown in **Figures 4 and 5**. Natural language extended percepts are provided upon request from the server. If represented by a human or robot avatar, the agent can obtain live video images from its corresponding avatar at a specified refresh rate or on demand. The software agent receives an artificially-generated image percept upon request from the server, as shown on the right in **Figure 4,** in this case depicting an obstacle straight ahead.

In our simulator, we provide the wumpus with a library of possible strategies. The agent will have to rely on its reasoning capabilities to outmaneuver the wumpus. Possible wumpus movement strategies can be specified including sit, spin, random, walk, chase-agent, move-to-nearest-gold, and move-to-nearest-pit.

Figure 3. WISE architecture

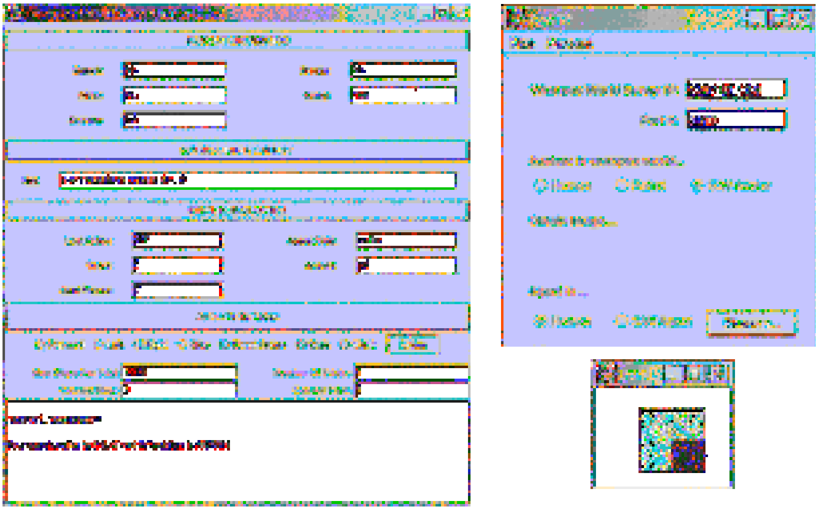

Figure 4. WISE basic and extended percepts

The WISE simulator behaves in the following manner. The user starts up the server, then avatars, then clients. Starting locations are specified for each agent, and then the server starts the first trial. During each trial, the server sends percepts to the agents (clients). WISE display these percepts as shown in **Figure 5**. The agent selects an action, the avatar executes the action, and the server updates the state of the game. The game trial ends when the agent is defeated or climbs victoriously out of the world. If all of the trials are completed, the server will calculate statistics for the agents.

Menus as well as command-line options are provided to specify the world layout, the number of game trials to simulate, the maximum number of steps per trial, the maximum number of agents that will be playing, the message handling scheme (round robin or asynchronous), and the connection wait time. The user can also generate the world layout interactively and save the current layout in a configuration file.

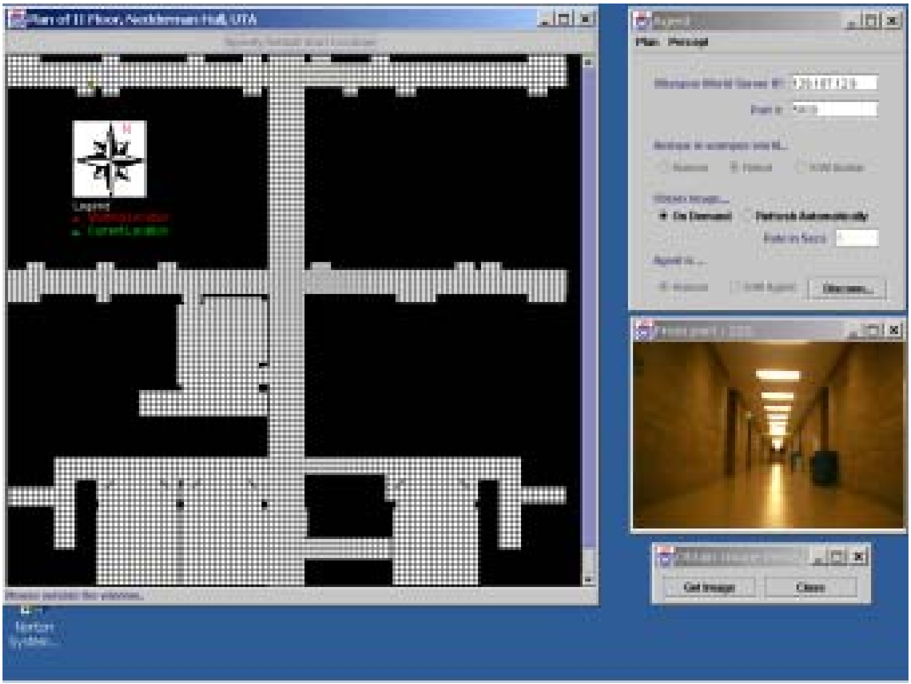

Figure 5. Obtaining image percepts from physical avatars

CURRICULUM DEVELOPMENT USING WISE

THE WISE ENVIRONMENT provides a unique opportunity for students in Computer Science classes to obtain hands-on experience in a distributed, dynamic environment. Classes that can benefit from this environment include artificial intelligence, wireless mobile computing, multimedia, and robotics courses. Lectures in these classes can use WISE as a medium for demonstrating techniques, and students can complete laboratory exercises that test, expand, or modify the simulator. Here, we describe the use of WISE in three Computer Science classes: artificial intelligence, multimedia computing, and wireless mobile networking and computing.

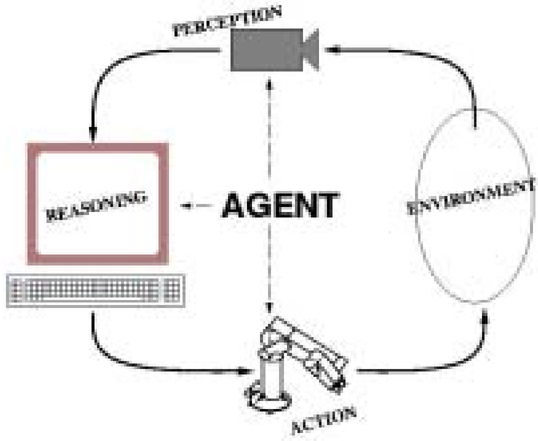

Figure 6. Intelligent agent design

ENHANCING AN ARTIFICIAL INTELLIGENCE CLASS USING WISE

A topic for which WISE is well suited is *artificial intelligence*. We demonstrate integration of WISE into the artificial intelligence (AI) curricula and evaluate WISE as a learning tool using feedback from students in the class.

We introduce artificial intelligence (AI) class material by describing the idea of an intelligent agent and then presenting techniques for creating such agents (Russell & Norvig, 2003). As shown in **Figure 6**, intelligent agents operate by receiving a percept, or set of sensory information that the agent can detect about the environment. An intelligent agent must employ reasoning techniques to understand the environment, formulate an appropriate action, and change the environment by executing the selected action. Students in the AI classes learn about the goals of intelligent agents, and learn current techniques for creating these agents.

In this study, WISE serves as the primary medium for exploring the capabilities of various types of intelligent agents. Using this simulator, students acquire hands-on experience with AI techniques. Because one common environment provides the foundation for a majority of student homework assignments and projects, students can compare the performance of an intelligent agent with and without each of the techniques introduced in class. The teaching environment we have developed provides not only the multiplatform simulated agent environment, but also a set of AI algorithms that operate within the environment, including tools for knowledge representation, search, planning, learning, vision, uncertainty reasoning, multiagent coordination, and natural language processing.

We have developed a set of assignments listed below that allow students to learn about a variety of artificial intelligence techniques in the context of the WISE simulator. Each assignment builds on WISE and enhances the student's previous addition, allowing students to see how performance of an intelligent agent can improve with each new technology.

1. **Simple Reflex Agent**. This agent is reflex-driven, reading the sensor input and making decisions about which action to select and execute according to a fixed set of rules. Because this agent is not looking beyond the current move, performance will be low. As a result, this agent provides a good starting point for comparison with agents that are more sophisticated.
2. **Reflex Agent with State**. This agent augments the capabilities of Agent 1 by adding state information. The state structure contains all information the agent has gathered about the world to this point in the game, which is used to select the next action.
3. **Search-Based Agent**. For this agent, students implement a search engine for a WISE agent. Using WISE, students study the properties of various search techniques and compare their features. Because this agent has the ability to look ahead at the result of sequences of moves, it will generally outperform Agents 1 and 2, but will require more processing time.
4. **Planning Agent**. The planning agent focuses on planning techniques that use principles of logic and constraint satis-

faction to generate a sequence of actions that achieves a goal. This approach is more efficient than using a search engine to consider exhaustively all possible combinations of moves.

5. **Decision-Theoretic Agent**. In contrast to the previous agents, this agent has the capability to reason about uncertainties in the environment using techniques from decision theory. Pieces of evidence, including sensing breezes, stenches and glitter, contribute to the probability that a pit, a wumpus, or gold is in a given location. These probabilities combine with associated rewards (or costs) to allow the agent to make a decision that will yield the greatest utility. Students use a structure called a belief network to represent probabilities of environment features, to update the probabilities with new evidence, and to decide upon the best action.
6. **MDP Agent**. Markov Decision Processes (MDPs) provide a means of computing optimal action policies from reward functions and probabilistic information about the behavior of the agent in the environment. This agent computes an action policy for an agent.
7. **Learning Agent**. This agent uses machine learning techniques to improve decision making with experience. The agent learns action policies from previous encounters with similar situations, and these policies become increasingly accurate as experience accumulates.
8. **Natural Language Agent**. Initially, agents use only the five basic percepts (stench, breeze, glitter, bump, scream) to select an action. However, the agent can also use natural language statements generated by an oracle in the WISE simulator. Examples of such hints include “The wumpus is in square 1 2,” “The wumpus is behind the agent,” “The agent is facing the wumpus,” and “The size of the world is 8 squares in each direction.” The agent must parse the sentence and extract the information that is beneficial to select an action.

Table 1
Ease of Use and Benefit of WISE

Difficulty of the tool (0=easy … 5=difficult)	
Understanding of the topics using the WISE tool as compared to material not using the simulator (0=much lower .. 5=much higher)	

9. **Vision Processing Agent**. This agent uses additional computer vision techniques determine what objects are in the grid square they are currently facing. The agent "looks" in a specific square and receives a PPM image such as the one shown in Figure 4. Noise is added to reflect typical inaccuracies that vision systems handle.
10. **Multiagent Cooperation**. The WISE simulator permits multiple clients to connect at the same time, facilitating the cooperation of agents on the same task. Here, multiple agents have to communicate information that is learned about the environment and make decisions that efficiently divide the work between them.

For this study, we integrated WISE into the second AI course in the sequence, which focuses on the more advanced techniques and includes a class project. In-class lectures cover the design and use of the tool. Students then design and test sample agents using WISE. Furthermore, students can choose one of the advanced agents (7 – 10) for a semester project. Because all projects employ the object-oriented WISE software, projects can be incorporated into the simulator to provide more functionality each time the class is taught.

To assess the potential benefits of teaching AI with WISE, we tested students' use of the tool. Students anonymously completed a questionnaire at the end of the course, which asks them to evaluate different aspects of the course and of the simulation tool quantitatively on a scale from 0 to 5. This survey was conducted during two consecutive years: seven of the students in the class responded to the each of the two years year. **Table 1** shows the student's basic impression of the WISE tool, averaged over the two years of feedback results.

Table 2
Effect of WISE on Learning and Interest in Assignments

	Mean (Shown for year 1, year 2)					
	Agent 1	Agent 2	Agent 4	Agent 5	Agent 6	Agent 9
Difficulty of the material (0=easy … 5=hard)	1.15, –	2.29, 2.71	2.33, 2.71	4.14, 4.14	3.86, 4.17	–, 3.67
Effect of WISE on learning (0=none … 5=large)	3.43, –	3.71, 4.29	3.33, 3.43	4.14, 3.43	4.14, 4.00	–, 4.00
Effect of WISE on interest (0=none … 5=large)	3.29, –	3.57, 4.29	3.05, 3.71	3.57, 3.56	4.00, 3.86	–, 4.17

Even though use of the simulator required making oneself familiar with Java and parts of the Wumpus World class structure, students generally judged the tool usable on all different computer platforms (students in this course used Windows and Linux PCs as well as Sun workstations). More importantly, despite the fact that material that made use of the WISE simulator was generally more complex than material that did not, students judged their understanding of the WISE material superior to topics not covered in simulation experiments.

To assess more closely the effect of the simulation tool, we asked students to rate their experiences for the individual assignments (since each student had a different project, statistical evaluation would be difficult, so we do not include the projects in this survey). The assignments used in the first year were Agents 1, 2, 4, 5, and 6. In the second year Agents 2, 4, 5, 6, and 9 were used due to a change in our introductory AI course (the first AI course investigated Agent 3 both years and Agent 1 the second year). **Table 2** shows the students' assessments of the difficulty of the assignments and of the effect of the WISE tool on learning and interest in the topics covered by the assignment.

With one exception in each of the years, this data shows a direct correlation between the difficulty of a topic and the perceived benefit of WISE on the learning experience. As the difficulty of the assignments increases, WISE becomes a more useful tool for the understanding of the material. A similar effect can be seen with respect to

Table 3.
Importance of WISE Tool

Importance of a common application domain and tool (0=none … 5=very important)	4.31
Importance of the Wumpus World domain (0=none … 5=very important)	3.93
Importance of the graphical interface (0=none … 5=very important)	4.00
Importance of a common application domain in different courses (0=none … 5=very important)	3.41

increasing the students' interest in AI topics. A possible reason for this is that while the overhead of using the simulator is perceived as significant for simple topics, the benefits of visualization and practical experimentation become dominant as topics become more difficult. An exception to this trend in the second year is the initial assignment (Agent 2), which received the highest ratings for the tool's effectiveness. A possible explanation for this and for the significant difference to its rating in the first year is that this assignment in the second year effectively required students to address the content of Agents 1 and 2. As a result, insight into the domain and the techniques had to be obtained in a shorter amount of time than in the first year, increasing the importance of the simulation tool. In the first year, the exception to the trend is the planning agent (Agent 4), which was not completely integrated into the simulator due to the lack of a Java-based planning package. In the second year, when the planner was fully integrated, the rating of the tool's effect improved. This indicates that the benefit for learning and interest draws directly from using a completely integrated tool.

Table 3 shows the more general effects of the WISE tool on the students' learning experience in the AI course and their assessment of using such a tool across multiple courses. This data demonstrates the importance of a common application domain and a corresponding simulation tool. In addition, the students consider a graphical interface to the tool to be very important. Although still significantly high, a somewhat lower importance is given to the Wumpus World as the particular domain used. This suggests that the main benefit achieved by using the WISE tool in the artificial intelligence course is derived from providing students with an easily usable, uniform simulator that covers all topics. Students also perceive a benefit from using this tool as the basis for experimentation across parts of the curriculum.

Overall, students' understanding, confidence, and interest in the topics that made use of WISE were generally greater than topics that did not make use of the tool. Moreover, the effectiveness increased as topics became more difficult. Students added comments to the survey indicating that this hands-on approach is a good way to learn material because the approach answers many practical questions of theoretical material. They also appreciated the simplicity of adding an agent to an existing environment. Another indicator of the success of this project is the number of other programs that are integrating the tool into their classes. Russell and Norvig (2003) have included a link to our site and our software is accessed many times daily, resulting in comments and questions from faculty using the tool in their programs.

ENHANCING A MULTIMEDIA COMPUTING CLASS USING WISE

As one of the extended percepts, the WISE server can provide an image of the agent's current location to the agent for processing. If the agent's avatar is a human or robot, this visual feedback is multimedia in nature, ranging from simple images (with possible noise) to video and possibly audio. It is important to derive and implement mechanisms for effective coding and compression of these media. The *multimedia computing* class investigates these topics and tests proposed techniques in the WISE environment. In addition, the streamed nature of audio and video poses quality of service demands such as delay bounds, jitter bounds, and synchronization, that cannot be met by traditional TCP/IP communication. The bandwidth-constrained wireless environment offered by WISE provides an excellent challenge for students in this class.

The goal of the Multimedia Course is to expose the students to the following three topics: 1) multimedia coding and representation, 2) multimedia compression techniques, and 3) multimedia communication. The first of these topics involves understanding various media, their analog characteristics and their digital representation. The second topic guides the student from an information theory standpoint towards redundancy in media and the means to eliminate it through compression. Fundamentals of compression are discussed in general and within the context of standards such as jpeg, mpeg, H 261, and H 263. The last topic handles the issue of communicating media (live, stored, or on-demand) over a network. Network scheduling (queueing), quality of service, jitter elimination and ATM are studied in depth.

The primary motive for projects assigned in this class is to give students hands-on experience with practical issues related to implementation of concepts studied in class. This particular course was offered at distance, so the project description provided to students had to include all supporting material (software, algorithms and coding). We designed the following three projects.

1. **Image compression**. Enhance the performance of the Wumpus world game used in the WISE project. Specifically, modify the current visual percept exchange of incompressed images to an exchange of compressed images based on the Adaptive Huffman compression mechanism.
2. **Voice commands and text-to-speech synthesis**. Add voice capabilities to WISE. Specifically, when the human agent wants to interact with the server he/she can now give voice commands, which are converted to speech and sent to the server for processing. When the server sends precepts to the human agent, the percept (text) has to be synthesized into speech (voice).
3. **Video compression**. Enhance video sent between the human or robot avatar and the human agent using compression techniques. Currently, the communication consists of single jpeg frames sent at a certain preset rate (e.g., 5 fps). Each frame is sent in its entirety. In this project, the student must implement a compression mechanism as shown in **Figure 7.** The mechanism is a simple differential coding technique where successive frames are compressed relative to the preceding frames. The compression mechanism used is the same one used in Project 1.

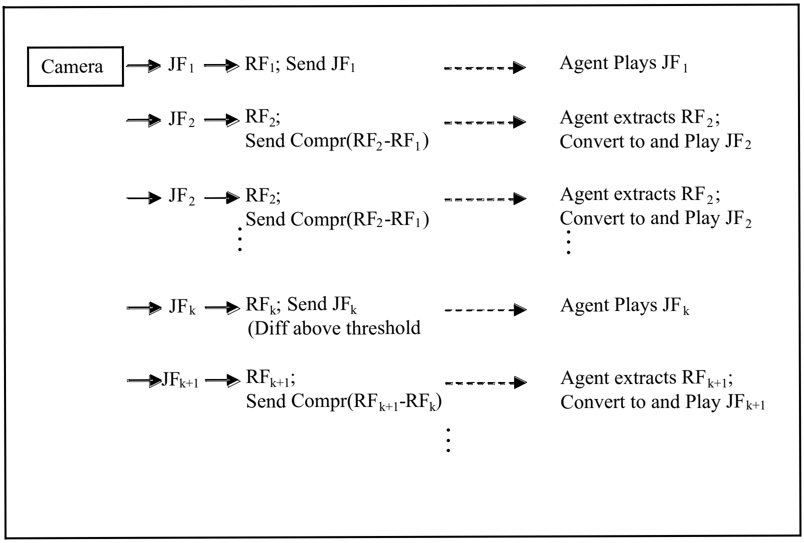

Figure 7. Compression mechanism video sent between an avatar and an agent.

Frames are extracted from the camera in jpeg format (JF_i), which are then converted to a raw format (RF_i). The first frame (JF_1) is sent as a jpeg frame, and the succeeding frames are differentially coded with respect to the previous frame, that is, the i^{th} frame is sent as a difference ($RF_i - RF_{i-1}$) that is compressed using a simple compression mechanism (the mechanism already implemented in Project 1). A simple threshold mechanism is incorporated to make sure that sending the difference as opposed to sending the original frame is justified. Accordingly, when the Mean Square Error of the difference exceeds a user-specified threshold, the frame is sent in its jpeg format. This is the case with the k^{th} frame shown in Figure 7.

All students in the class completed Project 1, and the other two projects were voluntarily chosen by some of the students wanting to conduct additional research on the topic. These students achieved promising results from their implementation (Yerraballi et. al, 2003).

To assess the impact of WISE on the multimedia class, nine students provided anonymous feedback at the end of the semester. The

Table 4.
Impact of WISE Tool in the Multimedia Computing Class

	Mean
Confidence in the course material (0=none … 5=high)	4.61
Confidence in the project material (0=none … 5=high)	3.61
Relevance of WISE as a project platform (0=none … 5=high)	3.33

results, shown in **Table 4**, indicate that students were generally positive about the relevance of WISE for the class. Becoming familiar with the tool through a distance class provided a greater challenge than for the on-campus AI class, however, and we will carefully address this issue in the future.

ENHANCING A MOBILE WIRELESS NETWORKING AND COMPUTING CLASS USING WISE

Students in the *mobile wireless networking and computing* class tackle challenges that arise from the increasing desire for ubiquitous information access provided by handheld portable computers, emerging wireless communication systems, and the advent of technologies such as cellular telephony, personal communication systems, wireless PBXs and wireless LANs. These challenges include dynamic network management, integration of wireless and wireline networks, system support for mobility, location management, quality-of-service provisioning, mobile file systems, mobile IP, wireless data access and networking. In this context, the WISE environment provides the ability to simulate various protocols and algorithmic solutions to these challenges. The ability to communicate between agents, avatars, and the WISE server will depend on providing efficient and robust solutions to these problems.

During the first offering of this class, WISE was used as a platform for a team class project. The main object of the project was to enhance the WISE with wireless network capabilities. The WISE server can now inform an agent about its current wireless signal strength, warn about possible upcoming QoS difficulties, and assist in selecting the best access point with which to connect. The students also added capabilities for WISE to locate and track agents using wireless signal information. The project was successfully completed and demonstrated.

CONCLUSIONS

IN THIS PAPER, we described an interactive game environment for enhancing Computer Science classes. The foundation for the environment is an integrated tool that simulates agent-based technologies mobile computing and networking, multimedia computing, robotics, and distributed environments. Preliminary results indicate that students benefit from using this tool in terms of subject interest, confidence in the material, and ability to understand and utilize the presented techniques. We also demonstrated that a single integrative tool could be useful for diverse courses in the Computer Science curriculum.

Our next step in this effort is to test the environment in additional Computer Science courses, including robotics, human-computer interaction, distributed environments, and computer games. We are also creating a Web interface that will allow students, faculty, and researchers to make use of the WISE environment from their remote location, increasing the impact of this environment on interactive learning.

ACKNOWLEDGEMENT

This work is supported by National Science Foundation grants EIA-86260 and MRI-115885.

REFERENCES

Clark, W.M. (1997). Using multimedia and cooperative learning in and out of class. *Proceedings of the 1997 Frontiers in Education Conference.*

Graham, C.R., & Trick, T.N. (1998). Java applets enhance learning in a freshman ECE course. *Journal of Engineering Education, 87(4),* 391-397.

Hanks, S., Nguyen, D., & Thomas, C. (1993). *A beginner's guide to the truckworld simulator*. Technical Report 93-06-09, University of Washington.

Holder, L.B., & Cook, D.J. (2001). A client-server computational tool for integrated artificial intelligence curriculum. *Journal of Computing in Higher Education*, 12(2).

Montgomery, T.A., & Durfee, E.H. (1990). Using MICE to study intelligent dynamic coordination. *Proceedings of the Second Computer Society International Conference on Tools for Artificial Intelligence,* 438-444.

Pollack, M.E., & M. Ringuette (1990). Introducing the Tileworld: Experimentally evaluating agent architectures. *Proceedings of the National Conference on Artificial Intelligence* 183-189.

Reinhardt, A. (1995). New Ways to Learn. *Byte, 20(3).*

Russell S., & Norvig P. (2003). Artificial Intelligence: A Modern Approach. New Jersey: Prentice Hall.

Vincent, R., Horling, B., & Lesser, V (2001). An Agent Infrastructure to Build and Evaluate Multi-Agent Systems: The Java Agent Framework and Multi-Agent System Simulator. *Lecture Notes in Artificial Intelligence: Infrastructure for Agents, Multi-Agent Systems, and Scalable Multi-Agent Systems*, Volume 1887.

R. Yerraballi, X. Zhao, & J. Kanabar, "A New Asynchronous Hybrid Mechanism for Video on Demand," *To appear in Proceedings of the 29th Euromicro Conference on Multimedia and Telecommunications,* Turkey, 2003.

ABOUT THE AUTHORS

Diane J. Cook is a Professor in the Department of Computer Science and Engineering at the University of Texas at Arlington. Her research interests include machine learning, data mining, and intelligent environments. Present address: Department of Computer Science and engineering, University of Texas at Arlington, Box 19015, Arlington, TX 76019 E-mail: cook@cse.uta.edu

Manfred Huber is an Assistant Professor in the Department of Computer Science and Engineering at UTA who specializes in machine learning and robotics. E-mail: huber@cse.uta.edu

Ramesh Yerraballi is a Senior Lecturer in the Department of Computer Science and Engineering at UTA with research emphases in multimedia communications. E-mail: ramesh@cse.uta.edu)

Lawrence B. Holder is an Associate Professor in UTA's Department of Computer Science and Engineering who performs research in the areas of machine learning and automated discovery. E-mail: holder@cse.uta.edu