The Use of Agents in Human Learning Systems

Elizabeth Sklar
Dept of Computer and Information Science
Brooklyn College, City University of New York
2900 Bedford Ave, Brooklyn, NY, USA 11210
sklar@sci.brooklyn.cuny.edu

Debbie Richards
Dept of Computing, Div of Information and
Communication Sciences
Macquarie University
Sydney NSW 2109 AUSTRALIA
richards@ics.mq.edu.au

ABSTRACT

This paper examines agent-based systems designed for a variety of human learning tasks. These are typically split into two areas: "training", which generally refers to adult learning of job-related skills, frequently but not exclusively in military settings; and "education", which generally refers to child and adult learning in academic settings, including primary and secondary schools, colleges and universities. While the terms may indicate diverse areas within the field of human learning, from the standpoint of agent-based systems development, many of the more prominent issues are held in common; as well, these issues can be generalized to most interactive agent-based environments. Here, we categorize three major trends in development of agents to assist human learners: pedagogical agents, peer learning agents and demonstrating agents. We highlight recent work within each of these categories, bringing to light common themes and issues.

Categories and Subject Descriptors

I.2.11 [**Artificial Intelligence**]: Distributed Artificial Intelligence; K.3.1 [**Computers and Education**]: Computer Uses in Education

General Terms

Design, Human Factors, Experimentation

Keywords

learning, training, education, pedagogical agents

1. INTRODUCTION

The primary goal in a human learning environment is for the learner to advance. Software applications built for this environment, unlike applications for other environments, are not designed to simplify or perform a task *for* the user, but rather to help the user *learn* how to accomplish the task herself. This goal is strikingly different from typical interactive systems, where agents are constructed specifically to assist users by handling tedious or complicated jobs. Such systems address tasks such as browsing [33], sorting email and filtering news group messages [13, 17, 29, 30], or finding other users who share similar interests [4, 15, 28]. By embedding personalized agents and applying a variety of adaptive techniques, these systems can be thought of as tools designed to relieve the user's burden by taking over repetitive or complex duties from an overwhelmed human.

Malone [37, 38] makes an important distinction between *toys* and *tools* when discussing computer games. He defines toys to be systems that exist for their own sake, with no external goals; in contrast, tools are systems that exist because of their external goals. Good tools should be easy to use, in order to expedite the user's external goal. Good games are difficult to play, in order to increase the challenge provided to the player. Whereas typical interactive systems are like good tools, systems built for human learning should of course be easy to use, but primarily should provide challenges for the human learner. The external goal is for the user to learn how to perform a given task, so the system should make the process of learning how to accomplish that task easy — the *process*, not the task [64].

Systems designed to aid human learners are typically split into two categories:

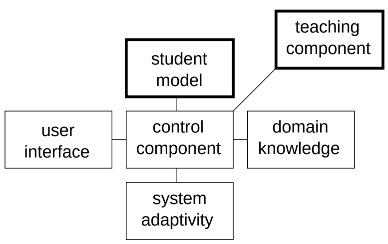
- **training**, which generally refers to adult learning of job-related skills, frequently but not exclusively in military settings; and
- **education**, which generally refers to child and adult learning in academic settings, including primary and secondary schools, colleges and universities.

While the terms "education" and "training" may indicate diverse areas within the field of human learning, from the standpoint of computer systems development, many of the more prominent issues are held in common. We use the general term "interactive learning system" (ILS) to include not only the more specific (and perhaps more familiar) term "intelligent tutoring system" (ITS), but also to provide a broader definition encompassing environments that are designed for more exploration on the part of the student than ITS's (which are typically more structured and scripted according to highly engineered, domain-dependent models). A typical ILS consists of the following components [39, 63]: (1) *domain knowledge* — a representation of the topic that the student is learning; (2) *teaching component* — an instructional model that is used to guide the student through the knowledge domain; (3) *user interface* — the interaction mechanism that lies between the human student and

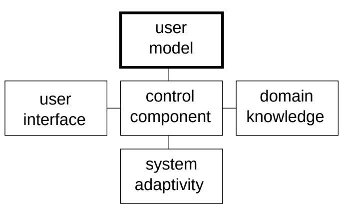
AAMAS'06 May 8–12 2006, Hakodate, Hokkaido, Japan.

the computerized system; (4) *student knowledge* — a "user model" of the student in relation to the domain knowledge, indicating how much of and how well the student knows the domain; (5) *system adaptivity* — the means by which the system adapts automatically to the student's behavior, backtracking when the student makes mistakes and moving ahead when the student demonstrates proficiency with portions of the domain; and a (6) *control component* — the central part of the system architecture which holds all the pieces together.

Figure 1a illustrates these components and their relationships to each other. This architecture is quite similar to that of a typical interactive system for non-learning applications, as shown in Figure 1b. The *student model* is replaced by a *user model*, and there is no *teaching component*. In both cases, different types of agent technology are included at different levels. For example, where *interface agents* are used in non-learning applications to help a user navigate a complex domain, *pedagogical agents* are implemented in a learning application to guide users through a problem space. Understanding and embracing the special characteristics that learning tasks require of an application is necessary in order to adapt techniques from non-learning applications to those designed for education and training. While the focus in this article is on agent-based environments for human learning, we also highlight the relevance of issues for any agent-based interactive application.



(a) interactive learning system



(b) interactive system

Figure 1: Typical system architectures

There are three main types of agents embedded in human learning environments. The first are *pedagogical agents* [23], personalized assistants that interact directly with a learner and explicitly guide her through the domain. Referring back to Figure 1a, pedagogical agents are involved in the teaching component and user interface. Typically they consult the user model in order to understand the student and provide feedback that encourages the learner within her appropriate "zone of proximal development" [71]. The second type of agents are *peer learning agents*, the explicit use of agents as interactive partners in the learning process itself [27, 58]. These agents are built into the user interface and, as with pedagogical agents, have knowledge of the user. While these agents may have teaching capabilities, they are typically less engineered for guiding learning overtly than pedagogical agents. The third type of agents are *demonstrating agents*, where the agents themselves are interactive mediums for learning; for example, interactive agent-based simulations [54, 73] or educational robotics [19, 65]. These agents embody the domain knowledge and are removed from the other components of the learning system. An open area for future work is the development of systems which combine this third type of agent with one or both of the others.

Many systems employ an underlying agent architecture that is not necessarily apparent to the user. Agents can be implemented as part of any system component, but are particularly useful for the control component and providing a means for system adaptivity (e.g., [63, 67]). This is a natural approach as the use of independent modules that have specific tasks to perform while being able to interact with other independent modules fits nicely with the agent paradigm and with the principles of high cohesion and low coupling in software engineering. An example is the Dynamic Adaptive Learning study [67] that uses an agent-based architecture involving a student agent, evaluation agent, record agent, learning object agent and modeling agent. In this architecture, the agents cooperate to determine the appropriate learning objects to be presented to the student.

This paper focuses on discussing the three primary types of agents in use by learning systems today. Thus the work is organized as follows: section 2 describes the use of pedagogical agents; section 3 discusses ways in which agents act as peers in learning envionments; and section 4 highlights examples of environments in which students learn about various phenomena by interacting with agents that demonstrate aspects of the phenomena. Section 5 delves into the topic of testing and evaluation, briefly outlining key components of assessment in human learning and evaluation of computer-aided learning systems. Finally, we close by discussing current open issues.

2. PEDAGOGICAL AGENTS

Much like a narrator in a movie who provides voice-over to explain scenes but never actually appears in the film, a *pedagogical agent* will "pop up" when the learner indicates that she needs help. There are three methods used to determine when this help is needed: (1) directly, upon request by the user, for example by clicking a "help" button; (2) indirectly, by the system monitoring the learner's performance and automatically detecting when she seems to need assistance; and (3) mixed initiative, which relies on a combination of (1) and (2).

The work of Lucas *et al.* [35] involves the use of pedagogical agents that assist learners when the agent perceives that the user needs help, based on a model of the learner and the learning goals of the system. Animated Pedagogical Agents (APA)s are used to assist students in distance learning environments. The initial project involves popping up and providing guidance in the use of a calculator. The APAs work in conjunction with the initial loader (instantiated by the FIPA-OS Agent-Loader) and a graphical interface agent called the CalcAgent for handling the display, input and output to the calculator. Mbalad and Anyouzoa [46] have also used pedagogical agents to support online distance learning.

Johnson *et al.* [25] present comprehensive research on animated pedagogical agents, highlighting issues that are faced when developing computer-based learning companions. Questions are addressed such as when to initiate dialog, as above, as well as how to include non-verbal communication and how to design for a wide range of client technologies. Animated agents can mimic human gestures and emotion-driven expressions, which can help engage students more than static avatars. Several agents were developed and tested, including: Steve (Soar Training Expert for Virtual Environments), Adele (Agent for Distance Learning, Light Edition) and Herman the Bug. Steve is embedded in an immersive three-dimensional environment that includes sound and was designed for naval training. Adele was built for use by university students over the internet, using standard workstation platforms and web technology. Herman the Bug is a "talkative" agent that gives advice to young students about plants, while teaching about botany. Animated agents can provide a number of advantages, such as navigational guidance, interactive demonstrations, gesturing for attention and use of other non-verbal communication devices. Empirical results using animated pedagical agents are positive, with some cautions. For example, there is evidence that young students may be distracted from the learning task by the agent's interface; however, a well-designed agent can keep students on-task and avoid this pitfall.

More recent work in this realm has focused on *interactive pedagogical drama* [41], where animated pedagogical agents become actors in a pseudo theatrical environment and learners either become immersed as participants in the drama or act as observers, like members of an audience. There are advantages to each approach; the former, active approach, requires learners to act in the drama which can be challenging and motivating, while the latter gives learners an opportunity to participate passively, with time for reflection and perhaps impartial analysis. Johnson [24] and Marsella *et al.* [41] use an interactive pedagogical drama for a system called "Carmen's Bright IDEAS" in which an adult human is guided through scenarios designed to improve problem-solving skills. Carmen, a character in the drama, is the mother of a pediatric cancer patient. She has a job and has another (healthy) child to take care of. When using the system, the learner observes Carmen's thoughts and can choose actions for her, in sessions with a counselor, discussions with her child's doctor, interactions with her boss, and so on. The pedagogical goal is for the human user of the system to improve her problem-solving skills and gain insight into similar situations in her own life.

The underlying architecture of Carmen's Bright IDEAS has been developed into a generalized framework called *Thespian* [62, 59] and applied to other domains. The Tactical Language Training System (TactLang) is a military application in which the learner engages in role-playing activities to acquire knowledge of the language, idiom and customs of particular geographic regions. One example is a drama which unfolds in a village cafe in the Middle East and the learner interacts with characters who are speaking Arabic. The system architecture integrates *PsychSim*, a multiagent system for mental modeling [52], with a story-line and basic script as well as pedagogical goals and social norms to guide agent-agent and agent-human interaction. The premise is that such an architecture will aid in rapid deployment of interactive pedagogical dramas.

3. PEER LEARNING AGENTS

There are many learning environments where agents interact with human learners as peers. These agents appear less intrusive than pedagogical agents. Many agent-based learning systems leverage game technology to provide both motivation and a situated, simulated training environment [31]. Human learners engage with agents as opponents or partners. These agents act more like peers than pedagagical agents who are more like tutors or instructors. The primary issues in developing peer learning agents involve (1) developing believable situated environments in which the human learner and agent peer interact, and (2) providing believable agents and a natural mode of interaction.

To address the issue of developing situated environments, which can become very expensive, many projects take advantage of a number of free or off-the-shelf game engines. For example, the TactLang mission environment (mentioned above) employs a modified version of the game engine Unreal Tournament, known as Gamebots [1]. Many games include *simulation elements* [3] that provide practical experience and *game elements* that offer an environment of engagement, discovery and competition [6]. Simulations can be particularly useful to provide training in several categories of skill sets: internalising processes, understanding systems, decision-making, perspective-shifting, team-building and cooperation [16].

Further, Wilkinson [74] states: "optimal learning occurs in the simulation of real world, problem-based activities. This happens in a safe environment where errors are expected, and failure deepens learning experience". The notion of a safe environment is particularly critical for the Mission Rehearsal Exercise (MRE) project [69]. Safety is also a key consideration in the training simulation being developed by Richards *et al.* [55] where agents are created in a game environment to allow the user to explore various risk scenarios. This project is focused on addressing issues concerning the agents' acquisition and reuse of knowledge and the language, cognitive and behavioural abilities of the agents to provide a more believable, engaging and immersive learning environment.

A number of game engines are available, with various strengths. Well-known engines include: Torque, Doom, Unreal Tournament, Aurora, Hammer, Embryo and Quake. Only some, however, are complete systems of high calibre, such as Hammer 4 and Doom 3. Simulation games have also been used in business environments, for example in teaching administrative skills. Off-the-shelf game simulations such as Doom II have been used in conjunction with free tools downloaded from the internet to provide cost-effective military training, for example, where real-world environments or locations may be unavailable to troops. The Gamebots project, mentioned earlier, started at the University of Southern California's Information Sciences Institute and is a prime example of the use of a game engine for agent-based training that was created for wider applications in artificial intelligence research. A good overview and lists of game engines is provided by Isakovic [21]. A shorter discussion is provided in Barles *et al.* [5].

To address the issue of providing believable agents, one strategy currently being explored by many comes from examining human characteristics such as emotion and empathy and extending agents to handle and even emulate such traits. Selvarajah and Richards [60] have developed an agent-based emotion architecture to be used for psychological testing of subjects in a three-dimensional virtual reality Cocktail Party World. The study compared an avatar and an agent world to see whether agents reacting to the emotions of the subject and other agents in their vicinity was more realistic than the avatar world in which emotions were scripted. It was found that the agent world more deeply affected the subjects than the avatar world.

Similarly, van den Broek [70] developed empathic agents that mimic human empathy in a study concerning stress levels in individuals, where stress was detected through analysis of human voice recordings. In both of these studies, the goal was not to use agents to teach the subjects, like pedagogical agents would, but rather to add human abilities to the agent software so that the results would be more sociologically valid, a key concern and common shortcoming of laboratory style testing with humans. Also related to better understanding humans is the work of Sehaba and Estrailier [58], who use an agent approach to help rehabilitate children with autism. A multiagent system is used to model the knowledge of the therapists, the child's profile and the dynamics of the interaction.

A key difference between agent systems for human learning and other agent systems is the need for human communication languages as well as agent communication languages (ACL). In general, it is unrealistic to expect a human to communicate with a software agent using a FIPA-compliant ACL. The burden is on the system developer to create a communication channel that bridges the human-agent gap. It is common to use a personal agent to bridge this gap. Lazzari *et al.* [32] use personal agents in their Remote Assistant for Programmers (RAP) system to allow human users to interact with other humans and other parts of the system, which are represented by other types of agents. Personal agents are created for each online and offline user. Personal agents handle a range of tasks such as selecting answer types, submitting queries, finding answers, finding experts, receiving expert ratings, selecting experts, receiving answers and rating answers. Some of these tasks are performed in conjunction with other agents in the system. As in the case of RAP, personal agents are tailored to the particular human user based on a user profile, stereotype or model and often require a high degree of sophistication. This is clearly true in the case of the User Observation Agent (UOA) used in the studies of the behaviour of autistic children by Sehaba and Estrailier [58]. This agent takes input from a software/hardware system known as FaceLab to capture features of the face and the orientation of the gaze. In addition, the UOA takes into account actions with the mouse, touch screen and keyboard.

In a similar vein, Kim [27] uses agents as learning companions. The idea is motivated by the pedagogical strategy of providing a learner with peer support. This work reveals that the competency of the learner has a large impact on the nature of the interaction with agents as learning companions, referred to as PALs. Strong students, identified by their GPA[1], preferred for the PALs to take a leading role and expected to be given correct advice. Weak students preferred to control the PAL and asked for assistance only when they wanted it and were happy with some wrong answers, as they found that a PAL that was always correct was intimidating. These interesting findings emphasize the varied influence of agents for human-based learning. Not only will differences in settings and interfaces affect learners, but also the personalities of the learners themselves will be a factor. Refer to section 5 for discussion of issues relating to assessment of agent-based learning systems.

4. DEMONSTRATING AGENTS

The notion of agency is useful for teaching and demonstrating a wide range of phenomena in the world. From introductory programming concepts [7] to dynamic systems [73], innovative researchers and teachers have developed virtual and embodied agents that interact with students of all ages. These environments take advantage of the popular and proven *constructionist* [49] pedagogical paradigm, motivating students and helping them to learn by doing. Here we highlight two complementary directions within this paradigm that concentrate on the use of agency to provide meaningful learning experiences: multiagent simulation and educational robotics.

Multiagent simulation allows students to program agents using simple commands and view graphically, and instantly, the effect of their code. This not only teaches students about programming concepts, but also provides powerful lessons in modeling. Students observe the world, invent rules about it, program them and analyse how well their rules represent the phenomena in the world they are attempting to model. Sengupta and Wilensky [61] use the NetLogo [48] agent simulation package to assist physics students to better understand the field of electromagnetics at the micro level. By modeling concepts such as electrons and atoms as agents, students are able to discover for themselves the emergent phenomena and learn to predict the behaviours in a way that produces deeper learning and understanding. The NetLogo Investigations in Electromagnetics (NIELS) studies demonstrate how learning in the domain can be broken down: "thinking in levels using multi-agent based models allows the students to establish concrete relationships between submicroscopic objects (e.g., electrons) which are shrouded in mathematical equations in traditional Physics instruction."

Blikstein and Wilensky [8] also employed the NetLogo environment, but this time to understand the difficult concepts involved in Materials Science. The MaterialSim system is a modeling-for-understanding framework that uses multiagent modeling languages where each agent is a basic computational construct with simple rules which control their behaviour and from which more complex higher level behaviours emerge. The study included classroom observations, pre/post interviews and data analysis of the usage session.

The simulation system Swarm [14, 68], based on the simple notion of ants in a nest gathering sugar, has been used to model, test and refine complex economic and language theories. Examples of other multiagent simulation environments include AgentSheets [2, 54] and RePast [53]. While the nature and architecture of environmental agents differs significantly from that of pedagogical agents, the use of modeling and simulation as a learning medium is shared by other applications. Examples are the training for risk situations project [55], MRE [69] and related projects.

Educational robotics refers to the use of robots in classrooms to teach a wide variety of topics, not necessarily robotics in particular. With the advent of the LEGO Mind-

[1] Grade point average

storms Robotics Invention Kit [47] in the late 1990's, today robots are being used all over the world to engage students from early primary through undergraduate classrooms [7, 65]. While some courses focus more on the hardware and engineering design aspects of robotics [42, 43], others concentrate on control mechanisms, agency, behavior-based paradigms and multiagent systems [44, 50, 76]. Many include some type of contest in the course to help motivate students as well as take advantage of fully tested and readily available environments, such as Botball [9] and RoboCup [56, 65]. A broad range of experience reports have been published detaling lessons learned using robotics with younger students, in primary and secondary school classrooms and after-school programs [19].

As well as hands-on, hardware-based approaches, a number of simulators have been developed recently to give students who do not have access to robot hardware an opportunity to explore the concepts behind controlling robots or to speed up development by providing a rapid-prototyping environment where debugging can occur more quickly than on real robots. Chu *et al.* [12] have developed an agent-based simulation environment for children, designed to be used in conjunction with the popular RoboLab [57] graphical programming interface and LEGO Mindstorms robot. The motivation is to give students an opportunity to learn about agent-based programming by using RoboLab in a "safe and friendly" place – they can "try out" programs in the simulator before loading them onto the robot platform and before being faced with real-world, physical constraints and issues such as noise.

When dealing with varying robotic platforms, even with a relatively simple one like the LEGO Mindstorms, it is effective for students to be able to classify and specify behavior patterns for their robots. Behavior patterns can range from basic behaviors, such as "move forward for 4 seconds", to complex behaviors such as "open a gate". Due to the complexity of defining and implementing robot behaviors, many control architectures are designed to build complex behaviors out of simple, low-level commands [45]. Another approach by Goldman [18] provides a custom behavior-based interface to RoboLab and an XML-based translator for downloading on to the Sony AIBO robot.

5. TESTING AND EVALUATION

One of the more prominent issues that separates human learning systems from other agent-based and interactive system development is the aspect of testing and evaluation. It is expected that not only the software learning environment be fully debugged and tested, but also, particularly amongst education researchers, the system must be evaluated with respect to its effectiveness as a learning environment. This section provides a brief description of evaluation in this context.

While there is no fixed standard for evaluating the effectiveness of interactive learning systems, there are two generally accepted categories of assessment [34, 40]:

- *formative assessment* tests the design and behavior of a system in-progress, generally performed by computer scientists, system designers and builders; and
- *summative assessment* evaluates the effectiveness of a completed system, generally performed by educators and/or psychologists.

Researchers begin by identifying what is being evaluated. Design and performance aspects need to be examined differently. The nature of the testing will vary depending on whether the goal is to assess the theoretical basis underlying the system or the software components themselves.

Each of a learning system's components (see Figure 1a) can be evaluated individually. Domain knowledge should be checked for accuracy and coverage. The teaching component can be evaluated for the range of instructional method(s) offered, its level of adaptability and the degree to which its instruction is based on proven educational and psychological methods. The user interface can be examined by comparing multiple user interfaces for the same underlying engine and looking, in particular, for improvement in student learning. System adaptivity can be compared using interactions at different skill levels. The control component can be evaluated using various system performance measures, such as speed. Finally, and probably the most important, improvement in student knowledge (i.e., learning) can be measured using the same criteria in a computer-based environment that are employed within standard educational and/or psychological testing. These include: (1) *validity* — does the test show evidence that it measures what it says it measures? (2) *reliability* — are multiple results for the same subject consistent? (3) *objectivity* — is the test administered and scored the same way for every participant? (4) *standardisation* — can results be translated into a meaningful representation of student performance?

The techniques for performing assessments vary depending on which component is being evaluated, where in the system development cycle the evaluation is being performed and who is performing the evaluation. Similar to typical HCI lifecycle models, such as the star life cycle [51] that involves evaluation after each step, evaluation of learning systems typically follows an iterative cycle. Beginning with *system development* and extending through to *experimental research*, steps may be revisited at any time during the formative phases of system development. Once summative assessment begins, in the experimental research phase, the system cannot change; otherwise, the summative results will be invalid. *Pilot testing* often occurs late during formative assessment, bridging the gap to summative assessment. There are three methods of pilot testing: (1) *one-to-one*, which is performed early in the development cycle, with one student, instructor, trainer or researcher providing feedback; (2) *small-group*, which is performed later in the development cycle, with a small group of students, instructors or trainers providing feedback; and (3) *field*, which is performed near the end of development, emulating experimental conditions with teachers or trainers and students in a "live" (i.e., classroom) setting.

Other techniques are more pertinent during summative assessment. In *criterion-based evaluation*, a general list of guidelines is developed and systems are evaluated based on their adherence to these guidelines, for example, program construction, behavior and characteristics. While developing relevant criteria is not an easy task, this method may prove useful in formative assessment and in comparing different systems. With *expert knowledge and behavior* assessment, system performance is compared to that of a human expert performing the same task. Software systems may be subjected to a standard *certification* process, through careful examination by qualified human experts. In *sensitivity* *analysis*, the responsiveness of a system is tested on a variety of different user behaviors. This may be particularly useful for evaluating system adaptivity. After system development and pilot testing are complete, *experimental research* can begin. The conditions should be the same as those during the field testing phase.

Two mechanisms for collecting evaluation data are common:

- *quantitative*, in which numerical data is analysed, frequently by comparing scores on pre- and post-tests and surveys, to measure changes in student performance and attitudes; and

- *qualitative*, in which interviews and surveys are conducted and observations are made.

Mixed methods research [22] combines the two, but typically, at least in the education arena, researchers tend to adhere to the methods of one category or the other. Quantitative methods rely on standards type testing, with multiple choice style questioning and Likert scale surveys. System logs are also examined. Qualitative, or "open", methods encompass data taken in both written and oral forms, as part of interviews, questionaires and open surveys containing short-answer questions (rather than multiple choice). Transcripts are "coded" and analysed based on measures such as frequency of broad term usage, often borrowing techniques from natural language processing in order to compute semantic similarity between answers.

Reviewing the literature describing agent-based systems issues, written from an agents research perspective (as opposed to an education research perspective), the primary type of evaluation is formative assessment, particularly testing the accuracy and functionality of system design and early pilot testing. Within the sampling of literature referenced earlier in this article, a wide range of test environments have been used, including formal primary, secondary and undergraduate classrooms, research laboratories, industrial workplaces, military bases and distance learning (or "e-learning") settings. As well, there are broad differences in the maturity of systems presented. Some have completed only the design phase while others have been fully implemented. Many have undergone some aspects of formative testing, including architecture and system design reviews, and user interface studies. Few have reached the summative testing phases, but some have conducted focus groups and pilot studies.

Many systems which are designed do not get past the prototype phase for various reasons. Firstly, gaining access to human subjects, particularly minors, may be difficult. Hurdles can include: availability of representative population; costs for setup and recruitment and human ethics requirements. One of the more serious issues is that learning systems take so long to develop, by the time they are operational, the customer does not need or want them any more. This is frequently due to the large costs (in time and money) associated with building a teaching component into a system, which adds to the typically unwieldy costs of developing software on time and within budget [10]. Nonetheless, just as one would expect user evaluation to be found in research publications concerning user interfaces, there is a higher expectation in the work reported on agent-based human learning systems that an evaluation with real users will be conducted, will be well designed and analysed carefully.

6. DISCUSSION

The types of research, application development and studies being performed using agent technology for human learning are varied. In some cases, the central theme of the project is to explore and extend current agent technologies and capabilities. In other cases, agent theory is not being developed but rather is being applied and the research focuses on the application.

Differences in learners' ages and genders must be considered when designing agents that will interact as pedagogical tutors or peer learners. While most learning systems using games are focused on making the learning more palatable for children, the motivation of Richards *et al.* [55] is quite different and in keeping with Kearsley [26] who has found that "instruction for adults needs to focus more on the process and less on the content being taught. Strategies such as case studies, role-playing, simulations and self-evaluation are most useful. Instructors adopt a role of facilitator or resource rather than lecturer or grader." Using these virtual environment technologies in conjunction with agent components allows production of less expensive and more accessible systems that offer increased control of the environment together with increased relevance to the real world.

Many studies have highlighted differences in the way females and males approach, interact with and think about technology. Inkpen [20] found gender differences in the way children approach game environments. Girls tended to perform better when another person was also playing on the same machine, but boys performed better when the other player was on a different machine. Girls were found to have less physical contact with their human partner or the mouse compared to boys. Brunner *et al.* [11] showed that females tend to view technology as a tool used to facilitate human interaction, whereas males tend to view technology as an object that can be used to extend their abilities and/or power. These gender-based attitudinal differences will effect a student's experience with a learning environment; as well, these issues generalize to any interactive agent-based system.

A more philosophical question addresses the intersection of agent-based technologies and interactive learning systems. What does each do for the other? Agent-based systems, since the infant stages of the field [36, 75], have been highly focused on finding effective, modular means to intelligently interact with human users in a personalized, but not annoying way (go away, Mr Clippy). The notion of an automated assistant is quite natural in a learning or tutoring environment. The difficulty comes in building generalizable systems because effective tutoring, by both humans and agents, involves deep understanding of the knowledge domain being taught. Knowledge engineering, identifying and fixing common bugs in students' learning paths and constructing tutoring agents that reflect the experience of a master teacher is a daunting endeavor, shadowed by the desire to avoid repeating the mistakes of expert systems. Agents have been shown to learn to adapt in dynamic environments, such as robotic soccer [66] and electronic markets [72]. If we view the human learner as an agent's changing environment, perhaps tomorrow's solutions will include agents that can learn to teach.

Acknowledgments

This work was partially supported by NSF #ITR-02-19347.

7. REFERENCES

[1] R. Adobbati, A. N. Marshall, A. Scholer, S. Tejada, G. A. Kaminka, S. Schaffer, and C. Sollitto. Gamebots: A 3D Virtual World Test-Bed for Multi-Agent Research. In *Second International Workshop on Infrastructure for Agents, MAS, and Scalable MAS*, 2001.

[2] http://agentsheets.com/.

[3] C. Aldrich. *Simulations and the Future of Learning : An Innovative (and Perhaps Revolutionary) Approach to E-Learning*. Pfeiffer, San Francisco, 2003.

[4] M. Balabanović. *Learning to Surf: Multiagent Systems for Adaptive Web Page Recomendation*. PhD thesis, Stanford University, 1998.

[5] J. Barles, M. Dras, M. Kavakli, D. Richards, and A. Tychsen. An overview of training simulation research and systems. In *Agent-based Systems for Human Learning, AAMAS Workshop*, 2005.

[6] R. Bartles. *Designing Virtual Worlds*. New Riders, Indianapolis, IN, 2003.

[7] D. S. Blank, D. Kumar, L. Meeden, and H. Yanco. Pyro: A Python-based Versatile Programming Environment for Teaching Robotics. *Journal on Educational Resources in Computing (JERIC), Special Issue on robotics in undergraduate education, part I*, 4(2), 2005.

[8] P. Blikstein and U. Wilensky. Less is more: Agent-based simulation as a powerful learning tool in materials science. In *Agent-based Systems for Human Learning, AAMAS Workshop*, 2005.

[9] http://www.botball.org.

[10] F. P. Brooks. *The mythical man-month: essays on software engineering*. Addison-Wesley, 1975.

[11] C. Brunner, D. Bennett, and M. Honey. Girls' games and technological desire. *From Barbie to Mortal Kombat: Gender and Computer Games*, 1998.

[12] K.-H. Chu, R. Goldman, and E. Sklar. Roboxap: an agent-based educational robotics simulator. In *Agent-based Systems for Human Learning, AAMAS Workshop*, 2005.

[13] A. Cypher. Eager: Programming repetitive tasks by example. In *Computer-Human Interaction*, 1991.

[14] J. M. Epstein and R. Axtell. *Growing artificial societies: social science from the bottom up*. The Brookings Institution, Washington, DC, USA, 1996.

[15] L. Foner. Yenta: A multi-agent referral based matchmaking system. In *First International Conference on Autonomous Agents*, 1997.

[16] L. Galarneau. The elearning edge: Leveraging interactive technologies in the design of engaging, effective learning experiences. In *e-Fest 2004*, 2004.

[17] D. Goldberg, D. Nichols, B. M. Oki, and D. Terry. Using collaborative filtering to weave and information tapestry. *Communications of the ACM*, 35(12), 1992.

[18] R. Goldman. From RoboLab to AIBO: Capturing Agent Behavior. Master's thesis, Columbia University, 2005.

[19] R. Goldman, A. Eguchi, and E. Sklar. Using Educational Robotics to Engage Inner-City Students with Technology. In *Sixth International Conference of the Learning Sciences*, 2004.

[20] K. Inkpen. Gender differences in an electronic games environment. In T. Ottman and I. Tomek, editors, *ED-MEDIA 94 World Conference on Educational Multimedia and Hypermedia*, 1994.

[21] http://cg.cs.tu-berlin.de/~ki/engines.html.

[22] R. B. Johnson and A. J. Onwuegbuzie. Mixed methods research: a research paradigm whose time has come. *Educational Researcher*, 33(7), October 2004.

[23] W. Johnson. Pedagogical agents for virtual learning environments. In *International Conference on Computers in Education*, 1995.

[24] W. L. Johnson. Pedagogical Agent Research at CARTE. *AI Magazine*, Winter, 2001.

[25] W. L. Johnson, J. W. Rickel, and J. C. Lester. Animated Pedagogical Agents: Face-to-Face Interaction in Interactive Learning Environments. *International Journal of Artificial Intelligence in Education*, 11, 2000.

[26] G. Kearsley. Explorations in learning and instruction: The theory into practice database. http://tip.psychology.org/index.html, 2004.

[27] Y. Kim. Agent learning companions: Learners expectations of the desirable characteristics. In *Agent-based Systems for Human Learning, AAMAS Workshop*, 2005.

[28] D. Kuokka and L. Harada. Matchmaking for information agents. In M. Huhns and M. Singh, editors, *Readings in Agents*. Morgan Kaufman, 1997.

[29] K. Lang. Newsweeder: Learning to filter news. In *Twelfth International Conference on Machine Learning*, 1995.

[30] Y. Lashkari, M. Metral, and P. Maes. Collaborative interface agents. In *Twelfth National Conference on Artificial Intelligence*. AAAI Press, 1994.

[31] J. Lave and E. Wenger. *Situated learning: Legitimate peripheral participation*. Cambridge University Press, Cambridge, England, 1991.

[32] L. Lazzari, M. Mari, A. Negri, and A. Poggi. Support remote software development in an open distributed community. In *Agent-based Systems for Human Learning, AAMAS Workshop*, 2005.

[33] H. Lieberman. Letizia: An agent that assists web browsing. In *International Joint Conference on Artificial Intelligence*, 1995.

[34] D. Littman and E. Soloway. Evaluating itss: the cognitive science perspective. In M. C. Polson and J. J. Richardson, editors, *Foundations of Intelligent Tutoring Systems*. Lawrence Erlbaum Associates, Hillsdale, NJ, 1988.

[35] J. P. Lucas, B. Wilges, and R. A. Silveira. Inserting animated pedagogical agents inside distributed learning environments by means of fipa specifications. In *Agent-based Systems for Human Learning, AAMAS Workshop*, 2005.

[36] P. Maes. Modeling Adaptive Autonomous Agents. *Artificial Life: An Overview*, 1995.

[37] T. Malone. Toward a theory of intrinsically motivating instruction. *Cognitive Science*, 4, 1981.

[38] T. Malone. What makes computer games fun? *Byte*, December, 1981.

[39] M. A. Mark and J. E. Greer. Evaluation

methodologies for intelligent tutoring systems. *Journal of Artificial Intelligence and Education*, 4, 1993.

[40] M. A. Mark and J. E. Greer. Evaluation methodologies for intelligent tutoring systems. *Journal of Artificial Intelligence and Education*, 4, 1993.

[41] S. Marsella, W. L. Johnson, and K. LaBore. Interactive Pedagogical Drama. In *Fourth International Conference on Autonomous Agents*, 2000.

[42] `http://fredm.www.media.mit.edu/people/fredm/projects/6270/`.

[43] F. Martin. *Robotic Explorations: A Hands-On Introduction to Engineering*. Prentice Hall, 2000.

[44] `http://www-scf.usc.edu/~csci445/`.

[45] M. J. Mataric. Behavior-based robotics as a tool for synthesis of artificial behavior and analysis of natural behavior. *Trends in Cognitive Science*, 2(3), March 1998.

[46] A. Mbala and A. G. N. Anyouzoa. A multi-agent system to support users in online distance learning. In *Agent-based Systems for Human Learning, AAMAS Workshop*, 2005.

[47] `http://www.legomindstorms.com/`.

[48] `http://ccl.northwestern.edu/netlogo/`.

[49] S. Papert. Situating constructionism. *Constructionism*, 1991.

[50] S. Parsons and E. Sklar. Teaching AI using LEGO Mindstorms. In *AAAI Spring Symposium 2004 on Accessible Hands-on Artificial Intelligence and Robotics Education*, 2004.

[51] J. Preece, Y. Rogers, H. Sharp, D. Benyon, S. Holland, and T. Carey. *Human-Computer Interaction*. Addison-Wesley, 1994.

[52] D. V. Pynadath and S. C. Marsella. PsychSim: Modeling Theory of Mind with Decision-Theoretic Agents. In *International Joint Conference on Artificial Intelligence*, 2005.

[53] `http://repast.sourceforge.net/`.

[54] A. Repenning and W. Citrin. Agentsheets: Applying grid-based spatial reasoning to human-computer interaction. In *IEEE Symposium on Visual Languages*, 24-27 1993.

[55] D. Richards, M. Kavakli, and M. Dras. Training for high risk situations. In *Agent-based Systems for Human Learning, AAMAS Workshop*, 2005.

[56] `http://www.robocup.org`.

[57] `http://www.ceeo.tufts.edu/robolabatceeo/`.

[58] K. Sehaba and P. Estraillier. A multi-agent system for rehabilitation of children with autism. In *Agent-based Systems for Human Learning, AAMAS Workshop*, 2005.

[59] M. Sei, S. C. Marsella, and D. V. Pyndath. Thespian: Using multi-agent fitting to craft interactive drama. In *International Conference on Autonomous Agents and Multi-Agent Systems*, 2005.

[60] K. Selvarajah and D. Richards. The use of emotions to create believable agents in a virtual environment. In *International Conference on Autonomous Agents and Multi-Agent Systems*, 2005.

[61] P. Sengupta and U. Wilensky. Niels: An emergent multi-agent based modeling environment for learning physics. In *Agent-based Systems for Human Learning, AAMAS Workshop*, 2005.

[62] M. Si, S. C. Marsella, and D. V. Pynadath. THESPIAN: An Architecture for Interactive Pedagogical Drama. In *International Conference on Artificial Intelligence in Education*, 2005.

[63] E. Sklar. *CEL: A Framework for Enabling an Internet Learning Community*. PhD thesis, Department of Computer Science, Brandeis University, 2000.

[64] E. Sklar. Agents for Education: When too much intelligence is a bad thing. In *Second International Joint Conference on Autonomous Agents and Multi-Agent Systems*, 2003.

[65] E. Sklar, S. Parsons, and P. Stone. Using RoboCup in university-level computer science education. *Journal on Educational Resources in Computing (JERIC), Special Issue on robotics in undergraduate education, part I*, 4(2), June 2004.

[66] P. Stone, G. Kuhlmann, M. E. Taylor, and Y. Liu. Keepaway Soccer: From Machine Learning Testbed to Benchmark. In *RoboCup-2005: Robot Soccer World Cup IX*. Springer Verlag, 2005.

[67] S. Sun, M. Joy, and N. Griffiths. An agent-based approach to dynamic adaptive learning. In *Agent-based Systems for Human Learning, AAMAS Workshop*, 2005.

[68] `http://www.swarm.org`.

[69] W. Swartout, R. Hill, J. Gratch, W. L. Johnson, C. Kyriakakis, C. LaBore, R. Lindheim, S. Marsella, D. Miraglia, B. Moore, J. Morie, J. Rickel, M. Thiebaux, L. Tuch, R. Whitney, and J. Douglas. Toward the holodeck: Integrating graphics, sound, character and story. In *Fifth International Conference on Autonomous Agents*, 2001.

[70] E. van den Broek. Empathic agent technology (eat). In *Agent-based Systems for Human Learning, AAMAS Workshop*, 2005.

[71] L. S. Vygotsky. *Mind in society: The development of higher psychological processes*. Harvard University Press, Cambridge, MA, 1978.

[72] V. Walia, A. Byde, and D. Cliff. Evolving market design in zero-intelligence trader markets. Technical report, Hewlett-Packard Research Laboratories, Bristol, England, 2003.

[73] U. Wilensky and M. Resnick. Thinking in levels: A dynamic systems perspective to making sense of the world. *Journal of Science Education and Technology*, 8(1), 1999.

[74] D. Wilkinson. The intersection of learning architecture and instructional design in e-learning. In *e-Technologies in Engineering Education: Learning Outcomes Providing Future Possibilities*, 2002.

[75] M. Wooldridge and N. R. Jennings. Intelligent Agents: Theory and Practice. *Knowledge Engineering Review*, 10(2), 1995.

[76] `http://www.cs.uml.edu/~holly/91.450/`.