Communication & Cognition
Vol. 42, Nr. 1 & 2 (2009), pp. 37 - 52

E-LEARNING IN VIRTUAL COMMUNITIES

L.J.M.Rothkrantz

Abstract

In recent years many systems for distance learning have been developed. Even though students have access to learning material any time and any place, current tools for e-learning still have their limitations. The main shortcoming, compared to real life learning is the limited opportunity for interaction between student, teacher and material. This lack of multimodal communication limits one’s sense of being a member of an academic community. Virtual communities offer a challenging opportunity to solve this shortcoming.

In this paper we report about a virtual community for e-learning using serious gaming technology. This virtual community is the first step to a virtual University, a virtual community for academic learning. The virtual community will be populated by agents representing students, teachers and campus employees. These agents are autonomous or controlled by the user. We focus on multimodal communication between agents realized by game technology. And interaction between agents and the virtual world realized by waypoints in the simulated world. The design and implementation of the communication model will be presented and the first test results of a user experiment.

Categories and Subject Descriptors

J.6 [Computer Aided Engineering]

General Terms

Management, Measurement, Performance, Design, Reliability, Experimentation, Human Factors, Theory.

Keywords

Virtual communities, e-learning, serious gaming.

1. Introduction

Academic education is very dynamic. In the course of the years many innovative teaching methods, educational models and theories have been introduced and in time replaced. In recent years there was a focus on mastery learning, drills and practice methods, discovery learning, project work, computer assisted instruction, and on many methods of distance learning. Lecturers have been using books for many years successfully but have had to switch to other teaching materials as well to maintain an appealing interaction that meets up to students' expectations.

At this moment we see a growing interest in virtual communities. Students spend a great deal of time playing games and maintaining social contacts via the internet. This trend challenges academic teachers to integrate new methods of communication in their education as well. This doesn't imply that the academic world has to follow all hypes but if students are becoming so accustomed to the use of virtual environments the academic community should recognize this and adopt new methods. The challenge is then to investigate how virtual environments can be integrated in academic education.

The paper is structured as follows: In the next section we discuss virtual communities, communication in those communities and learning. Next we present the design and implementation of our first prototype of the virtual world. And finally we end up with some user experiments.

2. Related work

2.1 Virtual communities

Currently there are many virtual communities. Very popular are the massive multiplayer online role-playing games such as “World of Warcraft” which attracts more than 9 million users worldwide. These games illustrate that it is technically possible to create an environment where many users take part in role playing games. As mentioned we will deploy a game engine with similar facilities.

The game engine enables us to create two types of inhabitants of the virtual world: agents that behave autonomously and avatars controlled by the users. The behavior of agents can be created by scripts, provided by the game engine. Unfortunately the user interface to create agents is not very human friendly. One of the project goals is to develop a multimodal Graphical User Interface (GUI) for this purpose.

2.2 Multimodal communication

There are many tools available for communication and interaction between agents. JADE (Bellifemine, Poggi, Rimassa, 2001, pp. 216-217) or COUGAAR software is most commonly used for agent communication. This way of communication is based on text based message passing. To realize a more human like communication, the keyboard-mouse interface should be replaced by camera and microphone. Nonverbal communication by facial expressions, gaze and other ways of body language and verbal communication by speech are more natural human like way of communication. Multimodal communication between agents in the virtual world can be realized by using game engines. Current game engines as DELTA3D or Second Life offer these facilities.

2.3 Distance learning

At this moment there are already many virtual communities, but to our knowledge no open academic virtual community exists. Our virtual community will offer another option of distance learning. In recent years many systems for distance learning have been developed. Most of those systems provide learning material via WWW. So students can have access to that material remotely at a time and place that suits them best. The interaction student-teacher takes place via e-mails or correspondence. The success of the Open University proves that there is need for distance learning for some students in our society. Some people are not able to enroll as regular students because of jobs, disabilities or other constraints. There is also need for cooperation between students. For that reason the Open University organizes group meetings on special time and date. But that violates the remote access.

Most of the learning material is based on written text and simulation software. At this moment many new materials have been developed using multimodal interfaces and multimedia. Real life recordings of lectures are distributed via web lectures.

Despite the advances that have been made, it has been proven that students miss the feeling of actual presence at the university. Students want to meet and interact with other students. In a MSN session for example it is visible who is online and participates in the discussion. But is unclear who is really involved. In virtual communities communicating agents group together, look to each other, and show by their body language if they want to communicate a message or want to leave the discussion. Visual control is very important in human communication. At this moment virtual communities for students are growing for purposes ranging from entertainment to academic activities. Most interaction takes place via chatting, e-mail, weblogs etc. But there is a start of multimodal communication.

3. Serious gaming

The term "serious games" refers to games which are not primarily designed for entertainment. Such serious games are being used for education, training and marketing. Serious games have a lot in common with interactive simulation. The idea of using serious gaming in education is not new, but the results up to now have not been overwhelmingly positive. A handicap has been a lack of tools to design games with a high level of user interaction, realism and complexity. But current massive multiplayer online games (MMOGs) offer opportunities for thousands of players to play simultaneously. This allows them to be involved in problem solving, social interaction and drills. Players can now join several communities, take on different identities or roles and practice different abilities and social skills.

The goal of this project is to use a game engine to set up a virtual academic world as a serious game for academic learning. Such a virtual world offers a lot of possibilities and challenges. At the moment there are already some learning environments for drill and practice. Some of them are used to train people in situations which are difficult or dangerous to generate in real life (explosions, terrorist attacks and fires). But our aim is to develop a learning environment in which users can develop and train higher order cognitive abilities such as critical, thinking, problem solving, strategic thinking and team building. We will implement different ways of communication. Human communication is regulated by rules which have to be learned. Agents can show interest by looking to each other. Talking agents look to each other. Agents not interested in communication look in another direction or go away. During lectures it is common used to ask permission to raise a question and not simple start talking. So these rules are context dependent. Cognitive and social skills will also be practiced. Players have to find partners to solve problems, set up teams and communities. The presented problems can’t be solved in an individual way, but only by teams.

The expectation is that learned and trained abilities can be used in real life because of the similarity between real and virtual life. One of the benefits of the virtual world is the similarity with the real world. On the other hand in the virtual world we are not limited by the constraints of the real world. In the design of our virtual University we would like to keep the benefits of the real University and remove the disadvantages.

As soon as habitants enter the virtual university they have the feeling of presence in a real world. They should be able to sense the environment in a multimodal way and get a feeling of context awareness. So in the virtual University habitants are able to show the same behavior as in the real world. So it is not necessary to learn new behavior but it is the behavior which is the result of life time learning.

From the other side the virtual University enables habitants to take different roles and identities. In different game communities participants take on different nicknames. After negative experiences they can start a new second life. Participants can try different roles. The contacts are based on the information provided. So the negative impact of cultural backgrounds, race and can be reduced.

In a virtual community events can be generated which are difficult to realize in real life. A challenging option for educational training is to mix real and virtual life. Real players can be represented as agents and get involved in situations and actions which are difficult to realize in real life. We mentioned already the crisis management module.

In a virtual campus different crisis can be generated. Different technologies, different communication tools and methods and different style of management and rescue operation can be researched and trained. Game technology developed in the framework of serious gaming can be used as technical environment to develop new educational methods and tools.

Students from different disciplinary have to play different roles ranging from rescue worker up to member of the management team.

At this moment there are many tools available to design a virtual community or to take part in a virtual community. We mention the following:

- VRML, introduced in 1992 (www.web3d.org), described by P. Anders in Envisioning Cyberspace-Designing 3D Electronic Spaces, McGraw-Hill.
- Blaxxun (www.blaxxun.com) was the first 3Dcommunity platform
- Alpha World (www.activeworlds.com/worlds/alphaworld), a world of avatars
- Second Life (secondlife.com), a virtual community with more than 4 million inhabitants.
- World of Warcraft (www.worldofwarcraft.com)
- Everquest (everquest.station.sony.com)
- Web 2.0 (mySpace, YouTube, Flickr) examples of community formation
- Delta3D (www.delta3d.org), open source game engine
- Half Life 2 SDK (half-life2.com), a commercial game engine

4. Simulation environment

As a proof of concept we developed a first prototype of a virtual community. This prototype is composed of a World Model, an event generator, which can generate disasters. The world is habitat by agents which are able to communicate in different ways (Benjamins, Rothkrantz, 2006).

4.1 Architecture

A global overview of the MACSIM system is given in Figure 3. These are all the components that can be distinguished from a global perspective.

- Simulation component: In this component a simulation of the concerned area can be given. The physical properties on a location (x,y,z) on time t can be read, and in this version of the program this means for instance fire, gas dispersion and explosions. For the gas dispersions formulae that are being used in commercial available software are used, the Gaussian Plume Model, to be exact.
- Event Generator: This event generator should be able to generate crisis events, which is an XML-based script. When the script unfolds, events are being launched and those events have their effect on the simulation

environment. Every scripts has some open slots such as starting time, intensity, duration. To launch a script the open slots have to be filled by values of parameters. These parameters can be defined by default by a predefined script of the simulation or by the result of action during the simulation.

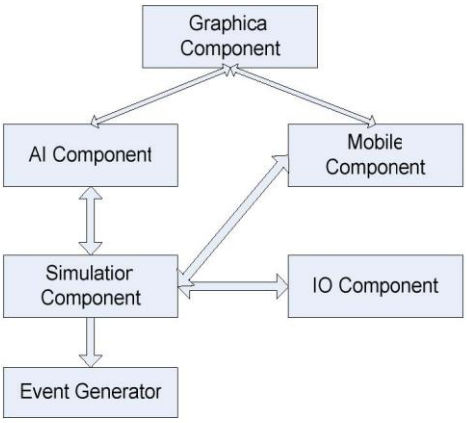

Figure 1: Component Archtecture

- Graphical component: The graphical component of this program will consist of several user interfaces. One user interface will be used to setup the scenario, and simulation parameters. Another Interface will be used as a representation of what is going on at the crisis center. It will also contain a graphical representation of the area that will show the incoming reports of the people that are currently in the simulation area.
- AI-component: This component consists of a knowledge based system that, based on decision rules that are derived from first-hand experience from experts and real-life complaints from people that smell gases, helps in deciding what probably is the most realistic scenario that is currently happening. This is of critical importance in the first stages of the development of a crisis, when not much information is known and a first hypothesis can be made through a knowledge-based system

4.2 Dataflow

The data flow between different components of IMACSIM (Benjamins, Rothkrantz, 2006) can be organized in a view as shown in Figure 1. In this view we can distinguish a Simulation Layer, Agent Middleware, an agent layer and one or more GUI's. It gives a clear overview of the flow of information and in which way it is being transferred to the components in the system.

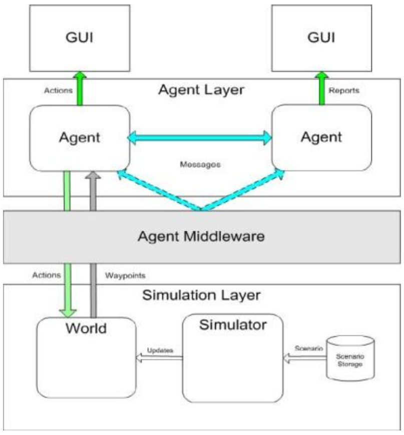

Figure 2: overview Communication

The Simulation Layer contains scenario storage. In this storage scenarios are being kept. A scenario can be acquired from the storage by the Simulator. The Simulator is in charge of simulating crises and for that it needs scenarios. Those scenarios are being transformed into a script internally by the simulator. A script is basically a timeline with a start time and an end time and certain events that can take place in the world in between. The simulator is processing those events and this usually means that as a result of a certain event the world is modified in one way or another, i.e. the simulator is updating.

This event generator should be able to generate crisis events, which is an XML-based script. When the script unfolds, events are being launched and those events have their effect on the simulation environment. Every script has some open slots such as starting time, intensity, duration. To launch a script the open slots have to be filled by values of parameters. These parameters can be defined by default by a predefined script of the simulation or by the result of action during the simulation. In case of a fire the fire script can be generated. But if a fireman flushes the fire the intensity of the fire should be reduced.

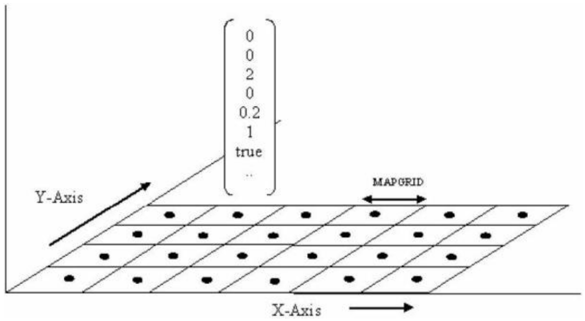

Figure 3: Waypoints

When the world is being updated, the agents in the world should be notified, because they have to sense changes in the world caused by the events. This is where the agent middleware comes into play. The agent middleware takes care that the agents in the simulation are receiving the updates of the world. The agents are receiving this information via agent middleware of because they are supposed to be autonomous. This means that the agents are supposed to work as independently as possible. Therefore other components should not have direct access to the agents, because that would imply some sort of ownership that does not fit inside the concept of independent agents. In the meantime, the agents are receiving data updates of the world in the form of waypoints. If these agents sense this data they can process and reason about it. Based on this reasoning the agents initiate actions. Those actions might have an effect on the world or not, but this is of course depending on the type of action that is the result of the agent's reasoning.

Besides reading waypoint data, the agents are also capable of sending messages to other agents. In the diagram the blue arrow indicates this, but it would be more accurate to connect the arrows via the agent middleware. This is because of the same reasons of agent independency. Those messages are being sent to other agents through the agent middleware as well. The agent actions that have an effect on the world are being propagated back again to the world to implement the changes. This requires a synchronicity scheme that ensures that the simulator applying the script to the world knows about the updates by agents, so it can update the world again according to the most recent changes.

Finally the agents are able to interact with the stream of events generated by the event generator. The users of IMACSIM will be able to view agent actions and play a certain part of an agent inside the scenario. This means that the agents will have to have some sort of GUI because otherwise the user will not notice their actions. If they have received or sent a report, then its GUI must also be showing those, for information purposes.

5. Communication

5.1 Interaction with the environment

Agent to agent communication is realized by message passing, implemented in JADE (Bellifemine, Poggi, Rimassa, 2001, pp. 216-217). The agent to environment communication takes place via waypoints. A waypoint is a vector of parameters representing the environment on a given time and place. Agents are able to sense the environment by reading the parameters in the nearest waypoint. Agents may also modify certain waypoint parameters as the enter it are undertake actions within the vicinity of the waypoint. As a simplified example, a certain waypoint may have parameters v1=0.6, v2=1, v3=2 representing density of smoke, a loud bang, number of agents in the area.

5.2 Communication modes

Our personal agents live in a virtual world. This virtual world is full of other agents, objects and ongoing events. Our agent is able to communicate with other agents and to interact with the environment in different ways:

5.2.1 Agent-agent communication

The agent-agent communication is defined and implemented by the agent framework JADE. Using this framework, agents can send messages to each other. Usually agents communicate by text messages. This is language dependent and for semantic understanding an advanced parser is necessary.

To reduce the ambiguity of NLP we developed an icon language (Fitrianie, Rothkrantz, 2007. Every message is composed of strings of icons. And every icon represents a semantic concept. The set of icons is context dependent. A special grammar controls the grammatical correctness of the sentences.

The biggest learning challenge is to design agents which are able to communicate automatically. The communication model is based on the well known Eliza-model (Weizenbaum, 1966, pp. 36-45)

In our case special key-icons or templates of strings of icons have to be recognized and connected to predefined answer strings. In our first prototype we have implemented a crisis scenario. There is an explosion in a chemical plant and a toxic cloud is spread over the area. Students are requested to define agents which are able to report about the crisis. They have to define simple sensor agents, which report about the composite of the air. More complex agents are human observers and agents in the control room. The last agents should be able to start a dialogue and reason about what is going on. The reasoning is based on scripts and discussed in another section.

5.2.2 Agent-environment communication

The communication agent-environment is realized via waypoints. The world is covered by waypoints. A vector of parameters is attached to every waypoint. Events are represented by values of these parameters. The intensity of fire or smoke, or a big bang or flashlight at a specific point (x,y,t) is represented by the parameters vi of the vector V(x,y,t). So our agents are able to sense the environment in a multimodal way, by reading the parameters in the nearby waypoint. Information about the static world, i.e. the location of buildings and other objects is also represented via the waypoints.

An important interaction is the sensation of moving objects, i.e. other agents. At every timestamp the information at the waypoints will be refreshed. So at some points a moving object can be detected. Not only the presence but also some characteristics of an object will be represented at the waypoints. Interesting information of an agent is some personal characteristics as sex, age, body movements such as running, walking but very important information about body language such as posture, facial expressions and gestures. In a graphical world this information will be represented so that it can be observed by human observers. But our virtual, autonomous agents are able to sense the environment via the waypoints. As we can see in Fig 3 every waypoint has some areas. As soon as an agent enters an area some information will be stored in the vector attached to those waypoints. This enables other agents to sense this information.

6. Experiments

In our experiments we focused on two topics communication and reasoning.

6.1 Communication

In the first prototype focused on nonverbal communication. Students have to define different scripts and scenarios for nonverbal communication. These scripts are dependent of the context and role students want to play. A first example is the “Hello” script. Every agent has a personal space. Agents are able to sense each other via waypoints in the intersection of the personal space. Every agent stores some information in that waypoint, i.e. his identity, possible goal and requests and the other agent is able to read that and to generate (automatically) an appropriate answer. In the “hello”-script meeting agents say hello to each other and that is all. If there is a request for social interaction the other agent can refuse that or accept that. In the last case agents can start sending messages to each. Of course the interaction is context dependent. During a lecture it is inappropriate to start a social interaction between agents in the role of students. But the cafeteria offers opportunities to start social interaction. Different scripts are activated by information in the waypoints.

6.2 Reasoning

The decision making process is distributed among autonomous agents. The reasoning mechanism in each agent is modeled as shown in Figure 2. This mechanism is based on a knowledge based system with a number of predefined prototype scenarios and a process that continually tries to recognize the most plausible scenario as more data arrives. This approach is similar to a well established and commonly adopted training methodology in crisis response, in which an instructor selects desired training objectives, and a crisis simulator (a computer program) automatically constructs a scenario, requiring application of the desired skills, based on a series of crisis events (Schank, Abelson, 1977). Subsequently, crisis workers are trained to recognize the type of crisis and to take appropriate actions by (gradually) identifying the crisis events that characterize the scenario (Stern, Sundelius, 2002, pp. 71-78).

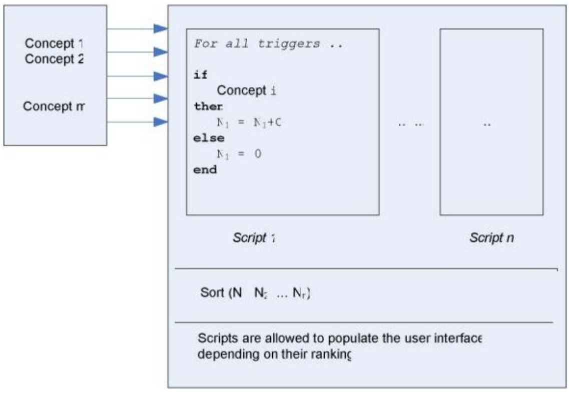

Figure 4: Calculation of the most plausible script

In the knowledge base, we represent a crisis scenario as a chronological ordering of characteristic concepts of the crisis situation (e.g. observations, actions), called a script. Our model contains a number of scripts, each representing a different scenario. The goal of the agent is to figure out the script (and thus, the scenario) that is most plausible given the information received thus far from the information channels available from the crisis simulator.

A script that does not contain any concepts that contradict with the information received so far by the agent is potentially the real scenario and is called a hypothesis. At the start of a real disaster the agent has several hypotheses available. These hypotheses are represented as frames or sets of properties and rules and actions. Each moment that new knowledge arrives, the scores for each different frame are placed on a scoreboard. The frame that is triggered the most is the frame or hypothetical scenario that is chosen as most probable scenario. This means that also information could come in that is conflicting with the hypothesis, but as long as the information rejecting the current hypothesis is not convincing enough, the current hypothesis is maintained. If there is enough conflicting information to reject the current hypothesis, then another more probable hypothesis frame is chosen.

Therefore a different set of knowledge will become available and this means also different actions that will be performed by that specific agent.

In the experiment students have to define several scripts. In every scripts they have to define the events and the if-then rules. The event generator will generate rules according to some scripts hidden by the students. They can verify if they recognize the right scripts or have to adapt the events or rules.

7. Conclusions

We developed a first prototype of a serious game or interactive simulation. The virtual environment, including a world with waypoints, a communication system and an event generator. The event generator is able to generate a crisis. Agents in the world can sense their environment via way[points and are able to communicate in a multimodal way. We designed human controlled agents and autonomous agents controlled by scripts. We defined nonverbal behavior of the agents via rule based scripts.

Next future we will select a game engine to realize a graphical visualization of our world. In our first experiments we realized different way of communication between agents

8. References

Bellifemine, F., Poggi, A., and Rimassa, G., (2001). *Jade: a fipa2000 compliant agent development environment*, pages 216-217.

Benjamins, T., Rothkrantz, L.J.M., (2006). *MACSIM: Serious gaming in crisis management via script-based simulation*, CGAMES 2006. October.

Friedman-Hill., E., (2003). *Jess in Action. Manning.*

Fitrianie, S., Rothkrantz, L.J.M., (2007). "A visual communication language for crisis management," *International Journal of intelligent Control and Systems*.

Klapwijk, P., Rothkrantz, L.J.M., (2006). "Topology Based Infrastructure for crisis situations", *Third International Conference on Information Systems for Crisis Response and Management ISCRAM 2006*, May.

Schank, R., and Abelson, R., (1977). Scripts, Plans, Goals and Understanding, Hilsdale, NJ: Erlbaum.

Stern, E. and Sundelius, B., (2002). Crisis Management in Europe: An Integrated Regional Research and Training Program International Studies perspective 3 (1), 71-78.

Weizenbaum, J., (1966). “A computerprogram for the study of natural language communication between man and machine”, *Communication of the ACM* 9(1) pp36-45.

About the author

L.J.M.Rothkrantz Delft University of Technology Mekelweg 4 2228 CD Delft +31(0)152787504, L.J.M.Rothkrantz@tudelft.nl.