Animal.Net: Cooperative learning through networking software

Gerri Sinclair
(Faculty of Education, Simon Fraser University, Burnaby)

Malcolm Colton
(Mitek Data Corp., Vancouver)

A Local Area Network in the computer classroom offers exciting new possibilities for cooperative learning and student interaction. Few existing LANs, however, provide good interaction capabilities between computer stations, and even fewer programs have been written to take advantage of this multi-station interactivity. We have taken a well-known animal guessing game from the Public Domain and converted it into a networking program. Animal.Net is offered as an example of a new genre of software.

The original, stand-alone version of Animal is a learning program in which the user teaches the program to distinguish between animals. The program asks the user to think of an animal and then asks a series of questions until it believes it knows which animal the user has chosen.

For instance the program may ask the user if the chosen animal "lives in the water". If the user says yes, the program may then ask if the animal "is warm-blooded". If the user says no, the program may suggest that the user's animal is a goldfish. If the user's animal is not a goldfish but an eel, the program will ask the user to type in a question that will distinguish a goldfish from an eel. The user may type something like "Is it long, black, and slippery". The program will then ask the user to indicate what the answer to the question would be for an eel. Because eels are generally long, black, and slippery, the user would presumably type "YES", and the program would add this question to its growing internal database that already includes the information that goldfish and whales live in water and that goldfish are not warm-blooded. The program then asks the user to think of another animal, and the cycle starts again.

In the networking version, Animal.Net is divided into two logical processes, a master and a client. The master holds the database and provides and stores animals and questions for the clients. The master can support a large number of clients, which in turn interact with the users, presenting questions and receiving answers. Clients forward the users' new animals and questions across the network to the master for insertion into the database.

Even though the basic rules of the game remain unchanged, this change in the architecture of the program has remarkable consequences for the game itself. For instance, when playing the stand-alone version of Animal, the goal of the game is to think of another new animal and teach the program to distinguish between it and the previously entered animal. Unless the user forgets that the program already knows a particular animal, the program never guesses correctly.

The experience of playing Animal.Net is very different. Because many players are simultaneously updating the database, no one player can know which animals the program can recognize at any point in time. After a relatively short time, the program starts to make correct guesses. The game thus becomes one of enquiring into the state of knowledge of a shared community database, and updating it to reflect new information.

Moreover, when playing the stand-alone version, the questions and animals that the program presents to the user are those that the user has typed in himself or herself (except for the initial pair of questions and answers stored in the internal database). In other words, there are no surprises for the user in the stand-alone version because the information input into the internal database has been consciously extracted from the single user's personal knowledge-base. Animal.Net on the other hand is full of the unexpected and unpredictable. When a class of students, each at a separate station, types in information distinguishing their personally selected animals, each user is constantly presented with the unique and often idiosyncratic questions and animals that his or her classmates have entered. The internal database of the program thus becomes a collective representation of the knowledge-base of the entire group. And the intelligence, careful categorization, humour, and spelling reflected by the information contained in the database is a representation of that contained within the class as a whole entity or culture.

The networking version of the animal guessing game provides a very simple illustration of an information utility in a way in which a stand-alone game cannot emulate. Students act as both producers and consumers of shared information, interacting with one another via the information services provided by the software over the network.