ELECTRONIC BULLETIN BOARD SYSTEMS (BBS): SUPPORT SOFTWARE FOR COMPUTER SIMULATIONS

David J. Fritzsche, Richard V. Cotter,
University of Portland University of Oregon

ABSTRACT

While mainframe simulations were difficult to install, they offered some significant operating features which were lost when games were transferred to micro computers. Most of these features can be incorporated into micro systems through the use of bulletin boards and associated software. Thus the administrator is now able to benefit from the operating features provided by mainframes in addition to the unique set of features offered by micro computer. The authors review the development of desirable operating features on mainframe computers. Then techniques for incorporating these features into micro systems are discussed.

HISTORY

The mechanical tasks associated with administering an educational simulation have tended to decrease the attractiveness of using simulations as pedagogical tools. These tasks have consisted primarily of input-output activities associated with the computer (Fritzsche 1978). Fortunately the mechanical tasks have been significantly reduced as mainframe computer technology has developed.

For early simulations, team decisions were typically entered by giving the simulation administrator a completed decision form. The administrator would then punch the decisions on computer cards. When it was time to run the game, the administrator would submit the computer simulation program card deck along with the decision cards to the computer operator. The operator would load the simulation and the decisions via card reader and run the simulation. The simulation output would be listed on the computer printer and returned to the simulation participants by the game administrator (Fritzsche 1987).

When magnetic disk drives became available for program storage, the simulation program was transferred from cards and stored on magnetic disk. The administrator then needed only submit the decision cards to the computer operator prior to the simulation run. When card punches became available for student use, many administrators required the teams to submit their decisions on punch cards thus transferring the responsibility for correct decision entry to the simulation participants.

The next development occurred when terminal access to mainframe computers became available to faculty. Some administrators began entering student decisions directly into the machine for storage on magnetic disk, thus eliminating punch cards. Responsibility for accurate decision entry was thus transferred back to the game administrator. The game administrator could also execute the simulation from the terminal thus taking control for running the simulation from the computer operator.

When terminal access was extended to students, an opportunity was provided to transfer the responsibility for correct decision entry back to the Simulation teams. Utility programs were written which enabled teams to enter their own decisions directly to disk and to instruct the computer to print results on demand (Fritzsche 1979). Thus the administrator was completely out of the input-output loop, with the only remaining mechanical responsibility being to run the simulation. The utility programs also included error checking routines which significantly reduced the need to rerun the simulation due to errors in data entry.

As modems became available, input-output activities could take place at remote locations. This was particularly advantageous for graduate courses which often meet once a week. Teams could enter decisions and print output using terminals at work or at home. Thus several decisions could be run per week without requiring the teams to travel to campus to turn in decisions and pick up simulation output.

Modems and portable terminals also allowed the administrator to run the simulation at remote locations during times other than regular business hours. The simulation schedule could be more nearly tailored to the schedules of graduate students by running the simulation at night or on weekends.

The final responsibility of running the simulation at a specific time was automated by one of the authors through the use of timed submit batch files which automatically run the simulation. A batch file is created to instruct the computer to run the simulation for a specific period. The file is then submitted to the computer system with instructions to be run at a specific time and date. When the appointed time arrives, the computer automatically runs the simulation with no input from the administrator. More than one batch file can be submitted which provides the opportunity to schedule multiple runs of the simulation prior to the beginning of a week. Thus the simulation can be run at times which are most convenient for the participants without regard to the administrator's schedule.

The mechanical tasks may now be nearly eliminated for the administrator providing more time for the creative tasks of the simulation. Team members are now responsible for correct decision entry. They input team decisions directly into the computer and obtain output in the same manner. The magnitude of the input-output tasks transferred to the participants have been significantly reduced and occur at times more convenient for the team members. The simulation can be set to run at a future date which eliminates the need for the administrator to be at a terminal when the simulation is run.

While the advent of the microcomputer has provided many advantages, improving upon the ease of running an installed simulation is not one of them. A simulation installed on a micro network maintains most of the development advantages cited above. However, unless it may be accessed via modem, it cannot be accessed remotely. A simulation installed on a stand alone micro with a hard disk requires that decision files be either entered on hard disk, which provides potential security problems, or submitted on floppy disk. In both cases, participants must be on site to enter decisions or to deliver disks. Micro based games normally allow teams to enter their own decisions which makes the team responsible for correct decision entry. However, the administrator must collect disks when floppies are used and must distribute the output either on disk or on paper. There is no possibility for the simulation to be set to run automatically at some future time.

ELECTRONIC BULLETIN BOARDS

The authors would like to share an alternative method of communicating with a microcomputer which provides most of the advantages of using mainframes while preserving the benefits of using a micro computer (See Fritzsche 1987, for a discussion of advantages). During the past year, both authors have used a bulletin board in conjunction with running a total enterprise simulation. The board was used with graduate classes as well as with an intercollegiate competition.

A bulletin board provides several of the benefits cited above for mainframe computers but not currently available on micros. It provides a remote method of submitting decisions and obtaining output without requiring teams to come to the host computer site. This can be a significant advantage for graduate classes which meet once a week but make two decisions per week. Second, it provides a communication channel

between students and the simulation administrator which is not dependent upon individual schedules. The. bulletin board message system permits either party to leave and retrieve messages. In addition, the administrator can leave bulletins on the board for all teams to see. Third, updated copies of programs and files can be made available to teams upon demand.

Requirements

In order to run a bulletin board, one needs the following equipment:

1. computer with a hard disk
2. modem
3. telephone line
4. bulletin board Software package

While public bulletin boards require a large hard disk, a bulletin board for a simulation does not as only a few files and messages are stored on the disk. The modem should preferably be a 2400 baud unit to insure timely transmission although one can get by with a 1200 baud unit. The telephone line does not have to be dedicated to the board. One of the authors turns on the board when he leaves the office for the day and turns it off when he next arrives in the office. (We suspect this has surprised some of his callers when they reach a computer instead of a human voice.) The line is thus available for voice communication while in the office and bulletin board communication when not in the office.

Bulletin Board Software

The selection of bulletin board software is a complex decision. An article in InfoWorld reviewing bulletin boards provides some insight into desirable features (Hildum and Needleman 1988). The authors chose RBBS-PC, a shareware product from Capital PC Users' Group. The board may be downloaded free of charge from bulletin boards and is available from the user group for $16. It is a full-featured board with a top rating in the InfoWorld evaluation. There are also other excellent boards available with programs being updated regularly.

The main concerns in selecting a bulletin board to support a simulation include unloading and downloading capabilities, security and message functions. An additional feature which one of the authors has found very helpful is the ability to exit to DOS and return.

When selecting bulletin board software, one should look for a board which supports a variety of communication protocols. For example, RBBS-PC can support Xmodem, Xmodem/CRC, Ymodem, Kermit and Windowed Xmodem among others. As users will employ a variety of communication programs which support different Beta of protocols, the board must offer a sufficient variety to meet the requirements of the users.

The user must set the protocol for the board to use prior to downloading files. The protocol should be easily selected with board instructions for setting in plain English. Many users will only have a vague idea of what a communication protocol is. The protocol selected must match one available in the users communication software.

A bulletin board allows users to get into the host computer! Thus security of the computer is a critical factor. One may wish to lock out all users other than the players comprising the simulation teams. That will keep the general public from doing any damage. It will also significantly decrease the chances of encountering viruses. The teams will be uploading only data files, a type of file which are not normally infected. The simulation administrator (Sysop or system operator) should have a higher security level than the teams. Security should be set so that teams are limited to uploading and downloading files and sending messages while the Sysop should have complete access to the machine.

A second dimension of security involves individual team files. This is preferably done with passwords. RBBS-PC allows the sysop to password files available for downloading. Thus each team can be given a password which is assigned to its output file. This can be the same password which is used by the team to log onto the board. Unfortunately, one cannot assign passwords to uploaded files. While security levels can and should be set so that teams cannot access uploaded files, a team could enter an unfavorable decision in its local decision file, change the name of the file to that of one of its competitors and upload the file. Fortunately, the board can and should be set so that a user cannot overwrite an uploaded file by uploading another file with the same name. Thus the deception would be discovered when the real team tried to upload its file. In addition, some simulations such as the Business Policy Game check the internal identification of each decision file when the program is run and thus the deception would be discovered when the simulation was run.

Ideally, separate subdirectories could be established for each team with the subdirectories being password- ed. Then files could be uploaded to and downloaded from each directory. However, the authors are not aware of any boards which currently have such a capability.

Messages should be easily created and read to facilitate communication. A capability to issue bulletins which every team is forced to read at logon is useful to disseminate critical information quickly and completely. A pre-logon message is also desirable to provide information to users prior to actually logging on the board. The pre-logon message may be used to inform nonqualified users that access is denied, and it can also be used to inform teams that a specific period has been run. Thus if for some reason a decisions is run late, teams do not have to log onto the board only to find that their output is not yet ready.

Setting up a bulletin board program is a bit more complex than many other software programs. Most of the bulletin board programs provide assistance for the potential sysop by providing setup programs. Menus, bulletins and other screens are generally ASCII files which can be created by most text editors and word processors.

RUNNING THE SIMULATION

Setup

The simulation should be set up to run entirely from the hard disk of the host computer. Thus, subdirectories should be created for each team so that decisions can be read from the individual subdirectories. Team decision files are copied from the upload directory of the bulletin board to each of the team's subdirectory. Each teams output is also written to the subdirectory.

The bulletin board is set up with one upload subdirectory and one download subdirectory. Each team is provided a user name and a password to use when logging onto the bulletin board. The same passwords may be used for each team's output file which will reside in the download subdirectory. Files uploaded by teams should be the same files that are created by the student decision entry program. The file names are those that the simulation program expects to find when reading team decision files from the hard disk subdirectories.

It is useful to review the use of the bulletin board with the simulation teams prior to beginning the competition. While some individuals will be experienced board users, the majority will be somewhat apprehensive toward the new technology. Information on setting communication protocols and using the board will help put their minds at ease.

The Simulation Run

Prior to running the simulation, the team files in the upload subdirectory must be copied to their respective team subdirectories. This can easily be accomplished by creating a batch file using replaceable parameters for time specifications. When the files have been copied, the simulation is run in the manner it would be if the decisions had been directory entered into the team subdirectories.

After the simulation has been run, the team output must be copied to the download subdirectory. While the decision files are small data files which take a

very short time to upload, the output files may be much larger and require a significantly longer time to download. In addition, several output files may be required to print a team's output. The authors suggest using a file compression program to create a compressed file of each teams output prior to transferring it to the download directory. The compressed file will be smaller in size and will thus require less time to download. In addition, multiple files may be combined into one compressed file so that a team can obtain all required files by downloading one file. The individual files are recovered when the files are unpacked from the compressed file.

After the compressed files have been created using files in the team subdirectories, the compressed files must be copied to the download subdirectory. Both tasks can be accomplished in one step using a batch file with replaceable parameters. The batch file will create a compressed file in the download subdirectory for each team consisting of the team output files from the team subdirectory.

Several shareware file compression utilities are available for downloading from bulletin boards. ARC by System Enhancement Associates and PKZIP by PKWARE, Inc. are two of the most popular utilities. If ARC. -EXE is used, the teams must be provided with a copy of ARCE.COM to unpack the files. MKSARC.EXE may be used to create compressed files which can be unpacked by simply typing the file name. However, the self unpacking files are somewhat larger than those created with ARC.EXE. PKZIP.EXE is used to compress files and PKUNZIP.EXE is used to unpack files. MAKESFX.COM and ZIP2EXE.EXE are used to create self unpacking compressed files.

Bulletin board systems require teams to enter their own decisions into computer files and thus maintains team responsibility for the correctness of decisions. A board also makes teams responsible for transferring files to the host computer and obtaining and printing team output. Thus only mechanical task remaining for the simulation administrator is to run the simulation program.

A bulletin board offers one more advantage lost through the adoption of micro based simulations. Remote access on mainframes allowed the administrator to run a simulation from home during nonbusiness hours. This same benefit can be achieved using a board. If the board can exit to DOS, the simulation can be run remotely given the proper software. This becomes a bit more difficult on micros than on mainframes as the micro host writes some types of output directly to the screen. This output is not directly available for output through a modem.

Several commercial programs have been developed which provide remote access to host micros. Remote access programs were reviewed by InfoWorld in 1988 (Tapanila and Bigley). In addition, a shareware program, Doorway, has successfully been used by one of the authors to remotely run a simulation. Setting a board up for remote running is somewhat technical and should probably not be attempted by a novice computer user.

If remote running is desired, some type of carrier detect program is useful. WATCHDOG.COM by Jim Reinders is included with the RBBS bulletin board program and thus is readily available if that board is utilized.

By using a bulletin board, the simulation administrator can recover most all of the desirable operating features of a mainframe computer on a micro computers. Teams take responsibility for entering their own decisions. They upload decision files and download output files. The simulation can be run remotely by the simulation administrator. The only benefit missing is the timed delay simulation run. The authors may be able to add this by the time ABSEL meets.

APPENDIX

SOFTWARE REFERENCES

ARC
Systems Enhancement Associates
21 New Street
Wayne, NJ 07470

DOORWAY
Marshall Dudley
406 Monitor Lane
Knoxville, TN 37922

RBBS-PC
Capital PC Users Group
P.O. Box 6128
Silver Springs, MD 20902

ZIP
PKWARE, Inc.
7545 North Port Washington Road
Glendale, WI 53217

BATCH FILE FOR RUNNING RBBS

This file called RBBS.BAT should normally be run automatically from the AUTOEXEC.BAT file. Then if for some reason there is a power outage, when the power is restored the board will automatically reinstall itself. The command RBBS.BAT should be the last command in the AUTOEXEC.BAT file. The two WATCHDOG statements refer to a carrier-detect signal program and are included only if one desires to run the simulation using a computer at a remote location.

```
CTTY CON
CD \RBBS
IF EXISTS RBBS1F1.DEF DEL RBBS1F1.DEF
WATCHDOG OFF
RBBS-PC
WATCHDOG ON
IF EXIST RBBS1F1.DEF DOORWAY COMl /Ml00 /G:ON
/A:ON /V:D^U /S:* /O:T /C:DOS
C:
CD\
RBBS. BAT
```

REFERENCES

Fritzsche, David J., "A Data Entry and Retrieval System for a Computer Simulation (DERS)," Insights into Experiential Pedagogy, The Proceedings of the Sixth National ABSEL Conference, New Orleans, Louisiana, 1979, pp. 291-294.

Fritzsche, David J., "Lets Simplify the Administrative Requirements of Computerized Educational Simulations," Exploring Experiential Learning Simulations and Experiential Exercises, The Proceedings of the Fifth National ABSEL Conference, Denver, Colorado, 1978, pp. 39-43.

Fritzsche, David J., "The Impact of Microcomputers on Business Educational Simulations," Simulation & Games, Vol. 18, 1 2, June, 1987, pp. 176-191.

Hildum, Mike and Raphael Needleman, "BBS Software', InfoWorld, vol. 10, #33, August 15, 1988, pp. 39-47

Tapanila, Glen and Tom Bigley, "Remote Communications Software," InfoWorld, Vol. 10, #46, November 14, 1988, pp. 89-104.