

Developments in Business Simulation & Experiential Exercises, Volume 9, 1982

ADAPTING MAINFRAME BUSINESS SIMULATIONS TO MINI COMPUTERS

William D. Biggs, Beaver College
Theodore J. Smith, Beaver College, Student

ABSTRACT

This paper deals with our experiences in adapting business simulations which were originally developed on and designed for larger mainframe computers to the much smaller mini computers. Based upon our experiences, suggestions are provided for potential users of simulations who will have to make such adaptations. Problems encountered with specific simulations as well as the solutions to the problems are detailed. Finally, suggestions are provided for simulations designers to help alleviate some of the problems.

BACKGROUND

Prior research has shown that institutionally users of computerized business simulations "tend to be 4-year rather than 2-year, public rather than private and large rather than small" (2, p.204). With the advent of the mini computer, however, many small colleges and universities can now afford to own and operate a computer. It is likely, therefore, that with the hardware available more institutions will be interested in adopting computerized business simulations, particularly if they can easily adapt them to their computer system relatively quickly. Adopting such simulations may be difficult, however, because to date most major business simulations require implementation on a large scale (mainframe) computer with a large amount of available main memory. In spite of this situation, major business simulations can be adapted to the smaller mini computers if the requirements of the simulation (software) can be identified and adjustments made to reduce these requirements without affecting the outcome of the simulation. In addition, the differences between the mainframe and mini computer need to be identified so that errors due to the components of the computer (hardware) can be eliminated.

DEFINITIONS

Before presenting our specific suggestions we need to provide the reader with our definitions for mainframe versus mini computer. Although the distinction between the mainframe and the mini computer is ambiguous, we will use computer word size and the amount of available main memory to differentiate between the two. Thus, a mainframe has a word size of 32 bits and at least 50K words of available main memory. Since the IBM 360 and 370 computers are among the most common mainframes on which business simulations were written, we will use them throughout as an example of a mainframe. A mini computer on the other hand is much smaller than the mainframe with a word size of 16 bits with less than 50K words of available main memory. The mini computer to which we adapted the simulations was a DEC PDP-11/45 with a 16 bit word size and 28K words of available main memory.

SUGGESTIONS FOR POTENTIAL USERS

Adapting a large simulation to fit a small mini computer can be very tedious if it is done haphazardly or by using a trial-and-error method. After adapting a few large simulations, a method came into focus which enabled us to proceed in an orderly step-by-step fashion, to get large main frame oriented business simulations to work correctly on a mini computer. This approach considerably reduced the amount of time needed and the number of errors to overcome in adapting the simulations. The major steps identified are as follows:

1. Read the student manual to assess what the simulation is designed to do and requires as input.
2. Read the instructor's manual to ascertain the computer requirements for the simulation.
3. Read the System's User Manual and Language Reference Manual for your mini computer to locate differences and possible trouble spots."
4. Compare and identify possible problems due to hardware and software restrictions of your system.
5. Be familiar with the methods of segmentation and overlaying available as discussed later in this paper.
6. Adjust, implement and test the simulation.

It is important that the user do steps 1 and 2 thoroughly; while steps 3 through 6 may be handled by someone knowledgeable about the computer. These steps represent a generalized method which outline how to implement a simulation by recognizing the requirements which cannot be met directly by the mini computer and where adjustments will probably be needed.

The user needs to be thoroughly familiar with the student manual if the simulation is to be used effectively and efficiently. The student manual provides the rules of the simulation and the input required of the participants. The user also assesses at this point whether or not the simulation will meet the objectives for which it is being considered. While this step appears to be common sense it is well observed that "common sense is not so common" (11, 1764).

It is important to become knowledgeable of the simulation limitations and computer requirements for the simulation. This information is usually located in the instructor's manual. The Instructor's manual should contain the following:

1. The amount of main memory required to operate the simulation.
2. The I/O devices needed.
3. The variable definitions and their data type
4. The maximum and minimum range for numeric variables.

Developments in Business Simulation & Experiential Exercises, Volume 9, 1982

5. A flowchart of the simulation program routines to include overlay and segmentation information.
6. The descriptions of Processor-defined functions utilized.
7. The unusual or non-standard language applications.
8. A general description of all subroutines.

Unfortunately, most instructor's manuals only describe a few of the above requirements and limitations. Therefore, in adapting the simulation the user may spend a great deal of time "re-inventing the wheel;" may be forced to contact the simulation designer for assistance; or may just give up and not use the simulation.

The System User's Manual and the Language Reference Manual should be checked to identify the limitations of the mini computer and any limitations pertaining to the source language (usually FORTRAN) of the simulation. This step is intended to locate the major barriers to overcome in starting up the simulation. As a consequence, this step will also help in the design of any adjustments for the simulation, because the limitations of the mini will be fresh in the designer's mind.

Once the requirements of the simulation and the limitations of the mini computer have been identified, factors which will prevent the proper performance of the simulation will become evident and the adjustments for them can be designed. This step is crucial and if it is done properly will prevent errors, thus, saving considerable time and effort. Below are some of the common differences between the mainframes and mini's which might be identified in this step:

1. Hardware considerations
 - a. Available main memory
 - b. Word size of computer (Affects range of Integers)
 - c. Secondary storage mediums (disks, printers, card readers, etc)
2. Software considerations
 - a. Range of Integers
 - b. Methods of I/O (file processing)
 - c. Non-arithmetic processing (alpha-numeric)
 - d. Subroutine entry points

The most obvious of the differences will be between the amount of available main memory and the amount needed by the simulation. However, the other differences are just as important to adjust in order to obtain the correct results.

Segmentation and overlaying are the two possible solutions to reduce the amount of main memory needed by a simulation. Segmentation is used when a program routine is too large to compile. Segmentation is accomplished by splitting up a routine into smaller routines. COMMON blocks should be used to retain values that may be destroyed or altered in subsequent calls or if the routine is overlaid. Overlaying is used when the total amount of memory required to run the simulation exceeds the amount available. It is necessary to construct an overlay structure which allows the swapping of routines not currently needed out of main memory onto a secondary storage device (usually disk) until they are called by the main (root) routine. A flowchart of the simulation program structure shows when and where each subroutine is called.

Using this flowchart, an overlay structure can be constructed easily. Ease of construction, however does not guarantee success. Care must be taken when creating overlays because as Hughes, Pflieger, and Rose (5, p. 269) note: "The added time that is required to process calls among overlay routines is not trivial, however. A programmer seeks to plan an overlay structure that minimizes this time." It should be noted that segmentation and overlaying are not mutually exclusive approaches; depending on the simulation, either or both methods may be needed to reduce the required amount of main memory. An overlay structure for one simulation is presented later in this paper for illustrative purposes.

Adjustments can now be made to the simulation, so that, its limitations and requirements can be brought within the constraints of the mini computer. When making adjustments, care should be taken to avoid unintended alterations caused by the necessary adjustments. Working on one adjustment at a time can avoid making any unforeseen or unwanted alterations to the simulation. Some of the most common adjustments used in adjusting the simulation to the PDP-11/45 are outlined below:

1. Segmentation and overlaying were used to reduce the amount of main memory needed by the simulation.
2. Conversion of some integer data types into real data types was required in most of the simulations adapted, due to differences in the word size between the mainframe and mini.
3. Alpha-numeric variables needed to be adjusted because the amount of alpha-numeric characters that could be contained in a data type was dependent on the word size of the computer. On the mainframe, eight alpha-numeric characters could be contained in a real data type, whereas, the mini used could only contain four alpha-numeric characters in a real data type.
4. OPEN statements were added to the simulations' source code to gain better control of the input history and decisions decks and output from the simulation.
5. Disks were used in place of the card reader and printer. Output from the simulation was later sent to the printer. This was done in order to acquire multiple copies of output without re-executing the simulation.

In addition to the above adjustments, some simulations required additional adjustments to achieve the proper results. Various simulations will be used to show some of these additional adjustments.

THE SIMULATIONS ADAPTED

As noted previously the comments in this paper are based upon our experiences with a number of simulations which we successfully adapted from the large mainframe to the smaller mini computer configuration. The specific simulations were those by Brooks (2); Gitman, Robana and Biggs (3); Greenlaw, Frey and Vernon (4); Jensen and Cherrington (7); and Scott and Strickland (10). In addition, our comments are based upon a number of other simulations which we have not successfully adapted as of the date of this writing.

A minor problem presented itself when we attempted to adapt the simulation PORTSTRAT: a Portfolio Management Simulation by Gitman, Robana, and Biggs (3).

Developments in Business Simulation & Experiential Exercises, Volume 9, 1982

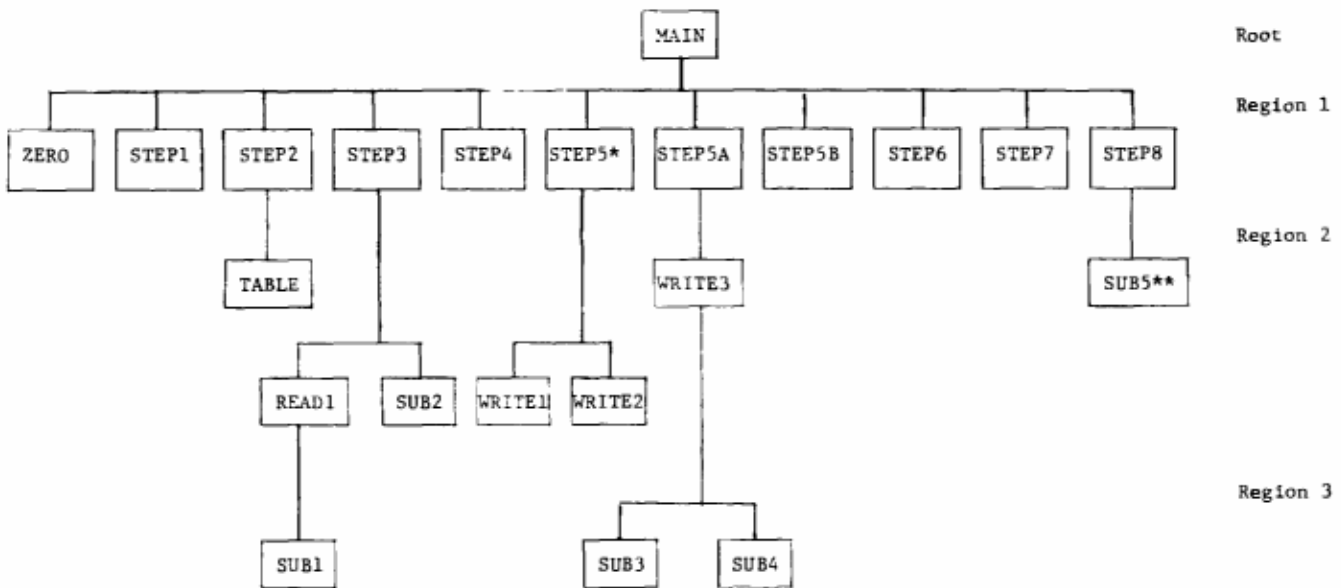
The word size on which the original simulation was written permitted up to four characters per variable (integer data type) while our system only permits up to two. This problem was easily corrected by changing the data type under which the variable was defined into a real data type which can hold four characters per word. Game designers can eliminate this potential difficulty by declaring their alpha-numeric variables as a logical*1 data type (on IBM use the character*1 data type) and by dimensioning the variable to a size that will contain the maximum length of the alphanumeric string.

When adapting FINANSIM: A Financial Management Simulation by Greenlaw, Frey and Vernon (4), it became necessary to segment the simulation. This was done because the original source program was written as one routine and was too large to compile on our computer. As a consequence, the simulation was segmented into three segments. The first segment read in the history and decisions, and processed them. The second and third segments wrote the results and new history for the current period. Similar segmentation was also required for the simulation FINGAME: The Financial Management Decision Game by Brooks (2).

When adapting the simulation Tempomatic IV: A Management Simulation by Scott and Strickland (10), overlaying was required and although a flowchart of the program structure was provided by the authors Scott and Strickland (9), it required modification so that an overlay structure could be constructed that operated on our system. It was necessary to duplicate a subroutine. In order to correctly obtain the overlay structure for our system. Figure 1 shows the modified overlay structure which is being used on our mini computer.

One of the easiest simulations to adapt was Business Management Laboratory by Jensen and Cherrington (7). The simulation was written in small routines in which an overlay structure was easily constructed. The Instructor's manual was very helpful in the adapting of the simulation, due to the fact that the authors included test data and output in the instructor's manual. They also provided suggestions for starting-up the simulation and its maintenance. Although there was some difficulty in adjusting for the values of integer type variables, and the random number generator, the adaptation of this simulation was accomplished rather quickly and without requiring major adjustments.

FIGURE 1
FLOWCHART OF OVERLAY STRUCTURE FOR TEMPOMATIC IV FOR PDP-11/45



Overlay Structure:

Root is MAIN

Region 1 Contains: ZERO, STEP1, STEP2, STEP3, STEP4, STEP5, STEP5A, STEP5B, STEP6, STEP7, STEP8

Region 2 Contains: TABLE, READ1, SUB2, SUB5, WRITE1, WRITE2, WRITE3

Region 3 Contains: SUB1, SUB3, SUB4

Notes:

* STEP5 was segmented into STEP5, STEP5A, STEP5B

** SUB5 is a duplicate of STEP1 which was required to balance the tree (flowchart).

Developments in Business Simulation & Experiential Exercises, Volume 9, 1982

SUGGESTIONS FOR GAME DESIGNERS

Our prior discussion obviously includes suggestions for game designers. Particularly, designers should consider the normative statements relative to information to be included in the instructor's manual. Designers also need to be aware that word size and random number routines vary from system to system.

The instructor's manual should provide users with statement numbers where word size would matter with modifications necessary. The random number problem is harder to deal with; however designers could provide the random numbers in effect when the trial results were generated and instructions on how to bypass the random routine and use the random numbers provided. This approach would permit the user to do a test run identical to the designers sample output to ascertain that all other aspects of the simulation are working correctly. Designers can also work to reduce the complexity of simulations in terms of the computer model complexity which refers to such items as the program language, number of lines or pages of output, memory required, etc. (8). Finally, as Jensen (6) suggests designers may develop simulations on microprocessors rather than moving programs from larger systems. Even here designers must make their simulations portable and generalizable.

SUMMARY

Our goal in this paper was to provide suggestions for potential users of computerized business simulations who wish to adapt a simulation onto their mini computer. We have generalized our examples so as to prevent hardware dependent solutions. Although the user may be discouraged by the notion of adjusting a simulation, it can usually be accomplished without severe difficulty or time consumption if the adaptation of the simulation is clearly and logically thought out, and executed. The method supplied by us is intended to serve as a base (starting) point in achieving a correctly adapted version of the desired simulation.

REFERENCES

- (1) Biggs, W.D., "Who is using Computerized Business Games?: A View from Publishers' Adoption Lists," in S.C. Certo and D.C. Brenenstuhl (Eds.), Proceedings of the Sixth Annual Conference of the Association for Business Simulation and Experimental Learning, 1979, pp 202-206.
- (2) Brooks, L.D., The Financial Management Decision Game (Homewood, Illinois: Richard D. Irwin, Inc. 1975).
- (3) Gitman, L.J., Robana, A., and Biggs, W.D., PORTSTRAT: A Portfolio Management Simulation (New York: John Wiley and Sons, 1981).
- (4) Greenlaw, P.S., Frey, M.W., Vernon, I.R., FINANSIM: A Financial Management Simulation (St. Paul, Minnesota: West Publishing Co., 1979).
- (5) Hughes, C.E., Pfeleger, C.P., Rose, L.L., Advanced Programming Techniques: A Second Course in Programming using Fortran (New York: John Wiley and Sons, Inc., 1978).
- (6) Jensen, R.L., "Simulations and Microprocessors," in W.D. Biggs and D.J. Fritzsche (Eds.), Proceedings of the Eighth Annual Conference of the Association for Business Simulation and Experimental Learning, 1981, p 86.

- (7) Jensen, R.L. and Cherrington, D.J., The Business Management Laboratory (Plano Texas: Business Publications, Inc., 1977).
- (8) Keys, J.B. (Ed.), "Total Enterprise: Computerized Simulations," in E.R. Horn (Ed.), The Guide to Simulation Games for Education and Training (Crawford, New Jersey: Didactic System, Inc. 1977).
- (9) Scott, C.R., and Strickland, A.J., III, "Computer Center's Manual" for Tempomatic IV: A Management Simulation (Boston: Houghton Mifflin Company, 1980), p 6.
- (10) Scott, C.R., and Strickland, A.J. , III, Tempomatic IV: A Management Simulation (Boston: Houghton Mifflin Company, 1980).
- (11) Voltaire (Francois Marie Arouet), Dictionnaire Philosophique, as quoted in John Bartlett, Familiar Quotations (Boston: Little Brown and Company, 1968), p 417.