

Developments in Business Simulation & Experiential Exercises, Volume 15, 1988

AN INITIAL STEP TOWARDS DEVELOPING AND USING AN EXPERT-SYSTEM WITH A BUSINESS SIMULATION

Theodore F. Gautschi, Bryant College
Mohad Prasad, Bryant College

ABSTRACT

This paper describes an initial step that was taken to develop an expert-system to be used with a business simulation program.

In view of the increasing importance of business simulation and its requirement for greater reality; and the emergence of AI technology, in particular expert-system technology; the question becomes not whether AI will have a place in simulation; but when and how. Rather than wait until the technology is more fully proven, we made a decision to develop a rule based expert-system application that would enable us to initiate some exploration of one aspect of AI without significant risk.

We chose to develop an expert-system using a commercially available shell. Our objective was to:

1. Reduce the student time and frustration involved in mastering the Bryant Management Laboratory (BML) simulation.
2. Acquaint students with some aspects of the field of expert-systems.
3. Develop an expert-system base that could easily incorporate future simulation enhancements.
4. Gain some practical development experience with expert-systems.

The current (Fall 1987) product is labeled "Trouble-Shooter." It will enable any student, with the appropriate disk operating on an IBM compatible PC, to determine why the BML simulation incorporated a seemingly "wrong-decision" or produced a seemingly "wrong-result."

Introduction

Business simulation games using computer support are widely used in both colleges and businesses. According to a recent survey:¹ "(1) Approximately 1,914 four-year schools (colleges) are using simulation games in some part of their business program; (2) Simulation exercises are being used in a minimum of 3,287 separate business courses; (3) There may be as many as 8,755 business instructors using simulation exercises at least once during an average school year; and between 6,100 and 7,200 business firms are currently using simulation exercises as part of their management training programs." Not only are such exercises being widely used, but "total enterprise games have improved greatly in educational value, simulated realism and administrative simplicity, since the late 1950's when games were first used in business schools".²

It is the contention of this paper that: (1) the current widespread utilization of simulation games will continue; and (2) simulation games will continue to be improved in terms of their educational value, realism and administrative simplicity as they make use of emerging advances in the information processing field. Easy student access to the smaller, faster, higher capacity, greater functionality computers such as the IBM PC's and the recently announced IBM PC/2 series coupled with expert-system developments from the growing field of Artificial Intelligence (AI) provides the opportunity to enhance the overall learning process.

The focus of this paper is to describe an initial step that was taken towards developing and using an expert-system with the Bryant Management Laboratory (BML) business simulation program. Believing that the question is not whether AI will have a place in simulation, but only when and how; a decision was made to try to develop an expert-system application: Our objective was to:

1. Reduce the student time and frustration involved in mastering the BML simulation.
2. Acquaint students with some aspects of the field of expert-systems.
3. Develop an expert-system base that could easily incorporate future simulation enhancements, and
4. Gain some practical development experience with expert-systems.

This paper will:

1. briefly describe the BML,
2. briefly discuss AI, and particularly expert-systems, their possible application to simulation games, and the decision to develop Trouble-Shooter,
3. list criteria for selecting an expert-system shell, and our selection,
4. review the development of Trouble-Shooter,
5. review the student Trouble-Shooter package and its operation,
6. evaluate our results, and
7. discuss future possibilities.

¹ "A Survey of the Use of Business Games in Academia and Business" by A. J. Faria, published in Simulations of Games, Vol. 18, No. 2, June 2, 1987 by Sage Publications.

² "Total Enterprise Business Games, An Evaluation," by J. Bernard Keys. Published in the ABSEL proceedings, 14th Annual Conference, 1987.

Developments in Business Simulation & Experiential Exercises, Volume 15, 1988

BML Simulation Program

The BML is a flexible, sophisticated, challenging, state-of-the-art computerized total-enterprise business simulation. It is an adaption and expansion of Barton's "The Imaginit Management Game"³ and is used in a dynamic-case format. Currently it has some 118 parameters which can be changed at any time by the administrator to establish the desired business environment and operational rules. The BML is designed to be flexible and issued by Bryant College at three different levels--each progressively more realistic and therefore more complex.

Level 1 deals with the Robot industry. It is the simplest dynamic-case version. It is used to support introductory business/management courses by establishing a common experiential base to which the students can relate.

Level 4 is concerned with the automobile industry. It is more complex than Level 1, and is used to support the undergraduate senior strategy and policy course by helping the students to integrate what they have learned in previous courses and to practice significant managerial skills.

Level 7 deals with the microcomputer industry. It is the most realistic version. It is used with the MBA capstone policy and strategy graduate course. In addition to integrating previous courses, this version serves as an interactive model for the implementation and testing of strategic management decisions.

Participant and administrative manuals for Level 7 are documented and maintained using word processing software. These, or appropriate subsets, are published as required to meet user requirements for each of the three levels.

Since the various BML levels are designed to stretch the participants' state of maturity and knowledge, it can be difficult for them to achieve an understanding of the simulated environment and rules to the degree that they can devote most of their time and effort to strategic decision making. Recent surveys indicate that many of the BML participants have this problem, and they suggest that more help be provided in this area.

Artificial intelligence (AI) and the Possible Application of Expert-Systems to Simulation Programs

The ultimate goal for AI is to develop machines that can emulate human ability to reason and think and make decisions, that is, to create machine systems capable of imitating human behavior⁴. The current focus of AI is on:

- * Expert-systems
- * Natural language processing
- * Intelligent computer aided instruction
- * Automatic programming
- * Visual recognition
- * Voice recognition
- * Robots that can respond to their environment
- * Specialized AI hardware and software

The area receiving the most attention, at least in commercial circles, is the expert-system. The underlying concept is that if a human expert, through a reasoning process can "do it," then a computer can also be programmed to "do it" using a similar reasoning process, provided that the computer is given the human expert's codified knowledge.

The market for expert-systems is in three areas:

1. The research and advanced development market which is concerned with advanced languages and the development environments for expert-systems.
2. The end user market which consists of smaller corporations and commercial organizations that want to purchase entire turn-key systems. This market is in its infancy, but it has good growth potential.
3. The mainstream market which is comprised of major corporations and other commercial organizations that develop significant software applications in-house using expert-system support tools provided by outside vendors. In their simplest form, these support tools are called "shells" and are often used with microcomputers.

We decided that a commercially available PC shell would best suit our purposes. Expert-systems could have applications to simulation games in the following areas:

1. Trouble Shooter--answering students questions regarding the simulation operation and the generated results.
2. Extending simulation capabilities.
3. Making the simulation process more transparent to the user.
4. Decision support systems.
5. Simulating competitor decision making for developing defensive strategy.
6. Implementing or supporting "hard-core analysis" activities.

From this list we selected Trouble Shooter as something we believed we could accomplish and would meet our four objectives listed previously in this paper.

Expert-System Shell Selection Criteria

The following criteria were considered as we investigated various expert-system shells:

1. It must be easy to use without requiring an extensive expert-system background.
2. It should provide for expansion beyond our present Trouble Shooter application.
3. It must be economical to purchase, and then later inexpensive to duplicate for student use.
4. The expert-system should be relatively easy to develop and modify.
5. It must operate on a PC.
6. It should have reasonable run time in operation.
7. It should provide both development and separate run time environments.
8. It must be currently available.
9. It must be supported and updated by an experienced vendor.

³ The IMAGINIT Management Game, Richard F. Barton, Lubbock, Texas; Active Learning 1978).

⁴ Series of "Management Forum" articles by T. F. Gautschi titled "A Management Perspective on AL Cahner's Publishing Co. Design News. Boston, Massachusetts. February 17-August 4, 1986.

Developments in Business Simulation & Experiential Exercises, Volume 15, 1988

We selected Texas Instrument's Personal Consultant Easy as the expert system shell that would most closely match the selection criteria. In addition to our selection criteria:

1. The TI "PC Easy" is a second generation shell that is upward compatible with the TI "Personal Consultant Plus." This would allow initial development under "PC Easy" and yet permit us to move up to the more advanced "Personal Consultant Plus" when we are ready.

2. A site license agreement with the vendor allows unlimited distribution of the run time environment software to the students.

Development experience using this shell has helped us identify other criteria that should have been included in the original list:

1. The level of documentation provided with the software shell.

2. The robustness of the development and run time environments. This may be loosely translated to say "freedom from unexplained and unrecoverable software package bugs which impede the expert-system development."

3. The ability and willingness of the vendor to provide solutions to major problems caused by the software. Note that this is not the same as fixing problems caused by the user.

4. The relative ease of editing the knowledge base; ability to use word processors or text editors versus the dedicated editor usually provided in the shell.

5. The response time required for a vendor/user to complete an agreed upon benchmark problem. This would provide an objective, or at least consistent, measure of run time efficiency. However, since many of the commercially available PC-based expert-system shells are at a relatively infant stage, it would be difficult to make criteria-based comparative evaluations of vendors and their offerings.

6. Theoretically, the expert-system should be independent of the shell used to implement it. This goal is difficult to attain with implementations such as the Trouble Shooter, where the design is dictated to a considerable degree by the capabilities and limitations of the shell.

The Development of Trouble Shooter

Recently Louis Fried stated,⁵ "Knowledge-based system tools and technology are at a stage comparable to that of data base management systems in the mid-1960s. They are poorly understood, few people know how to use them appropriately, and few people can determine what their benefits will be to the user organization." Notwithstanding, we decided to plunge ahead. At minimum it would be a good learning experience.

Trouble Shooter is designed to respond to two basic types of problems:

1. The student believes that the computer did not correctly accept the decision he/she made. These are called "wrong-decisions."

2. The student believes that the computer did not correctly process the decisions he/she made. These are called "wrong-results."

All problems that Trouble Shooter can deal with are presented to the student in menu form. "Wrong-decisions"

lists all possible decisions. "Wrong-results" has an entry for every line listed on the BML computer printout. The student indicates which decision or line item he/she believes to be wrong. Trouble Shooter will then lead the student via Trouble Shooter questions and student responses to a reconciliation of the selected problem. The reconciliation will be a valid answer, or a suggestion to see the administrator.

With this implementation, Trouble Shooter does not draw conclusions from the data fed into it, rather it follows a deterministic path of inquiry driven by student responses until the problem is reconciled. Because BML has a number of rules and constraints, it is often difficult for the student to understand why something has happened. In this case, after some research and reflection, he/she will ask the administrator what the problem is. Hopefully, Trouble Shooter will be more readily available, provide consistent answers, and reduce the amount of time that the student must spend resolving a problem.

BML experts prepared expert flowcharts in a tree-like form to communicate with the knowledge engineer, who, in turn, developed the expert-system knowledge base. See Figure WD-10 for example, this is one of the simpler flowcharts, but it demonstrates the process that was used.

You will note that the student must enter parameter and! or results information. This is because Trouble Shooter operates stand-alone" on a PC, and the BML is processed on an MVI0000 mini-computer and at the present time there is no direct communication between the simulation and Trouble Shooter.

Implementation

The knowledge engineer first reviewed the expert's flowcharts for logical completeness and continuity (as opposed to checking the accuracy of the subject matter itself), then developed decision tree logic from the experts' flowcharts. One or two levels of this logic was developed, as determined by the flowcharts' complexity. This logic was then expressed as rules using a series of IF-THEN statements. For example, WD-10 required five rules, one each for: "See Adm," "OK," "See Adm," "See Admin," and "not sure." Figure 2 is the decision rule for "not sure." A parameter list was also maintained, to allow orderly changes as the project progressed. The next step consisted of entering the IF-THEN rules and the many parameters, each with its unique set of characteristics and English language identifier. The final step consisted of fine tuning the system and straightening out the backward-chaining and forward-chaining pieces of logic. Testing was not a separate step, it was an integral part of the implementation process at every stage.

Student Packages for Trouble Shooter

The delivery system given to the student consists of two disks:

1. The Runtime Diskette which holds the PC Easy software.

⁵ "The Dangers of Dabbling in Expert-Systems" by Louis Fried, Computerworld, June 29, 1987.

Developments in Business Simulation & Experiential Exercises, Volume 15, 1988

2. The Trouble Shooter disk which holds the expert-system knowledge base in protected format. This format prevents changes to the knowledge base.

These two disks permit the student to use Trouble Shooter on any compatible PC. Some PC's in the computer lab are loaded with the Runtime diskette and the Trouble Shooter knowledge base; in this situation, students can use these PC's directly, without additional disks. Instructions for use are relatively simple and a one-time demonstration in the classroom paves the way for the students self-learning process using Trouble Shooter.

Future

1. Make greater use of the "Review" and "Playback" features built into the shell. This allows the student to review/change responses already given to questions presented by Trouble Shooter.
2. Make the expert-system increasingly data-driven.
3. Provide a direct link between the BML simulation and the expert-system.
4. Develop rules for hard-core analysis.
5. Develop models to enhance BML--for example:
 1. forecasting, finance, POM, decision making.
6. Expand the expert system to include more decision support capability.

Evaluation

We have not yet had the opportunity to use Trouble Shooter with a class. We are currently in the final stages of checking Trouble Shooter--Wrong Decision, and we plan to use it in a classroom setting Fall 1987 and will add an evaluation section to the final report as soon as possible. In any case, such evaluation can be reviewed in the March 1988 ABSEL conference.

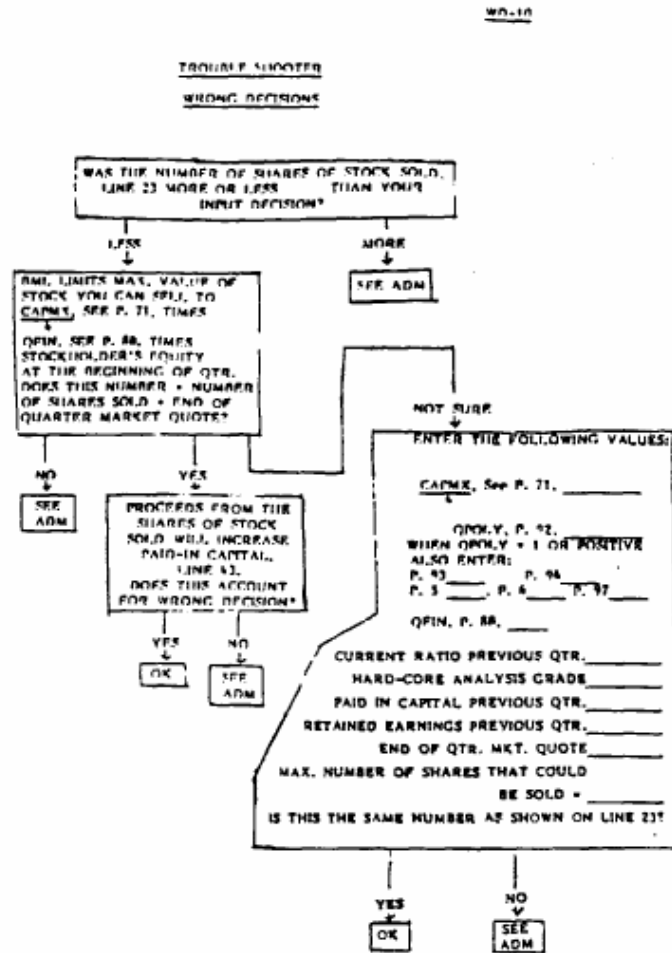


Figure 1

```

(RULE089 DESCRIPTION (SHARES SOLD 2/2 - SAMPLE CALCULATION) ACTION (DO-ALL (
CONCLUDE FRAME GOOD YES TALLY 100) (MPRINTT "SHARES SOLD" |LINE| 3
"The number of shares that could be sold is calculated as " (VAL1 FRAME CALC2)
|LINE| 1 "The QFIN used was" (VAL1 FRAME CALC1) |LINE| 4 |TAB| 26 |ATTR| (
QUOTE (WHITE REVERSE)) "CONSULTATION HAS ENDED." |ATTR| (QUOTE (WHITE)) |LINE|
3)) PREMISE ($AND (SAME FRAME GUD3) (SAME FRAME DETOP
SHARES OF STOCK TO BE OFFERED) (SAME FRAME Q32 LESS) (SAME FRAME Q33 NOT-SURE) (
SAME FRAME CALC2DUN)) SUBJECT DECISION-RULES)

(Q34A CONTAINED-IN (RULE119 RULE118) TRANSLATION (|paid| |in| |capital|) PROMPT
(|Please| |enter| |the| |value| |of| |paid| |in| |capital| |for| |the|
|previous| |quarter| |:) |LINE| 2) TYPE SINGLEVALUED EXPECT NUMBER RANGE (0
1.e10))
(Q34B CONTAINED-IN (RULE119 RULE118) TRANSLATION (|prev.| |qtr.| |retained|
|earnings|) PROMPT (|Please| |enter| |the| |value| |of| |Retained| |Earnings|
|for| |the| |previous| |quarter| |:) |LINE| 2) TYPE SINGLEVALUED EXPECT NUMBER
RANGE (0 1.e8))
(Q35 CONTAINED-IN (RULE119 RULE118) TRANSLATION (CAPMX |value|) PROMPT (|Next|
|enter| |the| |value| |of| CAPMX |:) TYPE SINGLEVALUED EXPECT NUMBER RANGE (0
2.))
(Q36 CONTAINED-IN ( ) TRANSLATION (QFIN |value|) PROMPT (|Next| |enter| |the|
|value| |for| QFIN |:) TYPE SINGLEVALUED EXPECT NUMBER RANGE (0.01 10.))
(Q37 CONTAINED-IN (RULE119 RULE118) TRANSLATION (|Market| |quote| |at| |end|
|of| |quarter|) PROMPT (|Finally| |enter| |the| |market| |quote| |at| |the|
|end| |of| |the| |quarter| |:) TYPE SINGLEVALUED EXPECT NUMBER RANGE (0 1.e7))
(Q17A USED-BY (RULE096 RULE097) TRANSLATION (QPOLY |greater| |than| |zero|)
PROMPT (|When| |QPOLY| |P92| |is| |greater| |than| |zero| |the| |firm's|
|current| |ratio| |from| |the| |previous| |quarter| |will| |change| QFIN.
|Enter| |the| |value| |of| QPOLY |:) TYPE SINGLEVALUED EXPECT NUMBER RANGE (-1
1))
(Q17B CONTAINED-IN (RULE097) TRANSLATION (|current| |ratio| |from| |previous|
|quarter|) PROMPT (|Enter| |the| |value| |of| |the| |current| |ratio| |from|
|the| |previous| |quarter| |:) TYPE SINGLEVALUED EXPECT NUMBER RANGE (0.01 100)
)
(Q17C CONTAINED-IN (RULE097) TRANSLATION (QF1 |:) P95) PROMPT (|Enter| |the|
|value| |of| QF1 (P95) |:) TYPE SINGLEVALUED EXPECT NUMBER RANGE (-100 100))
(Q17D CONTAINED-IN (RULE097) TRANSLATION (QF2 |:) P96) PROMPT (|Enter| |the|
|value| |for| QF2 (P96) |:) TYPE SINGLEVALUED EXPECT NUMBER RANGE (-100 100))
(Q17E CONTAINED-IN (RULE097) TRANSLATION (QF3 |:) (P97)) PROMPT (|Enter| |the|
|value| |for| QF3 (P97) |:) TYPE SINGLEVALUED EXPECT NUMBER RANGE (-100 100))
(Q17H CONTAINED-IN (RULE099) USED-BY (RULE100 RULE099) TRANSLATION (QFHI |:) P94
) PROMPT (|Enter| |the| |value| |for| QFHI (P94) |:) TYPE SINGLEVALUED EXPECT
NUMBER RANGE (0.01 100))
(Q17L CONTAINED-IN (RULE098) USED-BY (RULE100 RULE098) TRANSLATION (QFLO |:) P93
) PROMPT (|Enter| |the| |value| |for| QFLO (P93) |:) TYPE SINGLEVALUED EXPECT
NUMBER RANGE (0.01 10.))
(CALC1 CONTAINED-IN (RULE120 RULE119 RULE118) UPDATED-BY (RULE069 RULE070
RULE100 RULE098 RULE099 RULE082 RULE096 RULE089 RULE104) TRANSLATION (
|calculated| |value| 1) PROMPT ( ) TYPE SINGLEVALUED)
(CALC1A CONTAINED-IN (RULE100) USED-BY (RULE100 RULE098 RULE099) UPDATED-BY (
RULE097) TRANSLATION (|intermediate| |calculation| |of| QFIN) PROMPT ( ) TYPE
SINGLEVALUED)
(CALC2 UPDATED-BY (RULE069 RULE106 RULE116 RULE082 RULE117 RULE089 RULE104)
TRANSLATION (|calculated| |value| 2) PROMPT ( ) TYPE SINGLEVALUED)
(CALC2A CONTAINED-IN (RULE117) UPDATED-BY (RULE120 RULE119 RULE118) USED-BY (
RULE116 RULE117) TRANSLATION (|result| |before| |zero-check|) PROMPT ( ) TYPE
SINGLEVALUED)

```

FIGURE 2