CHAPTER 10

OPEN SYSTEM SIMULATION

Alan L. Patz

Most business policy simulations have a pedagogical intent (Thorelli and Graves, 1964). They are concerned with enhancing the degree to which students understand the flow, interrelatedness, and tong range character of general management decisions (Frazer, 1978). Moreover, most simulation research focuses on a determination of the conditions necessary to integrate more fully simulation exercises and traditional coursework (Klein, 1984). Most important, most simulations are closed systems. The algorithms that determine participant results remain the same throughout a given run or execution of the simulation (Patz, 1987).

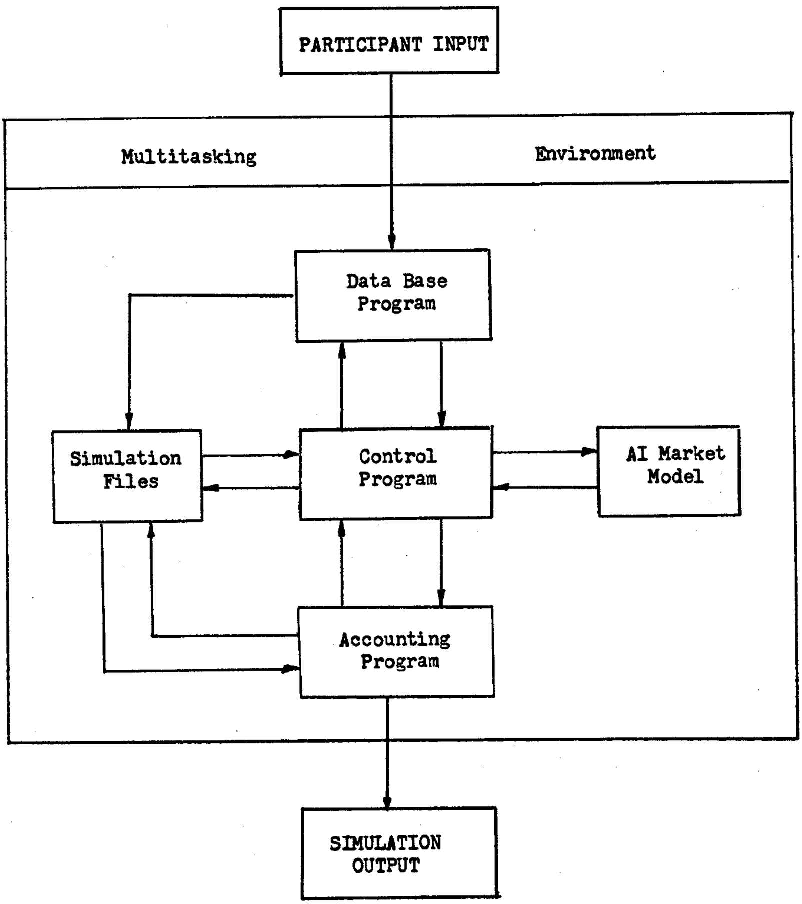
Of course, there is nothing fundamentally wrong with closed system simulations. They have been invaluable in accomplishing several business policy course purposes from both faculty and student points of view (Gordon and Howell, 1959; Patz, 1988a). Furthermore, as closed system simulation sophistication grows, so does the standing of this technique as one of the key bulwarks of the business policy field (Patz, 1988b).

Nevertheless, there are important reasons for looking beyond closed systems. Pedagogy is one, and research is the other. The purpose of this chapter, in short, is to explain why. Said in another way, the basic idea of this chapter is to argue for *open system simulation* research. This is an approach that does not rely upon fixed algorithms, and its applications are not limited to the business policy course.

The main cost of this gain in power will be an increase in simulation complexity that is several orders of magnitude beyond current models. As shown in Figure 10-1, even the most basic systems will require a multitasking environment running several programs simultaneously.

At the heart of open system simulations or, more generally, *artificial envimnments* is a control program that monitors data base operations, tracks all the usual accounting functions, maintains the simulation files, and. most important, directs the open system market model. As currently conceived, these market models will have an artificial intelligence (Al) base that generates market dynamics and eliminates the need for fixed or predetermined market algorithms.

FIGURE 10–1
A Generic Open Systems Design



Notice also that participant interactions with an open system are through a data base, allowing complex *decision rule* inputs rather than simple statements of prices, quantities, and production capacities. In other words, such a simulation may run continuously with participants entering new decision rules at their discretion or as indicated by current market conditions. Likewise, the output may be continuous generating, for example, the specific simulation's version of a daily *Wall Street Journal.* Overall, this means that simulations may assume the day-to-day character of ongoing businesses while encouraging the development of long range strategies.

These and other basic open system interpretations are discussed in more detail later in the chapter. For now, simply note that an emphasis on simulations stresses the pedagogical use of open systems. The more general emphasis on artificial environments (AE), using more elaborate Al models, stresses the equally important research possibilities of such systems. In fact, when open systems are finally made available, they are a way to challenge and improve our assumptions in the behavioral, economic, and policy sciences (Mitroff and Emshoff, 1979).

ASSUMPTIONS AND CLOSED SYSTEM EXPERIENTIAL LEARNING

There are three reasons for taking this open system point of view. First, regarding the day-to-day integration problems with simulations and other coursework, it is assumed that simulation technologies will have a far greater effect on what happens in classrooms than pedagogical sciences will have on simulations. This perhaps nonintuitive notion, that technology drives science more than science drives technology, is a well-known result of technological innovation research (Sahal, 1981). In other words, simulations and related computer technologies will assume an ever increasing importance in the classroom. The handwriting is already on the wall, but any further development of this topic is an issue for another discussion.

Second, closed system policy simulations provide students with an opportunity to deal with the fundamental interrelatedness of several business functions, but they do so in a fashion that is unique to the algorithms contained in the specific game. Policy simulations are “wired,” and there is little opportunity for participants to act upon rather than just react to the preset macroeconomic and commodity demand functions. While it is true that the participants’ collective decisions can counteract basic trend and demand conditions, their effects still lie within limits set by the established algorithms. Unlike real general managers, they do not create a competitive environment within a basically open system.

Third, closed system simulations do not provide a realistic data base for use in participant decision-making. Various well-developed codes (Keys and Leftwich, 1985; Scott and Strickland, 1985) generate a wealth of administrator and participant information. However, they ignore such crucial issues as possible new entries into the simulated industry, trade in markets other than those initially defined by the rules of the game, and *specific* administrator interventions.

New entries and new markets are almost always limited in closed system simulations. The number of markets is a “given,” and the number of firms is an administrator defined parameter. Likewise, the administrator is highly constrained regarding such issues as regional taxes, shipping charges, cash transfer or exchange rates, and team-by-team negotiations on any of these issues, including mergers and participation in different markets.

The opportunity for an administrator to participate as an informed, education-oriented competitor does exist simply by defining the number of teams to be (N + 1) where N = the number of student teams. But, decisions made by the administrator have to be based on the assumption that team (N + 1) is a going concern. Any other criterion will usually distort overall competitive results. In short, when a simulation's algorithms are fixed in advance, there simply is little room for maneuvering. Equally important, there is little room for research.

EXPERIENTIAL BACKGROUND

This disposition, of course, reflects something other than casual observation. It relies upon more than ten years of actual classroom experience with policy simulations. The games used range from an early microcomputer model (Nordstrom, 1972) to one of the most complex mainframe applications (Thorelli and Graves, 1964). In addition, the students involved range from undergraduate business majors to MBA's and practicing executives from the United States, Western Europe, Japan, and Southeast Asia. A list of the simulations employed is shown in 'Table 10-1.

This experience has been opportunistic because the search for appropriate packages has been limited to the ones made available by colleagues and publishers. No systematic evaluation of alternative total enterprise games was attempted. Furthermore, the approach was nonstructured because the implementation and evaluation procedures used for each simulation, with few exceptions, were the ones provided by the authors.

On the other hand, the student evaluations were generally positive with executives expressing the highest degree of satisfaction and undergraduates being the least satisfied. In other words, the degree to which students accepted the various simulations as a learning experience appeared to show a positive correlation with age.

All of these disclaimers, however, are irrelevant as far as the argument in this chapter is concerned. The key point is that in almost all cases, students learned how to "game-the-game." In general, the sequence of events followed this pattern:

1. All students experienced confusion over the rules of the simulation and how to execute a set of decisions.
2. Good students resolved the confusion, built cash flow models, and began to understand the simulation prescribed interrelatedness rules among marketing, production, finance, and overall economic functions.

TABLE 10-1
Simulation Experience Base

Simulation	*Authors*	*Period of Use*	*Student Type*
COGITATE	Temple, Barker & Sloane	1986-	Executive
International Operations Simn.	Hans B. Thorelli & Robert L. Graves	1983-1987	MBA Executive
Manager	Jerald R. Smith	1985-	Undergrad
MICROMATIC	Timothy W. Scott & Alonzo J. Strickland Ill	1986-	MBA
PLANETS II	U. S. Army Logistics Management Center	1982-1983	MBA
TASK	Alan J. Rowe	1980-	Undergrad
The Executive Simulation	Bernard Keys & Howard Leftwich	1985-	Undergrad
The Multinational Management Game	Alfred G. Edge, Bernard Keys & William E. Remus	1981-	Undergrad Executive
Top Managment Decision Game Executive	Joseph Nordstrom	1979-	Undergrad
USC Management Strategy Simo.	Paul A, Gruendemann	1972-1986	MBA

3. Average students witnessed the success of good ones, copied their procedures, and became more competitive.
4. Most students now “knew-the-rules” and played accordingly.
5. Poor students complained.
6. Minor changes in team standings occurred as students honed their gaming techniques.
7. Overall, there was no obvious correlation (based upon experience, not data) between the above references to good, average, and poor policy simulation performance and performance on case analyses, industry analyses, and specific company analyses.

Actually, more work needs to be done on this last point since some of the relevant data have been collected during the past few years. But, this is also a subject for another discussion. The key issue is that the first six findings are consistent over the ten-year period, three types of students, and different cultures. In fact, they are remarkably consistent, including a reasonable degree of satisfaction over all students even though the older ones express more interest.

EXPERIENTIAL INFERENCES

Assuming that these experiences are common among experienced simulation users, four important conclusions follow:

1. A relatively common technology, closed system simulations, produces common results in the long run. Students learn how to game-the-game.
2. Further empirical investigations aimed at arbitrary measures of student learning seem destined to repeat the same conclusions. They will reveal that this simulation or that one produces superior or inferior learning on one set of concepts or another.
3. It would appear that student interest in general management would be a more important target for closed system empirical research (Patz, 1988a). The question is: How can simulations be used to generate and nurture interest in general management as a career objective?
4. Other simulation technologies, for example, open systems, need to be developed if the full potential of simulations is to be exploited.

One part of the problem, especially in regard to the first two of these conclusions, is related to the process of closed system simulation design. That is, most simulations are developed because someone has an idea how to program:

1. An accounting sequence from the purchase of raw materials to the sale of finished goods.
2. An interesting demand function that goes beyond prices and advertising to include such factors as R&D, product differentiation, marketing efforts, and various leads and lags in these factors.
3. A special topic such as PERT or product life cycles.

Once programmed, including the basic 90% of the code concerned with accounting for each participant or team's position, the simulation is tested on a sample population, debugged, and retested. Finally, a user's manual is written to convey the necessary operating details to the uninitiated.

Herein lies the problem. The user's manual was not written first (Andersen, 3986). What people are supposed to do when participating in the simulation, playing the game, is decided after the fact of simulation design, not before.

Said in another way, the real purpose of the simulation is decided after software design problems have been solved. Typically, computer coding drives simulation purposes rather than the reverse. Thus, it is a small wonder that short-term research results lead to equivocal conclusions on simulation usefulness. Simulation purposes, for the most part, are decided by coding convenience rather than pedagogical, conceptual, or theoretical relevance.

In short, open system simulations need to proceed in the reverse order of closed system ones. That is, they need to begin with the user's manual, a statement of what is desired.

OPEN SYSTEM PRELIMINARIES

What is desired, of course, are basic statements of what needs to happen in an open system simulation. In particular, for policy simulations, answers are needed for the following questions:

I. What kinds of decision-making behaviors need to occur, and how will they be recorded in a fashion suitable for computer processing?
2. What kind of data base needs to be provided to participants prior to the decision-making sessions?
3. Can flexible computer routines be written that allow decision process modeling in a matter of minutes rather than days?
4. If the preceding problems can be solved, how can the models applicable to several participants or teams be made interactive?
5. Similar to closed systems, what sort of external constraints are required?

These are difficult questions, and answers will not be found easily. On the other hand, state-of-the-art simulation (Lee, 1987; Pritsker, 1986) and computer science (Hayes-Roth, 1985; Nii, l986a, 1986b) developments suggest that reasonable answers are within possible-bounds. Equally if not more exciting, these developments open new research and theory avenues.

A RESEARCH PREVIEW

Assuming for the moment that these "possible-bounds" are realistic, allowing for the development of open systems, research and theoretical interests may assume roles at least as important as pedagogical interests. The reason is simple. Open system simulations, by definition, do not depend upon preprogrammed algorithms. Rather, *they generate environments* that emerge over time as a result of participant behaviors. In other words, behavior can be observed under dynamic rather than relatively stable environmental influences Furthermore, in dynamic environments, the administrator may become one of the participants instead of acting as a referee over participant actions.

Considered in one fashion, this means that the heart of an open system simulation is a program or "driver" that generates the algorithms governing a particular run of the simulation. Then, based upon the results of the run, another algorithm set is generated. Associated with the driver is a data base generator that provides an environmental description for use by the participants. And third, another program combines the current algorithms with the data base and produces the results for the participants. This basic paradigm is shown in Figure 10-2.

Considered in a second fashion, an open system is a cross between many of the current activities in experiential laboratories and closed system simulations. Much of what happens in an experiential laboratory is for learning by observation purposes. Group and interpersonal phenomena such as decision-making, role differentiation, and communication can be observed firsthand as a practical test of fundamental motivation and perception theories. Moreover, basic changes, especially in group structures and processes, can be observed as they emerge over time.

On the other hand, policy simulation decisions are usually done in a rather "secret" environment. Participants have to infer from published data what the behavior of competitors means, and they are making their inferences with a *static* set of algorithms. Their basic environment, unlike the power-affect-status structures in group behavior, does not emerge over time.

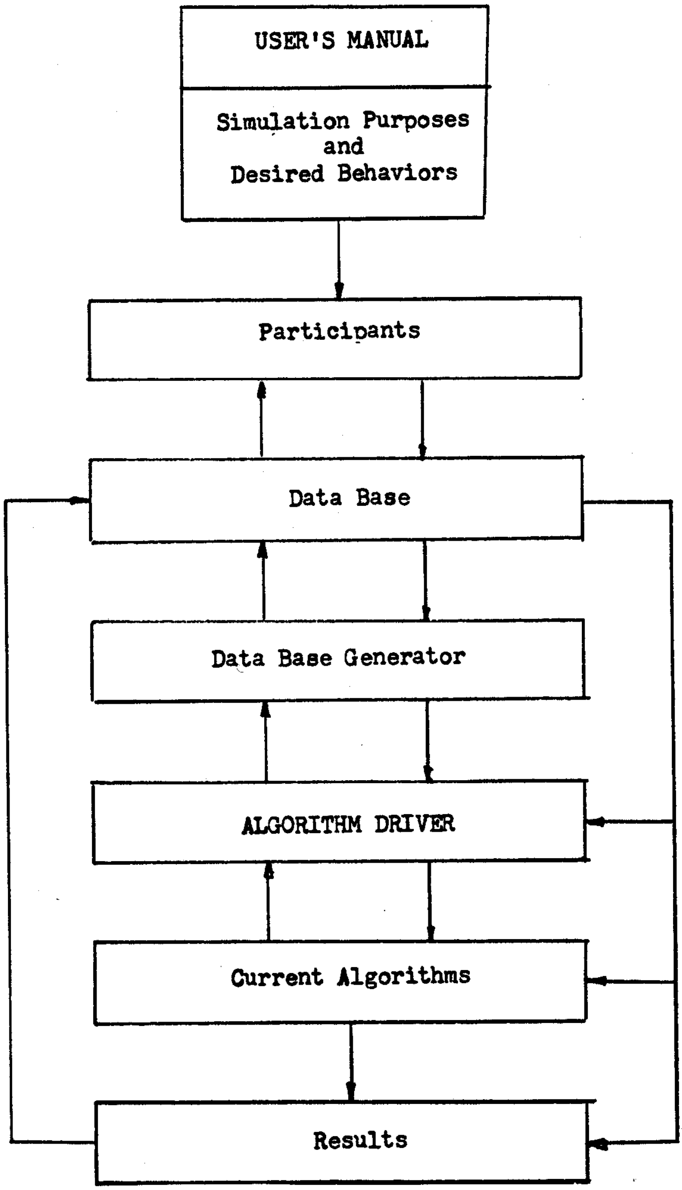
What is needed is a reasonable combination of the two. The payoff from such an amalgamation is the development of entirely new research methods and research areas that can be investigated in the simulation mode. Games will always be important as pedagogical tools, as they have been for centuries. But they have an equally, if not more important, role in research.

In fact, the list of research possibilities in dynamic, computer generated environments is endless, limited only by imagination (Emshoff and Sisson, 1970). For example, open system simulations can be used to study:

1. Market structure and dynamics in an unconstrained rather than algorithmically circumscribed environment.
2. Decision-making that leads to the creation of markets.
3. Collaborative or collective as well as competitive strategies.
4. Leadership tactics in a laboratory (classroom or experiential learning center) that approximates actual business conditions.
5. Group structure and dynamics as competitive conditions change.
6. Decision-making in environments where competition is conspicuously absent, such as large scale projects, again using the classroom or experiential learning center as a laboratory.
7. Resource allocation decisions in competitive environments that are threatened with new entrants, a feature absent from most simulations.
8. True product development and market introduction tactics beyond a known set of predetermined simulation alternatives.
9. Service industry rather than product industry phenomena.

One way to summarize this potential is shown in Table 10-2. The two key dimensions in this table are the simulation environment and the phenomena

FIGURE 10–2
A Basic Open System Design



under consideration. Pairing them, or considering the various combinations, suggests a large number of research possibilities for open system simulations. For example, very little is known about decision-making patterns in planned environments such as those common to large scale projects (Patz, 1986). A similar statement can be made regarding decision-making in collaborative environments such as those found in joint ventures. New entry threats in competitive environments have already been mentioned.

Moving towards more behavioral issues, leadership styles and group dynamics are obvious phenomena for open systems study. Much of the current literature in these areas depends upon highly constrained laboratory studies or public phenomena that are difficult to measure. An open system simulation would have the advantages of both approaches and minimize the disadvantages. In short, a simulation that allows behavior to emerge has the features of ordinary social behavior, but it does so along various prescribed dimensions similar to a laboratory experiment. Thus, the measurement of social phenomena is possible without overly constraining it.

Other phenomena of interest would include technological (product and process) innovation and high technology (Patz 1981). The research issues in this case would include resource allocation patterns and risk taking behavior in different kinds of open system environments.

As already mentioned, the list is endless. At a minimum, the three types of environments shown in Table 10-2 are only a beginning, as are the areas suggested for study. Once open system simulations are available, something like Table 10-2 can be expanded at will.

TABLE 10-2
Some Examples of Simulation Based Research Projects

	Open System Environments		
Phenomena	***Planned***	***Collaborative***	***Competitive***
Decision-Making Patterns	Large Scale Projects	Joint Venture Management	New Entry Threats
Leadership! Management Styles	Bureaucratic	Political	Strategic
Group Structure/ Dynamics	Power/Affect/Status Structures Social Roles and Social Systems Exchange/Reward Patterns		
Technological Innovation/ High Technology	Resource Allocation Patterns Risk Taking Behavior Success/Failure Patterns		

Some Research Caveats

Of course, some basic design efforts will be necessary to develop open system simulations. Just like the suggested issues than can be researched with them, they comprise an entire set of research issues. Nevertheless, some design issues are clear and they will be presented in the next section. For now, however, and at the risk of oversimplification, a few distinctions need to be made regarding open systems and other research endeavors such as artificial intelligence (Al), expert systems (ES), and decision support systems (DSS). The basic question is: Are open systems comparable to Al, ES, and DSS? Moreover, the basic answer is: No!

No doubt, it would be comforting to draw from other research endeavors such as Al, ES, and DSS. But, the problem is that none of them meet the necessary criteria even though each may benefit from open system simulation designs.

Al, for example, is an attempt to design computer hardware and software that learns from experience (Winston, 1984). Such learning may mimic human behavior, be superior to it, be altogether different from it, or be some combination of these alternatives. Any one of these results is desirable The idea is to create machines that act intelligently, machines with programs that reprogram themselves based upon experience.

Open system simulations, however, are qualitatively different. Here the idea is to use fundamental Al techniques (Tanimoto, 1987), create environments that can be recorded in a computer, and study people's behavior in the artificial environments (AE's) created by the system. Presumably, any intelligent system could participate in such a game, but Al is a very long way from duplicating even the most primitive human characteristics. A focus on artificial environments is much more practical for now.

Even more remote, ES does not attempt to approach the basic creativity problem. Like much so-called business policy research (Hambrick, 1983), it only tries to duplicate decisions. The basic paradigm has three steps, however complicated the intervening machinations: (1) program an expert's decision rules until a computer reaches the same decisions as the expert did in the past, (2) test the decision rules on new data, and (3) recycle until the program and the expert reach the same decisions on further new data. There is not much environmental creativity in the process.

Most remote, DSS is not intended to create an environment. Its key and very useful purpose is to help a decision-maker model an environment. Such models help in the solution of accounts receivable, inventory, cost of capital, and similar decisions, but they only reflect an environment. At best, they help in resource allocation decisions (Humphreys and Berkeley, 1986); at worst, they are super spreadsheets for a dull afternoon's entertainment (Keen, 1986).

SOME BASIC OPEN SYSTEM DESIGN CONSIDERATIONS

With these distinctions in mind, and referring again to the open system policy simulation questions noted at the beginning of the preceding section, answers can be suggested regarding open system designs. To repeat, the key questions are concerned with the decision-making behaviors, data base needs, computer routines, participant interactions, and external constraints characteristic of open systems.

Looking first at desired decision-making behaviors, or writing the user's manual first (refer to the top of Figure 10-2), it is clear that the traditional participant specifications of prices, production quantities, plant capacities, several marketing stipulations, and financial arrangements need to be enhanced. Various textbook models work in this fashion, but the workaday world does not (Quinn, Mintzberg and James. 1988). Sets of rules are much more important in competitive, decision-making circumstances (Cyert and March, 1963).

For example, prices are not specified precisely. Instead, limits are provided to salespeople governing what the prices will be under different buyer and competitor circumstances. Large industrial orders command lower wholesale prices, and minimum retail prices are "suggested" by almost everyone in the value added continuum. More generally, prices are simply negotiated. In the "long run," they may conform to demand and supply curve models. However, in more well-defined time intervals, decision rules are a mainstay of business conduct (Simon, 1957).

Similarly, decision rules are the basic "inputs" by participants in an open system simulation. Moreover, just as decision rules enhance the concept of decision specifications, the concept of decision inputs to a simulation needs some enhancement. This notion, shown in Figure 10-2 as the interaction between the data base itself and the results that update it, will be elaborated shortly. For now, in order to be more concrete regarding decision rules, two examples may be as follows:

1. Lower price by 10% if the firm's price exceeds the market average by more than 15% for two consecutive weeks.
2. Expand Capacity by 10,000 units when demand reaches 85% of existing capacity as long as the forecasted growth in GNP exceeds 4%.

DATA BASE CONFIGURATIONS

Decision rules, such as the two just noted, become simulation inputs only in the sense that they become part of the simulation's data base. In other words, participants are always working within and on the open system's data base. Working within the data base means that they are supplying the types of rules just noted (Widmeyer, 1987). Working on the data base means that they are analyzing the information made available to them, including their decision rule results (Gruendemann, 1987).

In this sense, a policy simulation becomes a continuous rather than a discrete process. Decisions rules may be entered at any time as part of the data base. and competitive results continue to occur whether or not any decision rule changes have been made. In fact, it is possible to design a standard closed system policy simulation using these concepts. All that is required is shown in Figure 10-2, omitting the algorithm driver and data base generator. Once again, the current algorithms would remain “wired,” but decision rules and a data base would be used rather than decision specifications and discrete inputs.

Practical applications of Continuous closed or open system simulations will require the posting of results on a routine basis, just like the finance section of a daily newspaper. Such continuous monitoring, however, is not a problem. Similar to current closed system policy simulations, all that has to be decided is the schedule for producing and displaying results.

COMPUTER ROUTINES

The algorithm driver and data base generator, of course, change a closed system to an open one. They modify the data base and algorithms that produce simulation results. What does this imply?

Any answer to this question probably will have several parts, and each multipartite answer can begin from many different Orientations. Even an incomplete enumeration of the alternatives, along with a discussion of the associated advantages and disadvantages. is an entire topic by itself. Therefore, only one view will be elaborated here, the one that begins with a focus on the data base generator and its associated data base. In this regard, four key points are of interest-access, security, data base contents, and interactions.

First, simulation participants have access only to the data base, not the generator that produces it. Second, this requires two kinds of security, one on the generator and the other on the data base. Similar to standard simulations, only portions of the entire data base can be accessed by a given participant. These areas would include the information generated by a participant for processing by the data base generator, information that has been “purchased,” and information that it made available routinely by the simulation as it proceeds. Obviously, assess to the generator itself would permit “data-snooping” and access to the proprietary information of other participants.

Third, the data base would include several different kinds of information, all managed by the data base generator. As already suggested, this would include each participant's current decision rules, current and historical simulation results, market and competitive data. and the usual income statement, cash flow, and balance sheet summaries. Except for the rules and more or less continuous operations with the data base by each participant, this content is similar to most of the well-developed games now available.

However, open system data bases need to include the continuous versions of discrete decision-making tools available in closed systems. These are the spreadsheet planning models (Keys and Leftwich, 1985; Scott and Strickland, 1985) that are available as adjuncts to the normal decision inputs of participants. What would make these models continuous is the capability to translate planning model results directly into decision rules within the data base. Equally important, however, forecasting, scheduling, and other analytical models need to be part of the data base accessible by participants in order to estimate the inputs for the planning models. In other words, a complete data base provides the capability to work through analytical models to planning models, decision rules, and the observation of results as well as other market and competitive information.

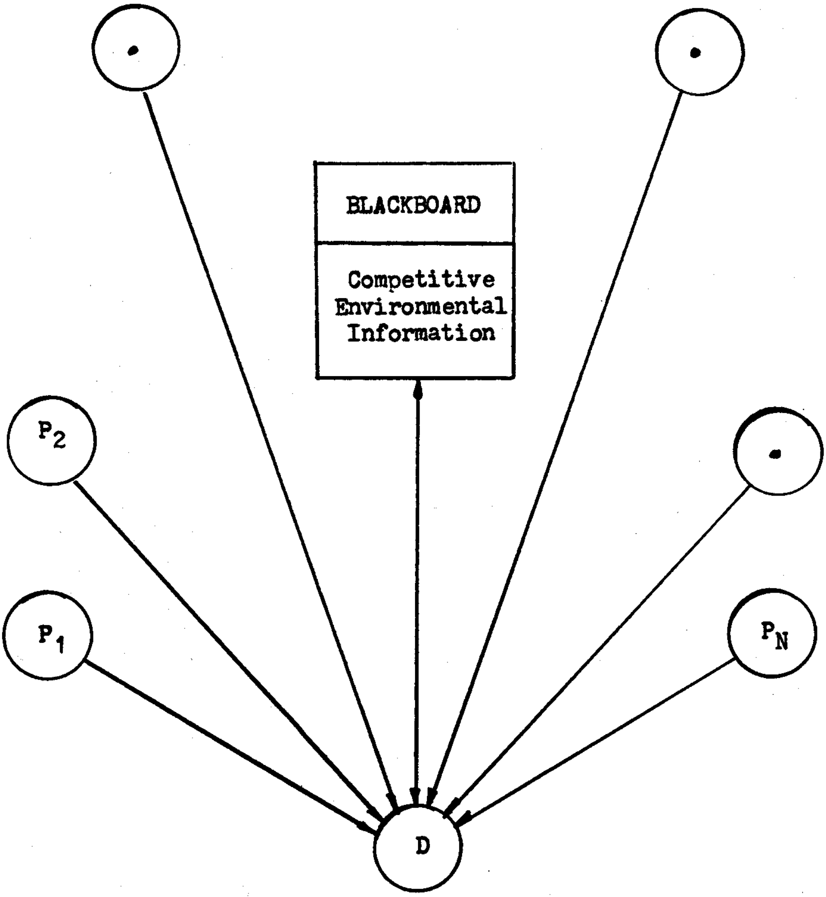
Fourth, the data base generator interacts with the algorithm driver as well as the data base. The key purpose in the relationship with the driver, in addition to the transmittal of competitive environmental information, is to transmit the current and historical sets of participants' decision rules. The environmental information and the rules, using the computer scientist's concept of a blackboard (Nii, 1986a, l986b), constitute the algorithm driver's basis.

THE ALGORITHM DRIVER

This basis is shown schematically in Figure 10-3 (Gasser, 1987). The accumulated sets of participants' decision rules are indicated as P1 through PN, and they surround the competitive environmental information, which is the basic blackboard. Another very key set of rules, indicated by the letter D in Figure 10-3, is the actual algorithm driver.

The driver in turn has several options in deciding the current algorithms (refer again to Figure 10-2). For example, it could select participant decision rules in an opportunistic fashion. That is, a single question would be asked:
Given the blackboard's current status, which set of participant rules, if applied first, would lead to the "best" outcome? In other words, each participant's rules would be applied to the data and a ceteria paribus outcome determined by holding all other decision rules constant at their previous status. The gain for each participant would be determined in the usual fashion combining measures of growth. profitability, and cash flow.

FIGURE 10–3
A Blackboard: An Algorithm Driver's Basis



Whatever set of participant rules provided the maximum gain would be applied first and change the blackboard. The same procedure would then be applied to all remaining participants until all decision rule sets operated on a current, updated blackboard and results determined. (Note that this procedure alone opens the system.) Each set of rules is applied to updated data because the blackboard changes after each application.

This is analogous to actual markets. Each competitor is subject to the opportunistic choices of rivals, and-contrary to most economic theories-no two participants ever work with the same set of data. In fact, those who make it their business to chronicle such events (Scherer, 1980) emphasize these points. Nevertheless, there are several other alternatives for an algorithm driver.

Decision rules can be combined by testing for and eliminating logical inconsistencies (Enderton, 1972). Then, market results would depend upon a shared or mutual understanding of competitive dynamics. Once again, this view is consistent with other theoretical positions (March and Simon, 1957).

Or, as a third alternative, each participant's inputs on a given dimension, such as price, could be considered as one variable in a linear constraint in N-dimensional space, where N = the number of participants. Thus, if there are five participants, the five prices form a constraint in a hyperspace of five dimensions. The five production quantities form another, as do all other decision rule dimensions.

Then, in linear programming terminology (Gass, 1975), all of these constraints form an N-dimensional convex polyhedron within which each participant's decision rules are represented. By mapping the polyhedron onto other (N + 1), (N + 2), . . . dimensions, results such as market share, growth, profitability, and cash flow can be determined.

The point is that all of these options and several others can be made available to an algorithm driver, can be selected by the driver in any kind of sequence, or can be applied according to a logic that a simulation administrator desires to demonstrate. This last point, according to a logic that a simulation administrator desires to demonstrate, is what permits open systems to be research tools. The investigator is no longer bound by the artificial constraints in laboratory settings and "natural" experiments. What is desired can be produced by programming the algorithm driver.

PARTICIPANT INTERACTIONS AND EXTERNAL CONSTRAINTS

All of these driver alternatives, as well as several others, open policy simulation systems. None of them depend upon fixed algorithms, and all of the possibilities suggested in the preceding paragraphs have some basis in theory and empirical research. Equally important, it is clear that participant interactions have been defined in a different fashion. They depend upon interactions with, examinations of, and modifications to a data base. They are not limited to a weekly specification of numbers on a decision sheet, whether or not it is computerized.

Likewise, participation includes the well-known game administrator as a participant, algorithm driver executive, or both. There is no practical limit to what can be defined as an algorithm driver once the concept is simply noted and put into practice. It is nothing more than operating on the same set of data in fashions, that while different, are mutually consistent.

However, there are some external constraints that must be kept in mind. First, extreme variations in driver alternatives, while interesting, may not serve many pedagogical purposes. Research interests are different, but only a few lessons may be learned within the confines of a semester. Second, any open system design must begin with a statement of purposes. This was noted earlier in the imperative to design the user's manual first. The point is that any open system, like any other system, will not satisfy all demands simultaneously. If anything at all has been learned from the designing of computer systems, it is that all of them are limited. Like theories, they have restricted application ranges.

For example, it helps to learn basic algebra before contemplating elementary calculus. Similarly, closed system simulations may be a good first step before open system ones. If nothing else, the comparison allows a distinction between basic closed textbook models and more realistic views of the competitive environment. A simple example of this last caveat is the difference between undergraduate and MBA education purposes in business policy. Closed systems are probably more than adequate for the former and open systems most appropriate for the latter.

Once again, however, research is a different issue altogether. Among the many purposes for investigating open systems as the basis for business-related investigations is that the appropriate environments cannot be devised in any other fashion. There just is no simple way to change competitive conditions, organizational arrangements, and control systems easily and in a sufficiently short period of time to observe behavioral changes. One of the great shortcomings of organizational research is that these shortcomings are usually insurmountable. In open system simulations, however, change is a matter of convenience. Culture can be defined rather than accepted.

CONCLUSIONS

This route to open system designs, through data base and blackboard technologies, is only one way to operationalize them. There are and will be others. For example, simpler open systems appear to be possible with simulation languages (Pritsker, 1986). Similar to the algorithm driver in Figure 10-1, interactive routines can be used to generate a new set of market behavior rules for each simulation execution. Such rules of course, would be based upon the participants' past and current decision rules.

Another way to think about open systems is in terms of "pure" algorithm drivers. This is a program that does not have to rely upon data base generators or participant decision rules. It generates dynamic market structures by varying algorithms and their parameters over time. Of course, a more realistic design would include participant behavior as an important determinant of market dynamics.

However the open system concept is operationalized, the important point is that they can be discussed in terms of current and developing technologies and realistic design efforts can begin. There is no reason to think of some distant future instead of the near term. Furthermore, open systems are a good example of where the often sought but seldom realized interdisciplinary research goal can be attained. For once, policy theorists have some common ground with decision and computer scientists.

Policy, however, is not the only business field that can benefit from the design and use of open systems. The focus in this paper is on policy simulations, but the same arguments apply to most other areas of business research and practice as well as several areas of engineering and science. Open systems apply anywhere the phenomena of interest can be described as emergent. That is, the various States of the subject chosen for study cannot be predicted over time given knowledge of initial states or conditions.

In fact, it is probably fair to say that the prediction of subsequent states given a knowledge of initial ones is not the main research focus in open ststems. Again. using the modifier emergent, interest is on the possibilities or alternative future states as some phenomenon, e.g., a policy simulation, develops over time. If prediction is a concurrent result of open system analyses, this is simply another research benefit. Nevertheless, the main focus is on behavioral possibilities rather than behavioral bounds or limits.

Research project examples that fit easily into the emergent phenomena category have already been listed in Table 10-2. Even for these few it is difficult to imagine any long term interest in closed system analyses. Consider for a moment the first topic listed in that table, decision-making patterns in large scale projects. Any Sort of complex behavior such as this can be constrained to a routine, closed pattern given access to sufficient resources. This Sort of finding does not help much with complex project management problems.

But, by simulating such an environment in an open system, where behavior can emerge and vary over time, insights into actual decision-making patterns can be achieved. First, a simulation allows the research to be done rapidly; and second, the participants can “behave” in an environment that develops over time rather than one constrained by artificialities dictated by the linear assumptions in experimental design models. In short, open systems will make pedagogy and research equal partners in the study and design of policy and several other kinds of simulations.