Proceedings of the 1986 Winter Simulation Conference
J. Wilson, J. Henriksen, S. Roberts (eds.)

EXPLANATION SYSTEMS FOR COMPUTER SIMULATIONS

David H. Helman
Center for Automation and Intelligent Systems
Case Western Reserve University
Cleveland, OH 44106, U.S.A.

Akash Bahuguna
Weatherhead School of Management
Case Western Reserve University
Cleveland, OH 44106, U.S.A.

ABSTRACT

Explanation systems supply information that clarifies the structure and problem domain of a computer program for the user. We begin our paper by describing the early explanation systems, which were built for expert system programs, and by reviewing some of the subsequent developments in artificial intelligence that relate to this area. The results of our research are consistent with some of the recent developments in artificial intelligence; we have found that there are a variety of kinds of information that are useful to naive users of computer programs. We have been particularly interested in writing programs that can supply such information to naive users of numerical computer simulations. We describe an implemented explanation system, NATURALIST, which explains the structure and domain of a simulation for inventory control. Our experience with the NATURALIST program suggests that explanation facilities may be valuable additions to numerical computer simulations.

1. INTRODUCTION

The concept of a symbolic explanation facility originated in artificial intelligence research of the early 1970s. The first explanation facilities merely repeated the propositions or clauses that were or could be proven in the course of a particular session with a rule-based expert system (Winograd 1972). If a user of such a system wanted to know, for example, how a program arrived at a conclusion C, the system would print the propositions that were proven in order to reach conclusion C.

A trace of the propositions that are or could be proven in the course of a particular session with an expert system program helps to clarify for users the control structure of the program. Subsequent artificial intelligence research has shown, however, that repeating the propositions that are or can be proven in an session with an expert system is only one technique for elucidating rule-based expert system programs (see Helman 1986). The research of Hasling (1984) and Chandrasekaran (1985), for example, develops techniques for manipulating and presenting knowledge pertaining to the justification of conditional chains of reasoning in rule-based expert systems. In these systems, the connections between nodes in the AND/OR graphs defined by the rules of the expert system are classified semantically. This semantic classification is based on the tasks these connections or inferential steps represent in terms of the expert reasoning processes which such programs simulate. Once such a classification is stored, a trace of system behavior can either repeat the propositions or clauses the system proved in a particular session or it can enumerate the expert tasks or strategies simulated by these chains of steps. The latter trace elucidates the control structure of the expert system program, and also clarifies the expert strategies which pertain to the given problem domain. This example is paradigmatic of recent artificial intelligence research pertaining to explanation. Such research has, in general, shifted from the question, "How do we trace the inferential steps of a program?" to the question, "How do we explain the problem domain of the program to a user?" (note Clancey 1983).

2. EXPLANATION FACILITIES AND SIMULATIONS

The focus of this study is the problem of writing explanation facilities for numerical computer simulations. The problem of writing explanation facilities for numerical computer simulations is analogous to the problem of writing such facilties for rule-based expert systems. An explanation facility for a numerical computer simulation can trace the steps of a simulation in a variety of ways (paralleling methods in artificial intelligence: note Heterick, Gerth, and Huebner 1977).

A major contrast between rule-based expert systems and numerical computer simulations is that in the latter there is no natural data structure that may be called "the smallest unit that requires explanation." The locus of explanation for rule-based expert systems is the clause or proposition. It may be noted parenthetically that programs which list chains of clauses or propositions may themselves be difficult to understand if the chains are long and complex. In these cases, it may be helpful to classify the nodes of the AND/OR graphs defined by the rules of an expert system by an abstraction hierarchy. That is, collections of nodes which have a great deal of structure may be explained, for some purposes, as individual units. In these kinds of programs the clause is still the basic unit of explanation. In numerical programs, however, it is not clear how the basic unit for explanation is to be determined: Does one explain each individual addition? In considering the problem of writing explanation facilities for computer simulations it is appropriate to abstract from the details of how an explanation system can trace the behavior of a program. We approach the problem of explaining computer programs as part of the study of the interface between the computer and the user, focusing on the kinds of explanations that (naive) users of computer simulations will find useful.

3. PHILOSOPHICAL BACKGROUND

Our theory of explanations, which is the basis for the simulation explanation facility NATURALIST (see Section 4 below), is derived from the literature in the philosophy of explanation. We have been particularly interested in the literature on genetic explanation (explaining an event by citing the history of the event - note Dray 1957), causal explanation (explaining an event with reference to the causes of the event - e.g., Salmon 1985), what-if explanation (explaining an event by contrasting it with what might have happened - Van Frassen 1980), functional explanation (explaining a variable or a component by elucidating its function within a larger system - see Hempel 1965), and how-possibly explanation (explaining an event by correcting the mistaken presuppositions of persons who do not understand the event - see Dray 1957).

Some philosophical accounts of explanation find that different aspects of explanation are central in distinct disciplines. For example, Hempel (1965) has noted that anthropology emphasizes functional explanations; the description of the function (within a larger context) of a particular rite or totem. The philosophical viewpoint the authors adopt in this study (naturalism) is that day-to-day explanatory behavior does not vary significantly from discipline to discipline. A physicist will also, for example, in his or her ordinary activities, talk of the functional role of an event or an apparatus. There are, furthermore, a number of distinct kinds of explanations that are appropriate in nearly all disciplines. In particular, the five kinds of explanations that provide the focus for this study (genetic explanation, causal explanation, what-if explanation, functional explanation, and how-possibly explanation) are used in a variety of contexts.

If our viewpoint correctly characterizes everyday explanatory activities, then an adequate explanation facility should be capable of providing a variety of kinds of information. By providing users with the kinds of information that human explainers typically make available, the NATURALIST program is intended as an illustration of such a facility.

Our method of investigation in this study may be contrasted with the methods used in work on cognitive process simulations (e.g., Kuipers and Kassirer 1984). If we had followed the cognitive process methodology in this study, we would have recorded what experts say as they explained their programs to naive users. We would then have derived our taxonomy of explanation types from these protocols.

The top-down approach adopted in this study (from a theory of explanations to specific applications) contrasts with the bottom-up approach of cognitive process theory. Both methodologies have their advantages. Cognitive process investigations gain insights through their in-depth studies of the thinking and problem-solving behavior of experts. They then extend these observations to general situations. The fundamental distinctions made in top-down or theoretical studies may be somewhat easier for naive users to understand. A complete comparison of these two methodologies is, however, beyond the scope of this paper.

4. NATURALIST

In order to (partially) test our theory of explanation, we needed to see if the kinds of explanatory information given in genetic, causal, what-if, functional, and how-possibly explanations could be supplied by a facility attached to a specific simulation. For this experiment, we used an inventory control model described in Gaither (1982). Gaither models inventory control as follows:

total material costs = annual carrying costs + annual ordering costs + annual acquisition costs + annual incoming transportation costs + annual carrying cost for safety stock + annual expected stockout costs

$$TMC = \frac{Q}{2}C + \frac{D}{Q}S + ac(D) + r(D) + (SS)C + A(S')\frac{D}{Q}$$

Variable Definitions

Q = fixed order quantitiy in units per order
C = carrying costs per unit in dollars per unit per year
D = annual demand in units per year
S = ordering or setup cost in dollars per order
ac = acquisition cost in dollars per unit
(this may be, for example, a constant function of Q)
r = incoming transportation cost in dollars per unit
(this may be, for example, a discontinuous function of Q)
SS = level of safety stock in units
A = probability of stockout in each reorder cycle
S' = stockout, reorder costs, etc., in dollars per stockout

In a simulation based upon this model, an initial estimate of the optimal order quantity (the Q that minimizes TMC) is made by calculating the quantity $1/2(\sqrt{2DS/C})$. Q is then incremented in a range determined by this initial estimate. For each Q tested, the simulation will determine the optimal safety stock (the SS that minimizes the sum of the annual carrying costs plus annual expected stockout costs) by varying the estimate of A. The sample partial output from our implementation of the Gaither Inventory Control Model (Figure 1 below) only indicates the optimal safety stock for each order quantity (Q):

Inventory Control Simulation for 1986 — Simulation Number 1 — Page 1

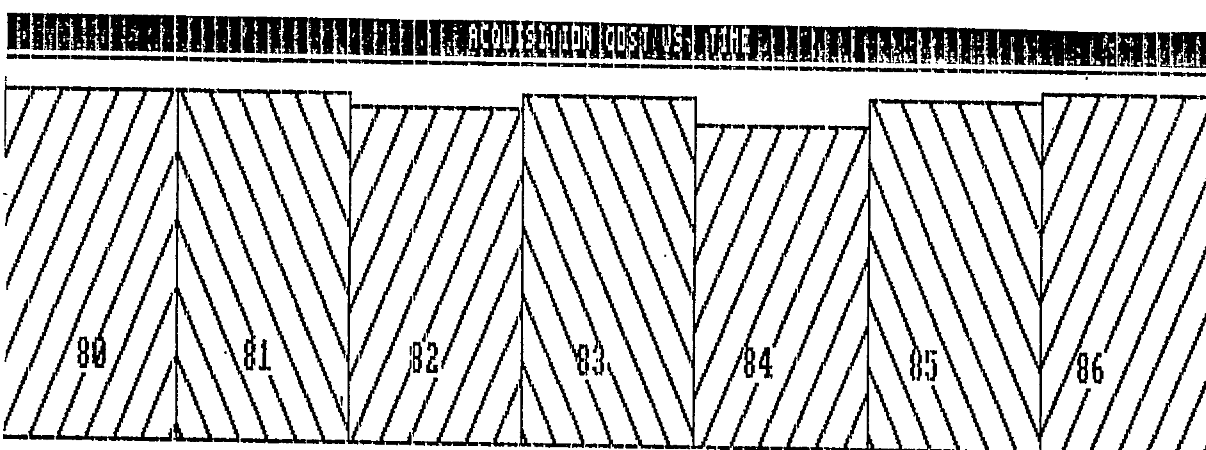
Order quantity (units)	Service level (1-alpha)	Safety stock (units)	Annual ordering cost	Annual carrying cost	Annual incoming transportation cost	Annual safety stock cost: Carrying cost	Annual safety stock cost: Expected stockout cost	Annual safety stock cost: Total cost	Annual acquisition cost	Total annual material costs
386	0.9953	118	1815.54	453.55	17955.00	277.30	29.87	307.17	268275.00	288806
394	0.9938	114	1778.68	462.95	17640.00	267.90	38.60	306.50	268275.00	288463
402	0.9938	114	1743.28	472.35	17325.00	267.90	37.83	305.73	268275.00	288121
410	0.9938	114	1709.27	481.75	17010.00	267.90	37.09	304.99	268275.00	287781
418	0.9918	109	1676.56	491.15	16695.00	256.15	48.12	304.27	268275.00	287442
426	0.9918	109	1645.07	500.55	16380.00	256.15	47.21	303.36	268275.00	287104
434	0.9918	109	1614.75	509.95	16065.00	256.15	46.34	302.49	268275.00	286767
442	0.9918	109	1585.52	519.35	15750.00	256.15	45.50	301.65	268275.00	286432
450	0.9918	109	1557.33	528.75	15435.00	256.15	44.70	300.85	268275.00	286097
458	0.9918	109	1530.13	538.15	15120.00	256.15	43.91	300.06	268275.00	285763
466	0.9918	109	1503.86	547.55	14805.00	256.15	43.16	299.31	268275.00	285431
474	0.9918	109	1478.48	556.95	14805.00	256.15	42.43	298.58	268275.00	285414
482	0.9918	109	1453.94	566.35	14490.00	256.15	41.73	297.88	268275.00	285083
490	0.9918	109	1430.20	575.75	14175.00	256.15	41.05	297.20	268275.00	284753

Figure 1: Partial Output From The NATURALIST Inventory Control Simulation

4.1. Genetic Explanations

A genetic explanation, when it is successful, makes sense out of a present event by citing a sequence of past events that led to it. For the concept of a genetic explanation to apply to the Gaither simulation, we must suppose that the simulation is run over an extended period of time, and that we have a record of the actual as well as the expected variable values. Graphing variable values provides some useful information, but such graphs will not tell us what to expect in the present time period, or why significant changes occurred when they did in past time periods.

The histogram in Figure 2 shows changes in acquisition costs for toasters over a period of seven years. During the first two years recorded, the user of the NATURALIST facility is asked to supply a text to accompany the recorded value for each variable. After the first two years, the user of the NATURALIST genetic explanation facility is prompted to enter explanatory text whenever a variable value differs significantly from the previous year.



1980: Cost of toasters as set by supplier
1981: No change
1982: Lower price from supplier due to new automated plant
1983: Prices returned to previous values
1984: Supplier drops prices, tries to stimulate demand
1985: Supplier raises prices to previous levels
1986: Shortage of heater coils

Figure 2: Genetic Explanation For Acquisition Costs

The genetic module of NATURALIST is best taken as one possibility that could be implemented within the next generation of genetic explanation facilities. The possibilities for advanced genetic explanation facilities are brought out in the following example of a genetic explanation:

> Why is the principal maternity hospital in the city of Alexandria located on the grounds of the navy arsenal?
>
> Genetic Explanation: By 1839 the Ottoman governor of Egypt had been at work for more than thirty years to equip himself with a fleet of warships in the Western style. He realized that his naval establishment would not be self-sufficient unless he had Egyptian workers build the ships, and they could only be trained by hired naval specialists from the West. Foreign specialists, however, were unwilling to come without their families, and they wanted to be sure of adequate health care. The Ottoman therefore hired Western physicians to attend naval experts and their families. The doctors found that they had extra time, and being community minded decided to aid the local Egyptian population. Maternity work was the first call and therefore a maternity hospital arose within the confines of the navy arsenal (Hempel 1965).

From a computational point of view, this example has two very interesting features. First, there is no reference, in the above genetic explanation, to the time intervals at which the key events took place. The time "grain" of the NATURALIST genetic explanation facility is, in contrast, set at one year and is not easily altered by the user. More advanced genetic explanation facilities would presumably be generic to some degree. Historical information pertaining to the simulation would still constitute a separate kind of information which the user could access, but the structure of this information could be easily tailored by the user of the simulation. In the object-oriented simulation programming language ROSS (McArthur, Klahr, and Narain 1986) there are commands that would allow a user to easily vary the "grain" of such a facility to suit his or her purposes. It would also be possible, in ROSS, to allow the user to abstract the sequence of important events from the dates of their occurrence, if these dates were not particularly important.

A second interesting feature of the above example is that the events described are "causally" related to one another. For example, we can understand how the desire of the Ottoman governor for a self-sufficient navy led him or "caused" him to establish a training program for Egyptian workers. In programs such as De Kleer and Brown (1984), Baskaran and Reddy (1984), and Widman (1986) the transitions between program states represent the causal interactions between parts of a system over a period of time. Thus, a trace of the changes in program states provides a causal as well as a historical account of the changes in a system. We might say, however, that such a program can only give "internal" genetic explanations. That is, the user of these programs only has access to a historical record of the variables that constitute the described system. The point of a genetic explanation facility is to allow a user to access exogenous variables when necessary.

In the NATURALIST system, we provide a way for the expert or the user to note exogenous variables which are important for an understanding of changes in variable values. Because we ask users to supply their own explanations of these changes, our approach seems to be at one end of a spectrum (from the perspective of artificial intelligence); at the other end of the spectrum, we find research that has been directed at the production of systems that supply their own explanations. Schank (1982), for example, presents an account of how intelligent systems can provide their own explanations of unexpected events. His approach cannot be sharply distinguished from our own, however, for it may be helpful as well in the design of systems that provide "canned" explanations.

4.2. What-If Explanations

A successful what-if explanation makes it easier to understand an event by contrasting the occurrence of the event with what might have happened (Van Frassen 1980). In our explanation facility for the Gaither simulation the concept of what-if explanations is implemented as a kind of spreadsheet. Once the user has run the simulation with a set of variable values, he/she may change any number of these values, and rerun the simulation. The system returns the effects these changes have on the six quantities (e.g., annual acquisition cost) that make up total material costs, as well as the the effect these changes have on total material costs. The user may also rerun the simulation after changing the functions of the model (the user can, for example, model transportation costs as a step, discontinuous, or constant function of order quantity).

4.3. Causal Explanations

A successful causal explanation makes sense out of a given event by citing the causes that led to that event. In the philosophy of science, the notion of a causal explanation has been closely tied to the notion of a general causal law. The classical philosophy of science theory is that an event E can be causally explained when E can be deduced from one or more general causal laws together with a statement of initial or antecedent conditions. Causal explanations, on this view, are in accord with the following schema:

C1, C2,......Ck	Statements of initial or antecedent conditions
L1, L2,......Lk	General laws
deduction	
→ E	Description of the event to be explained

Figure 3: The Structure of Causal Explanation According to the Classical Theory in the Philosophy of Science

Hempel (1965) gives the following example of a causal explanation that accords with this schema:

> To an observer in a rowboat, that part of an oar which is under water appears to be bent upwards. The phenomenon is explained by means of general laws--mainly the law of refraction and the law that water is an optically denser medium than air--and by reference to certain antecedent conditions--especially the facts that part of the oar is in the water, part in the air, and that the oar is practically a straight piece of wood. Thus, here again, the question "Why does the phenomenon occur?" is construed as meaning "according to what general laws, and by virtue of what antecedent conditions does the phenomenon occur?"

The causal explanation module in NATURALIST determines the causes of changes in total material costs from one year to the next. The causes of changes are determined in two steps. The first step of the NATURALIST algorithm for causal explanation is to calculate the effects that each change in variable value, on its own, would have had on total material costs (see Kosy 1984).

The second step of the NATURALIST algorithm for causal explanation is consistent with the classical view of causal explanation in the philosophy of science. We can, after the first step of causal explanation, know that the changes in acquisition costs, on their own, would have accounted for half of the change in total material costs. Why, however, did acquisition costs change as they did? To understand why acquisition costs changed, we must find a general principle or law which describes the change.

To supply the user with a general principle that accounts for changes in variable values, NATURALIST uses a backtracking rule-based reasoning program. The top-level goals of the AND/OR graphs defined by the rules of this program are alternative reasons for changes in variable values. For example, the top-level goals of some AND/OR graphs are the various reasons why acquisition costs might have risen. We have not, however, been able to produce a exhaustive classification of the reasons why variable values might change. There are cases where no conclusion can be drawn.

Goals can be proven in our reasoning program in one of four ways. The first two techniques are the standard methods for reaching conclusions in symbolic reasoning programs. First, a goal can be proven by deducing it from other goals that have been proven already. Second, a goal may be proven by asking the user a question, and getting the answer that is required to prove that particular goal.

In NATURALIST, we also use the simulation results and the information recorded in the explanation modules to prove goals in the AND/OR graphs defined by the rules of the reasoning module. Thus, the third method for proving goals is by finding that certain explanations have been entered in the genetic explanation module. That is, NATURALIST has a limited pattern-matching facility which searches for specific information that may have been entered by users as genetic explanations. Finally, we interleave the numerical simulation results with the AND/OR graph defined by the rules of the causal module. There are, in general, two ways in which numerical simulations may be interleaved with symbolic reasoning programs. First, a symbolic reasoning facility can reach qualitative conclusions. These can be mapped to numbers and then standard numerical simulation techniques may be applied (note Lee and Widman 1986). Second, conclusions in a symbolic reasoning program may be proven or not proven depending on the results of a numerical simulation (Clarkson 1963). The latter method is used in NATURALIST. Simulation results are used to help to discriminate the reasons for changes in variable values.

4.4. Functional Explanations

A successful functional explanation clarifies an event or a device by elucidating their roles in terms of the larger systems of which they are a part. The functional roles of an event or a device are, however, relative to the interests of the person who is studying the device. From one perspective, for example, the function of the fuel pump of a car might be its role in keeping the engine running; another person might elucidate the function of the fuel pump in terms of its role in keeping fuel consumption low.

The relativity of functional explanation is brought out when one gives this notion a computational interpretation. Consider, for example, several lines of code that are supposed to compute the square root function. From one point of view, these lines of code play a role in the numerical functions computed by a program. If these lines of code do not compute the square root function, the program will not work as it is supposed to. The lines of code that compute the square root function will also play a role in the control structure of the program. From this point of view, the function of these lines of code may be to receive values from one procedure and pass values to another procedure. Finally, one may characterize the functions of these lines of code in terms of the real-world sub-system that the code is supposed to model, and the relationship of this sub-system to the larger system which is modeled by the program as a whole.

To simplify matters somewhat, we may interpret the notion of a functional explanation, in the context of a computer simulation, as an explanation by trace. That is, a functional explanation traces the relationships between one part of a program and the rest of the program. One can, however, trace a program at any number of levels. In NATURALIST, the functional explanation module traces through a symbolic explication of the Gaither numerical model. Figure 4 shows a functional explanation NATURALIST produces for the computation of acquisition costs:

Functional Trace of Acquisition Costs

```
    Annual Carrying Cost
+   Annual Ordering Cost
+   Annual Acquisition Cost
+   Annual Incoming Transportation Cost
+   Annual Carrying Cost for Safety Stock
+   Annual Expected Stockout Cost
-----------------------------------------
=   Total Annual Material Costs
```

Acquisition Costs contributes to Annual Acquisition Costs.

Annual Acquisition Costs = Acquisition Cost * Demand

delta Acquisition Costs	= 2%
delta Demand	= 4%
delta Annual Acquisition Costs	= 6%

This change in Annual Acquisition Costs made a linear contribution to the rise of Total Material Costs (TMC)

Figure 4: Functional Explanation Produced By Naturalist for Acquisition Costs

The ideal functional explanation facility would allow the user to trace the behavior of a simulation on a number of separate but important levels. We may say, alternatively, that simulation explanation facilities would allow the user to verify the behavior of simulations by a number of distinct methods (e.g., tracing and testing numerical algorithms, tracing control structure, tracing the relationships between program structures which represent objects in the expert's model of the problem domain). Research pertaining to object-oriented programming environments has made progress in this area (Ruiz-Mier, Talavage, and Ben-Arieh 1985); still, much work remains to be done.

4.5. How-Possibly Explanations

An explanation is often required when a person finds that an event is unexpected; i.e., when he/she cannot understand how the event could have occurred. In these cases, an explanation should uncover the assumptions underlying the confusion of the person who did not expect the event in question, and then show either that these assumptions are false, or that they do not warrant the conclusion that the event could not have occurred (see Dray 1957). How-possibly explanations do not form a completely distinct category of explanation. A causal or a genetic explanation of an event, for example, may be required to remove the mistaken presuppositions of the person who finds an event unexpected. An explanation is a how-possibly explanation if it removes confused presuppositions; this basis for categorization is, perhaps, orthogonal to the logical and semantic features that distinguished what-if, causal, genetic, and functional explanations.

To implement the ideal how-possibly explanation module for a computer simulation it would be necessary to model the knowledge and presuppositions of particular users of the simulation. If this modeling task is too difficult or time-consuming in a specific context, it is still possible to identify the mistaken presuppositions that are most significant with respect to a particular computer simulation. A how-possibly explanation facility would, following this methodology, supply information that would remove the presuppositions of the user that are most likely to be mistaken or that are most likely to cause problems.

The how-possibly explanation module for NATURALIST is in the process of being implemented. In the design of this module, we have been particularly interested in giving the naive user access to information that pertains to the limitations of the NATURALIST simulation. The NATURALIST simulation equations for transportation costs, for example, cannot reflect month-to-month differences in the cost of shipping a unit. By making such limitations explicit, a how-possibly explanation module may play a role in the validation of a simulation. That is, such a facility reduces the practical significance of disparities between the simulation model and the real-world system that is being represented by the simulation model (note Bratley, Fox, and Schrage 1983).

5. CONCLUSIONS AND FUTURE DIRECTIONS

On the basis of this study, it appears that explanation facilities can be useful additions to simulation programs. Explanation facilities seem particularly helpful in situations where a novice needs to understand and use a complex numerical simulation. A future direction for this work is to test this hypothesis empirically. With Professor Elizabeth Short of the Case Western Reserve Department of Psychology, we have designed several experiments to explore the relationships between increased understanding of models and the use of explanation facilities.

Another direction of this research pertains to the design of simulation explanation facilities that present alternative models of a problem domain. We are presently, in collaboration with Professor Leon Sterling of the Case Western Reserve Department of Computer Engineering and Science, developing a program that presents alternative models for inventory control.

ACKNOWLEDGEMENTS

We would like to thank Drs. Miles Kennedy, Yoh-Han Pao, Elizabeth Short, Leon Sterling, and Lawrence E. Widman. We also thank Jeffrey L. Bennett and Antony W. Foster.

REFERENCES

Baskaran, V. and Reddy, Y.V. (1984). An introspective environment for knowledge based simulation. In: Proceedings of the 1984 Winter Simulation Conference (S. Shepard, U. Pooch, and D. Pegden, eds.). Institute of Electrical and Electronic Engineers, Dallas, Texas, 644-651.

Bratley, P., Fox, B.L., and Schrage, L. (1983). A Guide to Simulation. Springer-Verlag, New York.

Chandrasekaran, B. (1985). Generic tasks in expert system design and their role in explanation of problem solving. Technical Report, Laboratory for Artificial Intelligence Research, Ohio State University, Columbus, Ohio.

Clancey, W. (1982). The epistemology of a rule-based expert system -- a framework for explanation. Artificial Intelligence 20, 215-251.

Clarkson, G.P.E. (1963). A model of the trust investment process. In: Computers and Thought (E.A. Feigenbaum and J. Feldman, eds.). McGraw-Hill, New York.

de Kleer, J., and Brown, J.S. (1984). A qualitative physics based on confluences. Artificial Intelligence 24, 7-83.

Dray, W. (1957). Laws and Explanation in History. Oxford University Press, Oxford, England.

Gaither, N. (1982). Using computer simulation to develop optimal inventory policies. Simulation 39, pp. 81-87.

Hasling, D.W., Clancey, W.J., and Rennels, G. (1984). Strategic explanations for a diagnostic consultation system. International Journal of Man-Machine Studies 20, 3-19.

Helman, D.H., Bennett, J.L., and Foster, A.W. (1986). Simulations and symbolic explanations. In: Proceedings of the International Symposium on Methodologies for Intelligent Systems. ACM Sigart Press (to appear).

Hempel, C.G. (1965). Aspects of Scientific Explanation. MacMillan Press, New York.

Heterick, R.C., Gerth, J.A., and Huebner, N.D. (1977). In: Proceedings of the 1977 Winter Simulation Conference (H.J. Highland, R.G. Sargent, and J.W. Schmidt, eds.). Institute of Electrical and Electronic Engineers, Gaitherburg, Maryland, 558-567.

Kosy, D.W. and Wise, B.D. (1984) Self-explanatory financial models. In: Proceedings of the National Conference on Artificial Intelligence (R.J. Brachman, ed.). The American Association for Artificial Intelligence, Austin, Texas, 176-181.

Kuipers, B. and Kassirer, J.P. (1984). Causal reasoning in medicine: analysis of a protocol. Cognitive Science 8, 363-385.

Lee, Y.B. and Widman, L.E. (1986). Reasoning about diagnosis and treatment in a causal time-varying domain using semi-quantitative simulation and inference. American Association for Artificial Intelligence Workshop on Artificial Intelligence and Simulation, Philadelphia, Pennsylvania.

McArthur, D.J., Klahr, P., and Narain, S. (1986). ROSS: an object-oriented language for constructing simulations. In: Expert Systems: Techniques, Tools and Applications (P. Klahr and D.A. Waterman, eds.). Addison-Wesley, Reading, Massachusetts, 70-91.

Ruiz-Mier, S., Talavage, J., and Ben-Arieh, D. (1985). Towards a knowledge-based network simulation environment. In: Proceedings of the 1985 Winter Simulation Conference (D.T. Gantz, G.C. Blais, and S.L. Solomon, eds.). Institute of Electrical and Electronic Engineers, San Francisco, California, 217-222.

Salmon, W.C. (1984). Scientific Explanation and the Causal Structure of the World. Princeton University Press, Princeton, New Jersey.

Schank, R. (1982). Dynamic Memory. Cambridge University Press, Cambridge, England.

van Frassen, B. (1980). The Scientific Image. Oxford University Press, Oxford, England.

Widman, L.E. (1986). Representation method for dynamic causal knowledge using semi-quantitative simulation. In: Proceedings of MEDINFO 86 (to appear).

Winograd, T. (1972). Understanding Natural Language. Academic Press, New York.

AUTHORS' BIOGRAPHIES

David H. Helman is an assistant professor in the Department of Philosophy and a staff member in the Center for Automation and Intelligent Systems Research at Case Western Reserve University. He received a B.A. in philosophy from The University of California, Berkeley in 1978, and the Ph.D. degree in philosophy from Harvard University in 1983. He has been a visiting scholar at the Center for the Study of Language and Information, Stanford University (1985). His current research interests in artificial intelligence include explanation facilities for simulation programs, the use of simulation in communications programs, and the applications of expert systems to processes for collaborative composition and planning. He is a member of AAAI and ACM.

Akash Bahuguna is a Ph.D. student in the Weatherhead School of Management at Case Western Reserve University. He received a B.Tech. in Electrical Engineering from the Indian Institute of Technology at Kanpur (India) in 1981, and an M.B.A. from Indian Institute of Management at Calcutta (India) in 1985. His current research interests include cognitive psychology, expert systems, decision support systems, and artificial intelligence techniques in simulation.