Simulation in PASCAL

EARL HUNT

University of Washington, Seattle, Washington 98105

Some of the author's experiences in working with simulation languages are discussed. The use of PASCAL language is detailed.

The history of computer simulation of cognitive processes has been intertwined with the history of development of ever more powerful special-purpose languages for constructing simulation programs. The progression of languages from IPL-IV (for the true historian) to today's dialects of LISP attests to the usefulness of this effort. Even so, these languages still force the psychologist to do a fair bit of "reinventing the wheel" (e.g., rewriting programs for parsing or for production execution). To solve these problems, some more specialized translator systems have developed. These are exemplified by PRISM (Langley, 1983) and CAPS (Thibadeau, 1983). These systems make it easier for psychologists to write simulations of the type envisaged by the system designers. The translators themselves contain assumptions that have psychological content. If one can live with the restrictions, which may have substantial psychological content, programming is much more efficient.

Special-purpose simulation languages are impressive and are highly desirable. Nevertheless, they have two disadvantages. One that has been widely publicized is that the systems are not well suited for arithmetical computations. This is something of a chimera. Simulation languages can (and have) been written to encompass conventional computing, although at the price of requiring a larger machine. However, the cost of computing and memory is dropping rapidly, so computational power is not an issue. I do point out, though, that one wants to write a simulation of a model that is essentially an exercise in mathematical psychology, as opposed to "information processing" in the sense used by Newell, Simon, and most other cognitive scientists, and then the simulation languages may not provide reasonable ways to state your problem. Obviously, one would use BASIC if BASIC would do the job. The interesting question is "What do you do when your problem falls somewhere in between cognitive psychology and mathematical psychology?" It is at this point that a second, less publicized problem appears. I believe that it is crucial. The simula-

tion languages are big-city tools, the intellectual equivalent of Dior gowns worn in the Parises and Londons of cognitive science. Some of us are cognitive hicks. All we have are blue jeans. (The existence of designer jeans has nothing to do with this argument.)

To move away from the analogy, users of special-purpose simulation languages generally have two things: access to a very large machine and membership in a substantial community of artificial intelligence/cognitive science research workers who use, support, and understand the languages. The little old lady from Dubuque cannot play simulation because she does not have a good glove and because she does not have enough friends to make up a team. It has been suggested that equipment is no longer a problem; a substantial computer can be purchased for only \$250,000. Only? True to history, though, computer scientists have promised us that things will get better in the future. They probably will. I am confident that the equipment problem will be solved, although perhaps sooner at the more highly endowed universities. The Dubuquean still needs teammates. In the foreseeable future, a number of bright hardworking people will want to do simulation but will not be located near very many other people who have similar interests. My comments are addressed to the good people of Dubuque.

For the past several years, I and a small (i.e., two persons) group of colleagues have found ourselves in this situation. While the University of Washington has an excellent psychology department, the idea of simulation is somewhat exotic to my colleagues in physiological, social, and, indeed, traditional human experimental psychology. The Computer Science Department is very highly rated in its field, but it is not nor does it desire to be a center of artificial intelligence research. (Psychologists may find this hard to believe, but it is quite possible.) Also, my theoretical bent has always been to try to unite mathematical reasoning with simulation, and thus my own theorizing has inevitably been grounded in mathematical psychology as well as cognitive science.

We have worked quite well in this environment, by programming in general-purpose computer languages. Today we are working in PASCAL. In the past, we have produced simulation and artificial intelligence programs written at various times in FORTRAN and ALGOL

The research reported here was supported by a contract between the Office of Naval Research and the University of Washington. The opinions expressed are the author's and do not reflect policy of the Office of Naval Research. I am happy to acknowledge the helpful comments made by John Cotton, who read an earlier draft of this manuscript.

(Hunt, 1975; Hunt, Marin, & Stone, 1966; Quinlan & Hunt, 1968). Our present efforts will be used to illustrate the problems and potentials of the approach.

We are trying to construct a program that will simulate a variety of tasks in what is known as the "attention and performance" literature. Examples range from choice reaction time experiments to studies of quite complex tasks, in which a person has to split his/her attention over several different subtasks such as tracking, response selection, and memorization. The tasks combine low-level, but not negligible, reasoning requirements with a strong requirement to respond in real-time. Simple tasks in this field yield to mathematical modeling, but complex tasks do not. In order to deal with all of these different tasks within a single simulation, we decided to construct an interpreter that would be capable of executing productions and that would contain within itself a number of mathematical models of various "automated" information processing steps. These models would provide the building blocks for production selection and execution. As a very crude guide, we proposed using production systems to simulate how a person approached a task, consciously, and mathematical modeling to simulate the more mechanistic information processing that supports conscious reasoning.

The interpreter program contains about 1,100 lines of PASCAL code. It has been executed both on a VAX and on a small (65-KB) special microcomputer, called a "PASCAL microengine," that was built specifically for executing PASCAL programs. We have already done a considerable amount of simulation of choice reaction time experiments and have prepared a paper covering this topic. Studies of dual task phenomena and Stroop interference phenomena are now in progress. Clearly, we are in business. The details of our scientific work will be reported elsewhere. Here, I concentrate on what we have learned about simulation itself.

Our program could be thought of as a very limited and rapid CAPS program. In this sense, we invented the wheel at about the same time that it was invented in Pittsburgh. The basic program was written by myself and a technician (who was not a professional programmer) over a period of about 4 months, and then it was modified slightly by a computer science graduate student. From time to time, we were held up by "hack" programming, and we forced PASCAL to do things it was not intended to do. This was frustrating. However, most of my time was spent thinking about the psychological meaning of the programming solutions. That is good. If a packaged system had been used, we would either have had to accept the assumptions of its designers or reprogram someone else's system, in his absence, relying on whatever documentation he could afford to provide. This remark is not intended to be sarcastic. Academic research efforts should not divert much time to product maintenance, and they do not. I

want to be the intellectual colleague of centers in cognitive science; I am not sure that I want to be their customer.

How well is the approach working? The mathematical psychology aspects of our work require a great deal of computation. I am doubtful that some of the searches for good parameters, or estimations of population statistics, would be feasible on all but the most sophisticated LISP systems. Of course, how big a problem this is depends upon the true cost of computing. Our work fits very nicely into normal operations on a VAX system. Depending on the number of other users, simulation of a typical experiment will appear to take place instantaneously or may require 5-10 min of waiting for a response. The verdict is still out concerning the PASCAL microengine. The manufacturer's inadequate support has presented us with a number of problems. Note that here we reverse our usual roles. The centers that use artificial intelligence languages usually rely on large, very well documented systems produced by major manufacturers. If one really wants to cut corners in an effort such as ours, one uses a machine constructed by Mrs. Murphy's Storm Door and Computing Company. Sometimes small companies turn out to be the apple of our eye; sometimes they provide an orchard full of lemons.

Our typical use of the system is to assign a person a simulation project, give him his own file, and expect him to do the necessary reprogramming. Someone working with us can play with assumptions in a way that one probably could not play with the assumptions that are internal to CAPS or PRISM unless he was an experienced LISP programmer. Of course, to play in our league, you have to program in PASCAL. This brings up an extremely important point.

As scientists, we want to develop theories of cognition. As teachers, we want to teach psychology. Any training in the techniques of computer use is strictly subservient to those goals and, frankly, we would prefer to have such training done by other people. The ideal assistant arrives in the laboratory already trained on techniques and ready to begin work immediately on a substantive problem. A person who already knows PASCAL can work in our laboratory after about 1 week of studying the coding conventions. Furthermore, students are likely to know PASCAL. If they do not, computer centers offer courses quarterly. There is also a large "textbook industry" that, by a process of cut-throat competition, has produced excellent introductory and reference manuals for the language. By contrast, LISP programmers are not the rarest of birds in Washington's woods, but they are a bit uncommon. Rather like the pileated woodpecker. As for documentation? LISP documentation will never approach PASCAL's. There is a very strong analogy to getting parts for your Chevrolet or parts for your Mercedes.

It has been suggested that the problem of a common

community is a temporary one. When computer networks are installed, will it be possible to join geographically distributed but electronically centralized neighborhoods? I frankly do not think so. The problem is not communication between professors about ideas, it is a problem of communication between technicians and students about methodology. After some experience with system maintenance by telephone, I do not believe that people will be comfortable with substitutes for face-to-face communication. Computer mail is a highly useful thing, but it has its limitations.

I have emphasized the need to be part of the local computing culture. At the same time, one must be part of the national and international culture of psychologists who do simulations. We want to share ideas with other people. Therefore, it is crucial to ask whether or not we can export our programs for study by other people.

We really do not know. On one hand, for all the reasons I have given, it should be easier to export PASCAL programs than programs in more specialized languages. (Once again, the computer industry is working on the PASCAL export problem, whereas the artificial intelligence industry is working on the LISP export problem. Who's bigger?) However, our system is undoubtedly harder to use, because it lacks many of the convenient input/output features of LISP. For instance, a user must define a variety of parameter matrices by running an auxiliary program that is best described as user hostile. Of course, this is not a feature of PASCAL, it is a feature of the time that we have available to make things convenient. Admittedly, some of the conveniences we have to build for ourselves would be part of the standard package of a special system. This is potentially a serious problem. Standard programming languages are quite adequate to produce simulations for use within our own laboratory. After all, we have been doing this for years. Whether or not we can export programs is not so clear. (Parenthetically, I will be curious to see how well PRISM and CAPS export to people who have never worked in the sending laboratory.) We are currently in the initial stages of two export efforts. I hope that these are successful.

Naturally, the export problem has nothing to do with

the ability of the receiving institution to run our modest PASCAL program. Most community colleges could do this. (The program will not quite fit into an Apple system, but anything a little larger will do.) The problem is in finding how we are doing things and in recreating the psychological reasoning that led us to take a particular programming solution.

Reverse snobbishness is not appropriate. PASCAL programming is not in any sense better than programming in LISP, CAPS, or what have you. The best language for you depends upon your total situation: both the problem you have and the physical and social resources at your command. People who are at centers that develop special-purpose languages will often find that the language being developed is one that they should use. The point is that there is an alternative that deserves consideration, if you are not at such a center or if your problem is one that does not mesh well with the philosophy of the special system. Even were I to be at a major center of artificial intelligence research, I would seriously consider the PASCAL approach if the simulation was intended to contain a substantial mathematical psychology component. There is no universal answer to the question "What programming language should be used?" The decision must be made by considering the total context of the problem and the working situation. Dior gowns and LISP programs are beautiful things to see in the center of the world; blue jeans and PASCAL code fit right into the Puyallup (Washington State) Fair.

REFERENCES

- HUNT, E. *Artificial intelligence*. New York: Academic Press, 1975.
- HUNT, E., MARIN, J., & STONE, P. *Experiments in induction*. New York: Academic Press, 1966.
- LANGLEY, P. Exploring the space of cognitive architectures. *Behavior Research Methods & Instrumentation*, 1983, **15**, 289-299.
- QUINLAN, J. R., & HUNT, E. B. A formal deductive system. *Journal of the Association for Computing Machinery*, 1968, **12**, 459-476.
- THIBADEAU, R. CAPS: A language for modeling highly skilled knowledge-intensive behavior. *Behavior Research Methods & Instrumentation*, 1983, **15**, 300-304.