SESSION IX SIMULATION SYSTEMS FOR COGNITIVE PSYCHOLOGY

Experiences in building a simulation environment for psychology

ALAN LESGOLD

Learning Research and Development Center, University of Pittsburgh, Pittsburgh, Pennsylvania 15260

The hardware, software, and human issues involved in building a cognitive simulation facility are discussed.

The other presentations in this symposium deal with specific research tools and approaches. In contrast, I was asked to address the question of how to build a cognitive simulation capability in a research institution. That is almost an outdated question. As computer power becomes increasingly cheap, it will come to researchers just as other resources do, through seed money, research grants, and related mechanisms. However, perhaps a few of the lessons we have learned at the University of Pittsburgh still apply even as simulation resources are on the brink of being mundane, simple, and affordable.

The primary concerns in building a cognitive simulation facility are hardware, software, and human issues. Most of us have written programs and trained students and/or staff, so we tend to see hardware as the missing component of a simulation capability. However, this is a very shortsighted view. In any well developed use of computers, hardware is the smallest cost factor or constraint, even if it is artificially hard to acquire. Good systems development must proceed from the question "What do I want to do?" This suggests that software choices will dictate hardware needs and that training and other human issues will also be of great importance. Below, I address each of these three areas in turn.

HARDWARE

Selection of appropriate hardware requires some

Preparation of this paper was supported by a contract from the Personnel and Training Programs of the Office of Naval Research and by the Learning Research and Development Center through grant funds from the National Institute of Education. No endorsement by either organization of the views expressed should be inferred. VAX and VMS are trademarks of Digital Equipment Corporation; UNIX is a trademark of Western Electric Corporation; EUNICE is a trademark of the Wollongong Group.

understanding of the nature of cognitive simulation systems. Increasingly, such systems are becoming difficult tests for any general-purpose hardware. Simulation programs are often quite substantial in size, involving numerous rules operating on a very large declarative knowledge store. This means that substantial memory and processing power will be required. For all practical purposes, it can be assumed that the declarative knowledge base in a model will be accessed in a uniform random manner. Further, in a production system model, all of the productions are potentially active all of the time. This high degree of parallelism means that any kind of overlay or chaining approach will probably not be effective. For this reason, large (virtual) memory is a basic requirement for hardware that will be used in cognitive simulation work. Simulations are often very computation-intensive as well, so significant computer power is needed. Further, while the bulk of a simulation program involves symbolic processing, it is increasingly the case that substantial floating-point arithmetic is involved. This is especially true in parallel activation models and in learning models in which many alternatives of a production exist, each with separate strength. Artificial intelligence work, especially in the area of vision, will come increasingly to need floating-point accelerators and even array processors; psychologists will be following close behind in making use of the tools that artificial intelligence researchers develop.

There are four possibilities that are, or soon will be, presenting themselves as possible hardware configurations: the superminicomputer, the mainframe, the dedicated LISP machine, and the supermicrocomputer.

Superminicomputers

The appearance of the VAX-11 computer architecture, along with the many competitors now available, significantly altered ideas about university computing resources. Decentralized systems of powerful mini-

computers connected by high-bandwidth networks are increasingly common. For cognitive simulation work, the standard of comparison remains the VAX-11. This is not primarily because of the power of the hardware. Indeed, other machines are more cost-effective in terms of computational power per dollar invested. However, the bulk of extant simulation systems are programmed in either INTERLISP or a MACLISP variant. Further, because of government-enforced standardization efforts, most of the artificial intelligence community has, until recently, used VAXs running the UNIX operating system. As a result, there is a greater community of experienced workers, better documentation, and more effort aimed at further improvements for the LISP resources on VAXs than on competing systems. Specifically, there is an INTERLISP for the VAX as well as a MACLISP variant, called FRANZ LISP.

INTERLISP-VAX was developed at the Information Sciences Institute (ISI) of the University of Southern California, under ARPA sponsorship. It is now being supported by Digital Equipment Corporation (DEC), which will soon be distributing it. Until DEC takes over, ISI will be handling distribution. FRANZ LISP is available from the Department of Electrical Engineering and Computer Science of the University of California, Berkeley. It is part of *bsd4.1*, a Berkeley enhancement of the UNIX operating system, which originated in Bell Laboratories. Portions of the Berkeley UNIX development effort are now in the hands of private industry, just as is the case with INTERLISP. Consequently, even though nonprofit institutions will probably continue to enjoy price preferences, it can be assumed that LISP software will have increasingly significant costs tied to it in the future.

There are two operating systems common on the VAX, VMS (a product of DEC), and UNIX (available from Western Electric Corporation and, in enhanced form, from the University of California, Berkeley). UNIX is the favorite of the artificial intelligence community, but VMS has better real-time facilities, which may be important to experimentalists. Both FRANZ LISP and INTERLISP run under the UNIX operating system. A VMS version of INTERLISP is available, and one can (as I do) run FRANZ LISP under VMS by using an emulator called EUNICE, which is a product of the Wollongong Group.

In summary, the VAX-11 systems are the standard superminicomputers of the artificial intelligence world and, consequently, the most feasible systems for people entering the world of cognitive simulation modeling. There are thousands of machines in use, which means that many peripheral hardware products, such as disk systems and multiplexers, are being developed by other companies, insuring rapid enhancement and competitive pricing. However, as is noted below, there are choices other than superminicomputers.

Timesharing Mainframe Systems

Both INTERLISP and MACLISP were originally developed for the PDP-10 environment. Many computers with this architecture (DECsystem-10s and DECsystem-20s) are in current use and thus are good sources of computer power for fledgling cognitive simulation workers. Indeed, if one can have significant access to such a machine, it is the obvious choice for the next year or 2, since the hardware world is changing rapidly. However, because of the cost of such systems, they are usually shared with many other users. A small fraction of such a system may be enough for the most preliminary explorations, but it will not suffice for extensive simulation activity.

Dedicated LISP Engines

An important alternative to the superminicomputer or mainframe is the dedicated LISP engine. This type of system consists of several components: (1) a powerful computer with (2) microcode instructions that are optimized to support LISP, along with (3) a substantially enhanced LISP language and programming environment. Currently, there are at least three companies in this market: Xerox Electro-Optical Systems, Symbolics, and the Lisp Machine company. New products keep appearing, and prices keep dropping. Prices range between about \$25,000 and \$125,000 at this writing. Dedicated LISP engines are worth investigating for several reasons. First, they have versions of LISP that make full use of rich display systems, including multiple-screen windows, a variety of graphics forms, and high-level supporting software. Second, they have screen pointer systems (such as the mouse) and supporting software, so that menu selection schemes and related techniques for improving the human-machine interface can easily be developed. Third, they have a variety of supports for developers of very-large-scale software systems, making it easier for programmers to understand the implications of changes they are considering. Finally, in one way or another, most LISP engines support new approaches to system design, such as object-centered programming and other techniques that involve functions that are more or less concurrent, at least at a conceptual level.

Supermicrocomputers

A final possibility that will become increasingly feasible in the next year or 2 is the supermicrocomputer based on microprocessors with 16 or more bits of address space (e.g., the 68000, Z8000, and others). These systems will be of the same power range as many of the LISP engines discussed above. However, being developed for a wider market, they will (1) be useful for more than LISP programming, (2) be somewhat slower to offer solid versions of LISP, and (3) not generally have microcode optimized for LISP. One very hopeful sign is that the people who created Berkeley UNIX

have almost completed a version that will run on 68000-based systems. As of this writing, preliminary copies of the system have been released to several manufacturers of 68000-based systems. Thus, it can be assumed that systems running FRANZ LISP will be available, perhaps by the time this article is printed. In the future, it is quite likely that supermicrocomputer systems that offer (at least) FRANZ LISP in a UNIX environment will be strong competitors for present and future VAX-11 systems. Systems in the \$10,000 to \$20,000 price range should be appearing soon.

SIMULATION LANGUAGES

As discussed earlier, there are two main dialects of LISP that have been used in psychological simulation work. INTERLISP, which was developed initially at Bolt Beranek and Newman, Inc., and then enhanced at Xerox Palo Alto Research Center, is the basis for quite a bit of work in the expert systems area and has also been used for work on intelligent tutoring and cognitive simulation. FRANZ LISP, the Berkeley product based on M.I.T.'s MACLISP, has been the basis for a number of cognitive simulation languages, developed primarily at Carnegie-Mellon University. At the present moment, FRANZ LISP offers more specific tools for psychology, whereas INTERLISP offers a better programming environment. However, advantages of each system are making their way into the other with increasing speed. It seems appropriate to briefly discuss a few of the tools that psychologists have used successfully, simply to provide a sense of what is available.

AGE

This system was developed at Stanford University by Penny Nii, Edward Feigenbaum, and others. It is designed to permit people with minimal artificial intelligence skills to develop expert systems models of two general kinds. Blackboard models involve collections of knowledge structures that communicate with each other by placing their products in a public memory called a blackboard. Classes of knowledge structures (essentially, bundles of productions) can be enabled or disabled by the actions of their "colleagues." The HEARSAY model of speech understanding, developed by Reddy and others at Carnegie-Mellon University, is a blackboard model. Backward-chaining models, the other form AGE generates, work backward from a goal through a chain of actions that realize the conditions for that goal or for subgoals that can help toward achieving the overall goal. The MYCIN expert system for infectious disease diagnosis that Shortliffe and others at Stanford University have developed is an example of a backward-chaining system.

OPS

There are a number of generations of this production system language in existence. The more recent versions

reflect a number of optimizing techniques developed at Carnegie-Mellon University by C. L. Forgy, who is the current source of the language. OPS is a general production system language that has been used by a number of researchers for development of psychological models. In addition to being used directly, OPS forms a core for a number of languages that reflect more specific psychological hypotheses, such as those described next below.

ACT

This language was used by John Anderson for much of his work on learning. It reflects the theoretical position taken by Anderson in his book (Anderson, 1976) and recent *Psychological Review* article (Anderson, 1982). The language supports learning mechanisms, limited short-term memory, and a strength-based execution discipline for productions. It was not really developed for general distribution, however, and is not likely to be readily usable by people who have not had direct exposure to it. Anderson's current interests have led to a new language, GRAPES, which is designed to directly reflect goals and subgoals separately from a more general short-term (activated) memory. As of this writing, GRAPES is not yet ready for general distribution.

CAPS

This is the language developed by Robert Thibadeau (Robotics Institute, Carnegie-Mellon University) for initial use in simulating the reading process in studies with Marcel Just and Patricia Carpenter. What makes the language interesting is its absolutely parallel nature. All productions that are eligible to fire on a given cycle are in fact executed. This is in striking contrast to systems, such as GRAPES, in which only one production is executed on a cycle. Another unique property of CAPS is its form of spreading activation. Other systems spread activation in semantic memory by putting a certain amount of activation at a node and then letting it spread, according to path strength, to all nodes connected to that node. CAPS allows one to specify the spread of activation from one node to a named other node only.

PRISM

This is perhaps the most general simulation language in use that provides the tools psychologists are likely to want. PRISM was written at Carnegie-Mellon University by Patrick Langley and Robert Neches. As it is described in another presentation of this symposium, I will not elaborate on it here, except to note that it has been used at length in our laboratories, with general success and moderate ease.

Summary

We have used CAPS, PRISM, and GRAPES in graduate simulation seminar settings at the University of

Pittsburgh. All have worked reasonably well, with minor bugs (usually in the nature of misfits to our specific system configuration, but requiring LISP expertise to repair). I have used ACT and AGE briefly, with reasonable success. Overall, though, psychological simulation languages are not yet developed to the stage that a relative newcomer to the simulation world could use them with ease. This is largely because there has been no support so far for the extra testing and documentation that are needed to turn a personal research tool into a general resource for the field.

I should also note that a variety of tools of somewhat less generality have been developed and are in a similar state of semiportability. These include parsers, planning languages, and truth maintenance systems. Generally, the best way to locate such resources is to follow leads from the references cited at the end of this article.

THE HUMAN ELEMENT

Even if the right combination of hardware and software is delivered some morning by the tooth fairy, getting into the simulation business is not all that easy. As I have suggested already, the available software is not like a statistics package; it is not well documented, not exhaustively tested, and not perfectly portable (although things are improving on these fronts). This means that the success of a simulation resource still depends heavily on people. The following principles seem worth noting.

Someone Will Have to Work Hard at Shaping the Cognitive Simulation Environment

Ideally, the scholarly leader of a research group should be the best versed in the simulation tools being used. At least, he/she should understand and personally use them. Delegating simulation work is risky. You would not delegate the design of an experiment if you could not do it yourself. The same should be true for simulations, since hidden assumptions can easily creep into such complex programs.

Training Pays Off

I have found it helpful to teach both formal and informal courses in LISP and in the use of cognitive simulation languages. The more people using the tools, the faster they are debugged, and the easier it is to get help with a project. Literature seminars, in which papers reporting simulation work are studied, are also important to the success of efforts to build any cognitive science facility.

Documentation and Archiving Pay Off

Every aspect of computer simulation work of this kind is massive in scope. Consequently, it is essential that files be archived against the possibility of catas-

trophe. Also, good records of changes made as systems are debugged will make it much easier to keep local systems compatible with those at other institutions. Often, new versions of languages are distributed from time to time. Good records of how the previous version was modified to make it work in your facility will pay off when a new version arrives. Finally, it should be noted that documentation stands between many potential simulation users and serious projects. No one can afford to do fully adequate documentation of such systems, but every little bit helps. Efforts are afoot to secure grant support for central cognitive simulation resource centers to do this work, but times are hard, and we may have to each do a bit of the task.

We Need to Produce Students Who Can Work in This Field

There are both selfish and altruistic reasons for this. Many projects are only possible when faculty and students invest levels of effort that could not possibly be afforded if purchased at current salary levels. More important, though, every field depends on an influx of critically intelligent newcomers. Ours is no exception. This underscores the importance of developing facilities at which computational power is in sufficient supply to enable trial and error, the initial stumbling of newcomers. Cognitive simulations are not usually constrained well by the data currently available. This makes it especially important that we encourage both the empirical testing of public predictions and the inspection of models to determine the origins of various predicted effects. Without adequate computer resources, this will not happen sufficiently. Another speaker in this symposium suggested that I represent the elite fashion center of this field. His solution to the problems just mentioned was to write only simple, relatively short programs. If I believed that humans were simple entities whose capabilities corresponded to simple programs, I would do this. Since I do not, I must call for better resources, both for training and research.

THINGS TO READ

The *Handbook of Artificial Intelligence* (Barr & Feigenbaum, 1981, 1982; Cohen & Feigenbaum, 1982) is the ideal place to start learning about basic principles of artificial intelligence. Other basic reference sources include journals and related periodicals, such as *Artificial Intelligence*, *Cognitive Science*, and proceedings volumes from the annual meetings of the American Association for Artificial Intelligence and the biennial International Joint Conferences on Artificial Intelligence (IJCAI). Regrettably, that may not be enough. Most people active in this field have spent time in extensive workshops or sabbaticals at the small number of laboratories in which substantial simulation work is conducted.

REFERENCES

- ANDERSON, J. R. *Language, memory, and thought*. Hillsdale, N.J: Erlbaum, 1976.
- ANDERSON, J. R. Acquisition of cognitive skill. *Psychological Review*, 1982, **89**, 369-406.
- BARR, A., & FEIGENBAUM, E. A. (Eds.). *The handbook of artificial intelligence* (Vol. 1). Los Altos, Calif: Kaufmann, 1981.
- BARR, A., & FEIGENBAUM, E. A. (Eds.). *The handbook of artificial intelligence* (Vol. 2). Los Altos, Calif: Kaufmann, 1982.
- COHEN, P. R., & FEIGENBAUM, E. A. (Eds.). *The handbook of artificial intelligence* (Vol. 3). Los Altos, Calif: Kaufmann, 1982.