Carnegie-Mellon University

Department of Psychology
Schenley Park
Pittsburgh, Pennsylvania 15213
[412] 578-2781

May 4, 1984

Cuthbert C. Hurd
332 Westridge Drive
Portola Valley, CA 94025

Dear Cuthbert:

Spurred by my reading of Alan Perlis' fine chapter, Al and I have finished ours, which is enclosed. I hope that you will find it appropriate for the issue you are editing.

Please let us know if there is anything more you need from us.

With warm regards.

Cordially yours,

Herb

Herbert A. Simon

HAS:tmg

Information Processing Language V on the IBM 650

Herbert A. Simon and Allen Newell

Carnegie-Mellon University

Information Processing Language V on the IBM 650

Herbert A. Simon and Allen Newell
Carnegie-Mellon University

This brief article will complement Alan Perlis's paper on the IBM 650 at Carnegie Institute of Technology, by recounting the story of the construction and use of the list processing language, IPL-V, on that machine. In late 1954, the authors began a collaboration with J. C. (Cliff) Shaw in research on complex information processing with computers. The initial goal was to develop a computer program that could learn to play chess, but in the Autumn of 1955 this target was displaced by the interim goal of writing a program that could prove theorems in the propositional calculus of *Principia Mathematica*. Newell and Shaw, both on the staff of the RAND Corporation in Santa Monica, had access to the newly completed Johnniac computer there, while Simon, on the faculty of Carnegie Institute of Technology, was a consultant to RAND.

In order to achieve their goals, the researchers decided that it was essential to devise a programming language capable of handling large, complex, symbol structures whose size and configuration would be extremely difficult to predict in advance, and would have to be modified during run time. Our solution was to organize memory in terms of list structures and to construct a language designed to operate on lists. By the end of 1955, the first list processing language, IPL-II, had taken shape; early in the following year it was operating on Johnniac and a theorem proving system had been coded in it.

Meanwhile, Newell had moved to Pittsburgh in order to take a doctoral degree there, having interrupted his graduate education when he began work at RAND, some years earlier. In Pittsburgh, he retained his affiliation with RAND, and the research continued vigorously, but with Shaw and Newell on opposite ends of a teletype wire that connected them across the continent -- a sort of prehistoric network with human IMPs.

On the campus of Carnegie Institute of Technology, a few faculty members had been exposed, in the early 1950's, to the new electronic computers. As we have seen, one of these was Simon, another was Charles C. Holt, an economist and engineer. Both were associated with the Graduate School of Industrial Administration, a "New Look" business school established in 1949 with a strong emphasis on the developing tools of operations research and management science. Holt, Newell, and Simon thought the time ripe to bring a computer to the C.I.T. campus, and were successful in persuading the Dean of GSIA, G.L. Bach, and the Dean of Science and Engineering, Richard Teare, to underwrite the cost. Since the electrical engineers on campus were mainly concerned with avoiding responsibility for maintaining such a machine if it arrived, and most mathematicians and scientists could not see how or why they would use one, there was no objection to locating the computer in the basement of GSIA.

Choosing an appropriate computer called for consultation with other universities that already had one, and consultation led to Alan Perlis at Purdue, with the result that an IBM 650 and Alan arrived in Pittsburgh in the Spring and Summer, respectively, of 1956. Alan has told his own story of the 650 at Carnegie in another article in this issue. Here, we need only record our deep gratitude for his imaginative and productive leadership of computing at the university during the decade and a half he was our colleague.

By the Spring of 1956, a number of graduate students, enrolled in Simon's course on Mathematical Methods in Social Science, were considering doing theses on complex information processing (alias Artificial Intelligence). These included Edward A. Feigenbaum, Julian Feldman, Robert K. Lindsay, and Fred M. Tonge, and soon afterwards, Geoffrey Clarkson. Although we could provide them with some access to the RAND machines (Tonge wrote his thesis program in IPL-IV on Johnniac), it became imperative that we bring up a list processing language for student and faculty use at Carnegie. Today, a programmer might have second thoughts about putting a list processing language on a machine with only 2,000 words of high speed memory. When we remember that the high speed store of Johnniac was only 4,096 words (supplemented by a drum with about 10,000 words of usable capacity), the memory limits of the 650, while severe, seemed manageable.

IPL-V, the language developed for the IBM 650, "started out in late 1957 to be a "modified copy" of IPL-IV (then being implemented on Johnniac)... (Newell, 1963, p.86)." An initial running system was produced under Newell's direction in early 1958, mainly by Carleton B. Hensley and Fred M. Tonge (Hensley, Newell, and Tonge, 1958). Meanwhile, since the RAND Corporation had acquired an IBM 704, it was decided that the language should be designed to run on both the 650 and the 704. The revised language, with Newell, Tonge, Feigenbaum, Bert Green, Jr., and George Mealy as its principal designers, was described, in June 1958, in a preliminary version of the IPL-V Manual, and the system became operational on the 704 at the end of the Summer of 1959 (Newell, 1963).

Thus, IPL-V having been coded for both 650 and 704, and provided with a manual (1st edition, 1961; 2nd edition, 1964), became the first list processing language to be made available for public use. Subsequently, IPL-V was brought up on a substantial number of other computers and continued for a decade to be an important language, both for research in artificial intelligence and cognitive science and for teaching the basic concepts of list processing.

A glance at the pioneering research that is collected in Feigenbaum and Feldman's *Computers and Thought* (1963) shows that the IPL-V system on the 650 at Carnegie made important contributions to the foundations of AI and cognitive science. Among the programs written at that time were Feigenbaum's EPAM, a simulation of verbal learning processes, Julian Feldman's program for simulating human decisions in the binary choice experiment, Kenneth Laughery's program for concept formation, Robert Lindsay's SAD SAM, an early program with natural language and reasoning capabilities, and Geoffrey Clarkson's simulation of the investment decisions of a bank trust officer. As can be seen from this list, most of this research focused on the simulation of human cognitive processes.

But while these programs were written in IPL-V, and at least partially debugged on the 650 at Carnegie, most experience in running them on actual tasks was gained on other machines. Both the small memory of the 650 and its brief tenure at Carnegie after IPL-V became available prevented extensive runs of large programs with it. So the greatest significance of the machine in the history of AI was as the instigator and test bed of the first public list processing language, and as an instrument for teaching list processing to the first generation of students in cognitive science.

The availability of IPL-V on the 650, and of a carefully written manual describing the language (Newell et al., 1961, 1964), contributed much also to the diffusion of knowledge of list processing techniques to other university campuses and AI research groups. Because it could be run on a wide variety of computers of the 60s, someone quipped that IPL-V was a machine-independent language, and that the machine it was independent of was the 650.

An interesting, but probably undecidable, historical question is whether IPL-V made a significant contribution to the set of concepts that were some years later labeled "structured programming." At the time when IPL-V was produced, mainstream systems programmers and researchers on algebraic programming languages paid little attention to list processing languages, which were generally regarded as esoteric, and unbearably wasteful of machine time and memory. It was probably Chapter II of Knuth's memorable *Fundamental Algorithms* (1969) that first gave them a measure of credibility outside the AI community. Therefore, although a strong case can be made that the central principles of structured programming were developed and employed extensively by users of list processing languages, almost from the day those languages became available, it is likely that this was largely unknown to the developers of algebraic languages who independently reinvented these principles.

The IPL-V manual is quite explicit in its advocacy of top-down programming and independent closed subroutines. A few brief quotes will indicate the explicitness of its conceptions in this domain.

> One programming strategy, often called the "top-down" approach, is to divide each large process, no matter how complicated, into a small number of subprocesses. Each of these subprocesses is given a name, and its function -- the processing it accomplishes -- is defined precisely by specifying exactly what inputs it requires and what outputs it produces. How the subprocess will carry on this processing does not matter at this stage . . . (1964, pp. 104-105).

> Once any process is coded, attention can be directed to developing and coding each of its subprocesses, using exactly the same strategy of decomposing these into subprocesses. Ultimately, subprocesses are reached that can be defined directly in terms of the IPL primitive processes, so that the decomposition comes to a stop. Although apparently at each stage all the complexities are being relegated to the subprocesses and only codes for trivial processes are being written, it will be found at last that nothing complicated remains to be done at the bottom of the hierarchy . . . (1964, p. 105).
>
> Another principle may be called the principle of isolation. The flexibility in hierarchical organization depends on each subroutine being isolated from the rest of the program, except for a small number of well-defined connections. . . Concretely, one subroutine should not link to another . . . (1964, p. 109).

The top-down approach (with some needed qualifications, not quoted here), the characterization of processes solely in terms of inputs and outputs, hierarchical structure, and wariness of GOTOs are all here, quite explicitly. Nearly a quarter century later, the principles of programming enunciated in the IPL-V manual sound as modern and relevant as when they were written.

It can be seen that the 650 at Carnegie Institute of Technology played a significant role in the early research on artificial intelligence and cognitive science, not so much because it provided computing cycles -- although it did that too -- as because it it provided the occasion for developing the first widely used list processing language, and a facility for training many early computer scientists in the concepts and skills required for using computers to do complex information processing.

References

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

Feigenbaum, E.A. The simulation of verbal learning behavior. In E.A. Feigenbaum & J. Feldman (Eds.), *Computers and thought*. New York: McGraw-Hill, 1963.

Feigenbaum, E.A., & Feldman, J. (Eds.). *Computers and thought*. New York: McGraw-Hill, 1963.

Feldman, J. Simulation of behavior in the binary choice experiment. In E.A. Feigenbaum & J. Feldman (Eds.), *Computers and thought*. New York: McGraw-Hill, 1963.

Hensley, C.G., Newell, A., & Tonge, F.M. *650 IPL, Information processing language*. Unpublished C.I.P. working paper, Carnegie-Mellon University, 1958.

Kelly, H.S., & Newell, A. (Eds.). *Information processing language V manual (2d ed.).* Englewood Cliffs, NJ: Prentice-Hall, 1964.

Knuth, D.E. *The art of computer programming (Vol. 1).* Reading, MA: Addison-Wesley, 1968.

Laughery, K.R., & Gregg, L.W. Simulation of human problem-solving behavior. *Psychometrika*, 1962, *27*, 265-82.

Lindsay, R.K. Inferential memory as the basis of machines which understand natural language. In E.A. Feigenbaum & J. Feldman (Eds.), *Computers and thought.* New York: McGraw-Hill, 1963.

Newell, A. (Ed.). *Information processing language V manual.* Englewood Cliffs, NJ: Prentice-Hall, 1961.

Newell, A. Documentation of IPL-V. *Communications of the ACM*, 1963, *6*, 86-9.

Newell, A., & Tonge, F. An introduction to information processing language V. *Communications of the ACM*, 1960, *3*, 205-11.