



A handwritten signature of Allen Newell in black ink, written in a cursive style.

Allen Newell

March 19, 1927 — July 19, 1992

By Herbert A. Simon

WITH THE DEATH FROM cancer on July 19, 1992, of Allen Newell the field of artificial intelligence lost one of its premier scientists, who was at the forefront of the field from its first stirrings to the time of his death and whose research momentum had not shown the slightest diminution up to the premature end of his career. The history of his scientific work is partly my history also, during forty years of friendship and nearly twenty of collaboration, as well as the history of the late J. C. (Cliff) Shaw, a longtime colleague; but I will strive to make this account Allen-centric and not intrude myself too far into it. I hope I will be pardoned if I occasionally fail.¹

If you asked Allen Newell what he was, he would say, "I am a scientist." He played that role almost every waking hour of every day of his adult life. How would he have answered the question, "What kind of scientist?" We humans have long been obsessed with four great questions: the nature of matter, the origins of the universe, the nature of life, the workings of mind. Allen Newell chose for his life's work answering the fourth of these questions. He was a person who not only dreamt but gave body to his dream, brought it to life. He had a vision of what human thinking is. He spent his life enlarging that vision, shaping it, materializing it in a sequence of computer programs that exhibited the very intelligence they explained.

THE CAREER

In a remarkable talk about his research strategies and history given at Carnegie Mellon University in December 1991, seven months before his death,² Allen described his career as aimed single-mindedly at understanding the human mind, but he also confessed to four or five substantial diversions from that goal--almost all of which produced major scientific products of

their own. These "diversions" included his work with Gordon Bell on computer hardware architectures, the work with Stu Card and Tom Moran on the psychology of human-computer interaction, a major advisory role in the ARPA program of research on speech recognition, and his leadership in establishing computer science at Carnegie Mellon University and in creating the pioneering computer networking of that university's campus.

For the rest, Allen's work aimed steadily, from the autumn of 1955 onward, at using computer simulation as the key research tool for understanding and modeling the human mind. After the first burst of activity, which produced the Logic Theorist, the General Problem Solver, and the NSS chess program, he focused increasingly on identifying and overcoming the limitations and inflexibilities of these models that impeded their extension into a wholly general theory of the mind. His final book, *Unified Theories of Cognition* (1990), records the vast progress that he and others made over thirty years toward such generality, progress that in the final decade of his life focused on the emerging Soar system that he and his colleagues built.

HOW IT BEGAN

Allen Newell was born in San Francisco on March 19, 1927, the son of Dr. Robert R. Newell, a distinguished professor of radiology at Stanford Medical School, and Jeanette Le Valley Newell. An older sister was his only sibling. His father provided an important model for his son. In an interview (McCorduck, 1979, p. 122), Allen once said of him: "He was in many respects a complete man. . . . He'd built a log cabin up in the mountains. . . . He could fish, pan for gold, the whole bit. At the same time, he was the complete intellectual. . . . Within the environment where I was raised, he was a great man. He was extremely idealistic. He used to write poetry."

Allen's childhood was uneventful enough, many of the summers being spent in the Sierra Nevada at the log cabin his father built. Allen acquired a love of the mountains that never left him (an early ambition was to become a forest ranger) and a love of sports that, combined with his 6'1" height and sturdy build, led to the high school football team. He said of his own high school career (Newell, 1986, p. 347): "Allen was an indifferent pupil, though some people seemed to think he was bright. He went to Lowell High School--the intellectual high school of San Francisco--where he turned on academically. He also fell in love at age 16 with a fellow student, Noël McKenna, and married her as soon as tactically possible (age 20)." The marriage demonstrated that Allen and Noël were excellent decision makers even at that early age, for they formed a close and mutually supporting pair throughout the forty-five years of their marriage.

Allen graduated from high school just as World War II was ending, worked for the summer in a shipyard, and then enlisted in the U.S. Navy. Although close to his father and acquainted with many other scientists who were family friends, he had no intention, up to that time, of following a scientific career. Adoption of science as his vocation came, he said, rather suddenly, when, serving on a U.S. Navy ship that carried scientific observers to the Bikini nuclear tests and assigned the task of making maps of the radiation distribution over the atolls, he was infected with the excitement of the scientific enterprise.

On completing his service in the Navy, Allen enrolled in Stanford University, where he majored in physics. Undergraduate research led to his first paper, on X-ray optics (Newell and Baez, 1949). Stanford also exposed him in the classroom to George Polya, who was not only a distinguished mathematician but also a thoughtful student of mathematical discovery. Polya's widely read book, *How to Solve It*, published in 1945, had introduced many people (including me) to heuristic, the art of discovery. Allen came away from that experience aware that the processes of discovery could be investigated and

analyzed and that heuristic--the art of guided search--played a key role in creative thinking. (Our common fascination with heuristic helps account for the rapidity with which Allen and I established common ground on first meeting early in 1952.)

RAND

A year in mathematics (1949-50) as a graduate student at Princeton and exposure to game theory, invented shortly before by von Neumann and Morgenstern, convinced Allen that he preferred a combination of experimental and theoretical research to pure mathematics. Taking a leave from Princeton, he found a position at the RAND Corporation, the then-new think tank in Santa Monica, in a group that was studying logistics problems of the Air Force. Two technical reports he coauthored with Joseph B. Kruskal (*A Model for Organization Theory* [1950] and *Formulating Precise Concepts in Organization Theory* [1951]) demonstrate his interest at that time in applying formal methods to complex empirical phenomena. Both papers adopt a style of axiomatization that was fashionable then in game theory and economics.

A six-week field visit to the Munitions Board in Washington impressed Allen with the distance that separated the formal models from reality, and his trip report, *Observations on the Science of Supply* (1951), exhibits his sensitivity to and sophistication about the organizational realities that he observed (probably reinforcing his brief naval experience and summer's work in the wartime shipyard). Somewhat disillusioned with axiomatization as the route to reality, Allen then turned to the design and conduct of laboratory experiments on decision making in small groups, a topic of considerable active interest in RAND at that time.

Dissatisfied also with small-group experiments as a way of studying organizations, the RAND team of John Kennedy, Bob Chapman, Bill Biel, and Allen conceived of constructing and operating a full-scale simulation of an Air Force Early Warning Station in order to study the organizational processes of the station crews. This effort, funded by the Air Force in 1952, led to the creation of the Systems Research Laboratory at RAND (eventually spun off as the Systems Development Corporation) (Chapman et al., 1959). Central to the research was recording and analyzing the crew's interactions with their radar screens, with interception aircraft, and with each other. These data focused Allen's attention on the information-handling and decision-making processes of the crew members and led to a search for an appropriate technique for analyzing and modeling the process. I met Allen when I became a consultant to the laboratory, and in the first minutes of our initial meeting he and I found common ground in the study of information processes as a route to understanding human decision making in organizations.

One of Allen's special responsibilities in the project was to find a way to simulate a radar display of air traffic, for no technology was available to the lab for making appropriate simulated patterns of blips as they move over radar screens. While searching for computational alternatives, Allen met Cliff (J. C.) Shaw, a RAND systems programmer, then working with the Card-Programmed Calculator, a prehistoric device that just preceded the first stored-program computers. Allen and Cliff conceived the idea of having the CPC calculate the successive air pictures and print out simulated radar maps. This not only provided the required laboratory simulation but also demonstrated to Al and Cliff (and to me when I learned of it) that computers, even prehistoric computers, could do more than arithmetic: they could produce spatial arrangements of nonnumerical symbols representing airplanes.

Now two of the preconditions were in place for Allen's move to the goal of understanding human thinking. He clearly saw information processing as a central activity in organizations, and he had had a first experience in symbolic computing. A third

precondition derived from contact with the stored-program computer Johnniac that John von Neumann was building at RAND in about 1954.

At this time the ideas of cybernetics and artificial life were abroad. W. Ross Ashby had published in 1952 his *Design for a Brain*. W. Grey Walter (1953) in England had constructed some mechanical "turtles" that wandered about the room searching for a wall outlet when their batteries ran low, and similar creatures were built by Merrill Flood's group at RAND. By 1950 both Turing and Shannon had described (but not actually programmed) strategies for computer chess, and in 1952 I described (but did not implement) a program extending Shannon's ideas. On an auto trip en route to observing some Air Force exercises in the summer of 1954, Allen and I discussed at length the possibilities of using a computer to simulate human problem solving, but we were not then diverted from our current research on organizations.

THE COMMITMENT

In September 1954 Allen attended a seminar at RAND in which Oliver Selfridge of Lincoln Laboratories described a running computer program that learned to recognize letters and other patterns. While listening to Selfridge characterizing his rather primitive but operative system, Allen experienced what he always referred to as his "conversion experience." It became instantly clear to him "that intelligent adaptive systems could be built that were far more complex than anything yet done." To the knowledge Allen already had about computers (including their symbolic capabilities), about heuristic, about information processing in organizations, about cybernetics, and proposals for chess programs was now added a concrete demonstration of the feasibility of computer simulation of complex processes. Right then he committed himself to understanding human learning and thinking by simulating it. The student of organizations became a student of the mind.

In the months immediately following Selfridge's visit Allen wrote (1955) "The Chess Machine: An Example of Dealing with a Complex Task by Adaptation," which outlined an imaginative design for a computer program to play chess in humanoid fashion, incorporating notions of goals, aspiration levels for terminating search, satisfying with "good enough" moves, multidimensional evaluation functions, the generation of subgoals to implement goals, and something like best first search. Information about the board was to be expressed symbolically in a language resembling the predicate calculus. The design was never implemented, but ideas were later borrowed from it for use in the NSS chess program in 1958.

Newell's goals already extended far beyond chess: "The aim of this effort, then, is to program a current computer to learn to play good chess. This is the means to understanding more about the kinds of computers, mechanisms, and programs that are necessary to handle ultracomplex problems (Newell, 1955). When the paper was presented in March 1955 at the Western Joint Computer Conference, Walter Pitts, the commentator for the session, said, "But, whereas [the authors of the other papers] are imitating the nervous system, Mr. Newell prefers to imitate the hierarchy of final causes traditionally called the mind. It will come to the same thing in the end, no doubt. . . ." From the very beginning something like the physical symbol system hypothesis was embedded in the research.

THE LOGIC THEORIST AND LIST PROCESSING

Even before his "conversion" Allen had been making plans to move to Pittsburgh early in 1955, with Noël and their new son Paul, to work with me in organizational research and earn a doctoral degree (in industrial management!). RAND agreed to continue to employ Allen as its (one-man) Pittsburgh outpost. This plan was duly executed but with the crucial alteration that

the research was to be on programming a chess machine. It was arranged that Cliff Shaw at RAND would collaborate with us, and the program would run on RAND's Johnniac. For various technical and accidental reasons chess soon changed to geometry and geometry to logic, and the Logic Theory Machine (LTM), which discovered proofs for theorems in the propositional calculus, emerged as a hand simulation by December 15, 1955, and a running program in the summer of 1956.

Work was pursued simultaneously on a programming language that would be adequate for implementing the design, leading to the invention of the Information Processing Languages (IPLs), the first list-processing languages for computers. It is fair to say that the LTM and its successor, the General Problem Solver, laid the foundation for most of the artificial intelligence programs of the following decade. A genuine computer program performing a task of some sophistication has much more persuasive and educational powers than do verbal discussions of ideas. A running program is the moment of truth.

LTM was not a "deduction machine"--in fact, it worked backwards, inductively, from hypothesized theorem to the axioms. Discovering proofs is much like discovering anything else, a process of selective search. The fact that the task involves symbolic logic does not make the problem-solving process any more "logical" or less "intuitive" than if some other task (e.g., looking for a law that would connect the distances of planets from the sun with their periods of revolution) were in question.

Although this work was incorporated in Allen's doctoral dissertation, I never regarded him as my "student." Allen, Cliff, and I were research partners, each contributing his knowledge to a wholly joint product. Allen, when he arrived in Pittsburgh, already had five years of scientific work under his belt and needed colleagues more than teachers. I do not suggest that he did not learn--he never stopped growing and learning throughout his life--but he learned as scientists learn, from everyone and everything around them, especially from observation of nature itself.

Why did this particular work, which was part of an already existing Zeitgeist that had engaged the efforts of many able scientists, become highly visible and influential? An essential element in its impact was the actual running program. In addition, LTM and its successors were not directed at a single task. The specific programs were steps toward the solution of the general problem: understanding the human mind. The strategy is stated clearly in the first publication on LTM: "In this paper we describe a complex information processing system . . . capable of discovering proofs for theorems in symbolic logic. This system, in contrast to the systematic algorithms . . . ordinarily employed in computation, relies heavily on heuristic methods similar to those that have been observed in human problem solving activity. The specification is written in a formal language, of the nature of a pseudo-code . . . for digital computers. . . . The logic theory machine is part of a program of research to understand complex information processing systems by specifying and synthesizing a substantial variety of such systems for empirical study" (Newell and Simon, 1956).³

It is all there: complex information processing, symbolic computation, heuristic methods, human problem solving, a programming language, empirical exploration. These are the components of the fundamental research strategy of the Carnegie-RAND group in 1955 and 1956 that continued to guide Allen Newell's scientific work throughout his career. It led him continually to identify and diagnose the limitations of the programs he built and to ponder about architectures that would remove those limitations, and it led him in the last decade to Soar--not as the final answer, for he knew that there are no final answers in science, but as the next step of progress along a path that he followed as long as he was able to work.

EXPLOITING THE FIRST SUCCESS

For about five years after 1955 the Newell-Shaw-Simon team, aided by a growing circle of graduate students, pushed forward the research ideas opened up by LTM and the IPL programming languages. Among the main thrusts in which Allen was involved were thinking-aloud protocols, the General Problem Solver, information-processing languages and production systems, the NSS chess-playing program, and human problem solving. Until 1961 he remained on the staff of RAND (in Pittsburgh); in that year he accepted appointment as an institute professor at Carnegie Institute of Technology.

THINKING-ALoud PROTOCOLS

There are severe difficulties in testing a theory of human thinking that predicts the sequence of thought processes each of only a few hundred milliseconds duration. Apart from neurological evidence, which is only now beginning to become available for tracing some processes, there were few obvious ways of obtaining data while a task was being performed, even at a density of one data point per second. It occurred to the team to instruct subjects to think aloud while performing problem-solving tasks. However, fifty years earlier the method called "introspection" had been thoroughly discredited as a means of obtaining reliable data in psychology. Hence, it was necessary to show that the thinking-aloud method was quite different from classical introspection and to determine the circumstances under which it could provide objective evidence about thought processes. A program of laboratory experimentation using thinking-aloud methods was launched by the beginning of 1957; formal methods were developed for encoding protocol data (problem behavior graphs); and a decade later Allen and Don Waterman made the first, only partially successful, attempt at automating protocol analysis (Waterman and Newell, 1971).

THE GENERAL PROBLEM SOLVER (GPS)

In the summer of 1957, during a workshop at Carnegie Tech on organizational behavior, AI and I extracted from the protocol of a single subject solving logic problems what proved to be a key mechanism in human problem solving: means-ends analysis. In ME analysis the problem solver compares the current situation with the goal situation; finds a difference between them; finds in memory an operator that experience has taught reduces differences of this kind; and applies the operator to change the situation. Repeating this process the goal may gradually be attained, although there are generally no guarantees that the process will succeed.

The idea of M-E analysis led to the General Problem Solver (Newell, Shaw, and Simon, 1960), a program that could solve problems in a number of domains after being provided with a problem space (domain representation), operators to move through the space, and information about which operators were relevant for reducing which differences. The research also discovered schemes that permitted GPS to produce its own operators from a small set of primitives and to learn which operators were relevant for reducing which differences.

THE INFORMATION PROCESSING LANGUAGES (IPLS)

The IPL languages in artificial intelligence and their contemporary FORTRAN in numerical computing settled once and for all the essentiality of higher-level languages for sophisticated programming. The IPLs were designed to meet the needs for flexibility and generality: flexibility, because it is impossible in these kinds of computations to anticipate before run time what sorts of data structures will be needed and what memory allocations will be required for them; generality, because the goal is not to construct programs that can solve problems in particular domains, but to discover and extract general problem-solving

mechanisms that can operate over a range of domains whenever they are provided with an appropriate definition for each domain.

To achieve this flexibility and generality the IPLs introduced many ideas that have become fundamental for computer science in general, including lists, associations, schemas (frames), dynamic memory allocation, data types, recursion, associative retrieval, functions as arguments, and generators (streams). The IPL-V Manual (Newell, 1961), exploiting the closed subroutine structure of the language, advocated a programming strategy that years later would be reinvented independently as structured programming--mainly top-down programming that avoided go-to's. LISP, developed by John McCarthy in 1958, which embedded these list-processing ideas in the lambda calculus, improved their syntax and incorporated a "garbage collector" to recover unused memory, soon became the standard programming language for artificial intelligence (AI).

PRODUCTION SYSTEM LANGUAGES (OPS5)

Allen did not regard the IPLs or their successors as final solutions to the problems of organizing AI programs. Experience with the General Problem Solver revealed a tendency for the program to burrow into a deep pit of successive subgoals, with no way for the top program levels to regain control. A way out of the dilemma began to appear in the middle 1960s in the form of production system languages, introduced into computing by Bob Floyd and others to aid in compiling compilers. In a production system each instruction in the language takes the form of a condition followed by an action: "IF [such and such is the case] THEN [do so and so]." Completely general programming languages can be constructed on this plan.

The Carnegie-RAND group saw in production system languages a solution to the control problem, and Allen took leadership in the development of a succession of such languages, the best known and most widely used of which is OPS5. OPS5 in turn provided the central ideas for the language employed to program the Soar system. A closely related set of ideas that we developed at about the same time, out of concern with the program control problem, led to a decentralized system in which independent processes add information to a common memory ("blackboard") and obtain information they need from that memory. The blackboard idea has achieved wide use in speech recognition, vision programs, and elsewhere.

CHESS: THE NSS PROGRAM

The third main substantive product of the Carnegie-RAND group was a chess program named NSS, the initials of its authors (Newell, Shaw, and Simon, 1958). It was not the first chess program to be implemented and run (Alex Bernstein, among others, completed programs somewhat earlier), nor was it a very strong player: as critics of artificial intelligence were fond of pointing out, it was once beaten by a ten-year-old child. What the critics failed to understand was its purpose: to demonstrate how highly selective search guided by heuristics and by goals evoked by cues in the problem situation could achieve intelligent behavior in a complex task.

HUMAN PROBLEM SOLVING

With the completion of LTM, GPS, the list-processing languages and production systems, and NSS, Al, Cliff, and I began more and more to pursue separate projects in collaboration with other colleagues and graduate students. The last major project Allen and I undertook together was to summarize our research on problem solving--experiments, simulation, and theory--in *Human Problem Solving*, which was published in 1972. The gradual cessation of close collaboration reflected no

rift, as is evident from our joint 1975 Turing Lecture (Newell and Simon, 1976) and the weekly or more frequent conversations that continued until a few days before Allen's death, but a natural drift as each of us interacted with different graduate students and faculty colleagues and built our research strategies to reflect different bets about the locus of the biggest payoffs from studying intelligence.

COGNITIVE ARCHITECTURE

Allen, from an early stage of his research and increasingly as the years passed, was especially concerned with computational architecture and modeling the control structures underlying intelligence.

An architecture is a fixed set of mechanisms that enable the acquisition and use of content in a memory to guide behavior in pursuit of goals. In effect, this is the hardware-software distinction. . . . This is the essence of the computational theory of mind. (Newell, 1992, p. 27)

The early attention of the RAND-Carnegie group to flexibility and generality and the realization of these properties in the programming languages the group invented have already been noticed. The languages became part of the "hardware" that supported the underlying structure for the AI programs, anticipating the much later efforts of others to embed list processing in actual physical hardware. The languages also built into the AI systems some of the salient characteristics of human memory as revealed by psychological research, for example, its associative structure embodied in lists and schemas and the production-like character of stimulus-response connections.

UNSOLVED ARCHITECTURAL PROBLEMS

But important architectural problems remained unsolved. The experience with GPS underlined the importance of control structures for keeping a problem-solving system on course, neither dissipating its efforts in scattered random search nor following long narrow paths that often led, after much wasted effort, to dead ends. The concern for these problems can be traced through a series of Allen's publications beginning in the early 1960s and continuing through most of his career: "Some Problems of Basic Organization in Problem-Solving Programs" (1962), "Learning, Generality and Problem Solving" (1963), "The Search for Generality" (with G. Ernst, 1965), "Limitations of the Current Stock of Ideas for Problem Solving" (1965), "On the Representation of Problems" (1966), "The Trip Towards Flexibility" (1968), "A Model for Functional Reasoning in Design" (with P. Freeman, 1971), "A Theoretical Exploration of Mechanisms for Coding the Stimulus" (1972), "Production Systems: Model of Control Structures" (1973), and "How Can Merlin Understand?" (description of a "unified" architecture based on matching) (with J. Moore, 1974); then, after about an eight-year interval, "Learning by Chunking: Summary of a Task and a Model" (with P. S. Rosenbloom, 1982) and "A Universal Weak Method" (with J. Laird, 1983)--these last two papers being early descriptions of crucial components of what became the Soar system, which occupied the last decade of Allen's life.

THE MERLIN PROGRAM

MERLIN, an architectural enterprise undertaken about 1967, began as an attempt to build a pedagogical tool but became a serious effort to construct a system that had understanding. "MERLIN," Newell wrote, "was originally conceived . . . out of an interest in building an assistance-program for a course in AI. The task was to make it easy to construct and play with simple . . . instances of AI programs. . . . [T]he effort transmuted into . . . building a program that would understand AI--that would be able to explain and run programs, ask and answer questions about them. . . . The intent was to tackle a real domain of knowledge as the area of constructing a system that understood."

The basic ideas around which MERLIN was built were analogy and matching: "the construction of maps from the structure that represents what MERLIN knows to be the structure that MERLIN seeks to understand." The difficulties that were encountered en route to this goal were so severe that Newell regarded MERLIN as a failure, not reaching its practical goals and not producing results that had an impact on the rest of the field. It is described in a single published article (Moore and Newell, 1974). Many innovative AI ideas were embedded in MERLIN, but Allen was reluctant to publish them prior to building a complete running system that incorporated them all.

DIVERSIONS

The important work that Allen described as his "diversions" included research on computer hardware structures, the fostering of research on speech understanding, and research on human-computer interaction. Later I will mention other diversions in the form of institution-building activities.

COMPUTER STRUCTURES

It is perhaps not surprising that someone deeply concerned with program organization would become interested in computer hardware architectures, and Allen did. Nevertheless he regarded his work on this topic, which began with Gordon Bell's invitation in about 1968 to collaborate on a book on computer systems, as a diversion from his main objective. The strategy of simulating human thinking did not rest on any assumption of similarity between computer architectures and the architecture of the brain beyond the very general assumptions that both were physical symbol systems and that therefore the computer could be programmed to behave like the mind. Nevertheless, there are fundamental architectural problems common to all computing that reveal themselves in hardware and software at every level, for example, how to organize systems so that they can operate in parallel on multiple tasks with due respect for priorities and precedence constraints between processes.

Newell and Bell undertook to describe architectures at two levels: (1) the system level in terms of memories, processors, switches, controls, transducers, data operators, and links (the PMS language) and (2) the instruction level in terms of the detailed operations of the instruction set (the ISP language). Their book, *Computer Structures: Readings and Examples*, using the PMS and ISP languages to characterize a large number of computers, appeared in 1971. A revised edition coauthored with Bell and Siewiorek was published in 1981.

The work with Bell led Allen to other projects on computer and software systems design, and a number of his publications up to about 1982 were devoted to these projects. In 1970-71 Newell, McCracken, Robertson, and others built a language, L*, that aimed at providing systems programmers with a kernel that would facilitate building operating systems and user interface.

In 1972, in connection with an AI workshop that we organized, Newell, Robertson, and McCracken built a pioneering hierarchical menu system that gave the workshop participants access to demonstrations of an assortment of AI programs.

Some years later in 1978-82, using the new touch-screen technology, this idea developed into the hypermedia ZOG system, which became a tool for accessing the administrative data base on the newly launched aircraft carrier *USS Carl Vinson* (Newell et al., 1982). Other computer and systems design diversions for Allen included work with colleagues in the computer science department around 1971 on C.mmp and other parallel hardware cum software systems that were being designed there (Bell et al., 1971).

SPEECH UNDERSTANDING

A further major diversion for Allen in the 1970s resulted from ARPA's interest in the possibility of launching a program in automatic speech recognition. Specifically because he was not an active speech researcher and hence stood in a neutral corner, Allen was asked to chair a study group whose 1971 report formed the basis for a major ARPA research effort (Newell et al., 1971). Allen then became chair of the steering committee for the project and produced a progress report in 1975 (Newell et al.) and a final evaluation in 1977 (Medress et al.). His role in the speech effort illustrates both his stature in the profession and his willingness to accept "citizenship" responsibilities for the growth of artificial intelligence.

HUMAN-COMPUTER INTERACTION

When the Xerox PARC laboratory was formed in 1970, Allen, consulted about its research program, proposed a project that would apply psychological theory to human-computer interaction and, in particular, to the design of computer interfaces. Beginning in 1974, Allen with two of his former students, Stu Card and Tom Moran, began to bring together existing psychological data, examine them for regularities (such as Fitts's Law and the power law for learning), construct an engineering-level model of routine cognitive skills (MODEL HUMAN PROCESSOR) and a methodology, "GOMS," standing for goals, operators, methods, and selection, for analyzing new tasks in terms of the basic processes required to perform them. This work was brought together in *The Psychology of Human-Computer Interaction* (Card et al., 1983).

Though in one sense a diversion from his main concerns, the Xerox PARC activity brought Allen back from a preoccupation with computers to concern for the human mental architecture. Moreover, the requirement of modeling entire human tasks required the group to think in terms of a broad-gauged, unified theory. In this sense the project was a step toward the Soar system, planning for which began in the later 1970s before publication of the human-computer interaction book.

SOAR

Allen came to doubt that lack of experimental evidence was the limiting factor in the progress of cognitive psychology. Sufficient data, he thought, already existed to pin down much of the structure of the mind at the architectural level. Moreover, further experimental work would be well aimed and useful only if guided, not by particularistic microtheories, but by a broad theoretical framework. In his final book, *Unified Theories of Cognition* (1990), based on his William James Lectures at Harvard in 1987, he called for such theories and drew the bold outlines of what such a theory might look like, taking Soar as his model. He was careful not to refer to Soar as "*the* unified theory of cognition," but introduced it as "a candidate unified theory." Indeed, in his final chapter he gives a reasoned argument as to why "there must be many unified theories" on the road to developing a veridical one.

When existing unified theories are viewed closely, each can be seen to be built around a core cognitive activity, which is then extended to handle other cognitive tasks. In Anderson's Act* the core is semantic memory; in EPAM, perception and memory; in connectionist models, concept learning. In Soar as in GPS the core is problem solving, and the central GPS concept of problem space is taken over and expanded to allow the system to use multiple problem spaces in solving a single problem. The Soar program is a production system. To this were added two key components developed in collaboration with graduate students: learning by chunking (Rosenbloom and Newell, 1982), which produced a wide variety of kinds of learning obeying the empirically observed power law, and a universal weak method (Laird and Newell, 1983), which incorporated a method for universal subgoaling.

Learning by chunking derived from previous AI work on memory organization in terms of chunks and on learning by adaptive production systems (systems that created and assimilated new productions). What was new in Soar was the use of this mechanism as the sole learning mechanism and the demonstration that it was both powerful and consistent with the power law of learning.

The universal weak method of problem solving consisted at each step of finding which operators were then executable; if there were none or if there were more than one, declaring an impasse, and moving to a new problem space with a new subgoal to resolve the impasse. This procedure generalized the idea of problem spaces and established a consistent semantics for the possible relations among them.

The Soar project continued to grow through the 1980s and 1990s with steadily increasing numbers of active participants at Carnegie Mellon and elsewhere (including the University of Michigan, the Information Systems Institute at the University of Southern California, and several European sites). The effort was directed at extending and strengthening the basic Soar architecture and simultaneously demonstrating its capacity for handling a widening range of tasks, including language comprehension, complex problem solving, and even cryptarithmic--one of GPS's initial tasks. The scope of the system at the time of Allen's death can be seen from his *Unified Theories* book, and work on it continues actively today on numerous fronts.

While it would be hazardous to predict what resemblance there will be between Soar and the "ultimate" unified theory of cognition, it is already evident that Allen's strategy of putting all of his (and many other people's) energies into Soar has intensified interest in building broad-gauged theories that cover a wide range of cognitive processes and has left an important permanent mark on cognitive science.

SCIENCE STATESMANSHIP

It is hard to know whether to classify the time Allen spent as a citizen of the university and of the wider science community as one of his diversions or as part of the mainstream of his scientific work. From the time of his employment at RAND he was keenly aware of the dependence of progress in science upon the institutions that housed and nourished it and he identified closely with the institutions in which he worked. During the early years of his stay at RAND he was persuaded that the think tank was the preferred research organization of the future, but he gradually came to believe that universities had capacities for self-renewal that were hard to maintain in independent laboratories. This change in belief played an important part in his decision to move in 1961 from RAND to the faculty of the Carnegie Institute of Technology.

Allen played an important leadership role in every organizational setting in which he found himself: RAND, the computer science department (later a school) at Carnegie Mellon, the whole university, the national and international computer science research community, and ARPA as a part of it. In general he did not do this by occupying formal administrative positions but by taking on specific assignments and by serving as a very active and highly valued elder statesman. For these purposes he was, as I have remarked, "elder" all his life.

THE COMPUTER SCIENCE DEPARTMENT

While still a doctoral student Allen was already called on for advice as we first brought computers to Carnegie Mellon University. (The first one arrived with Alan Perlis in about 1956.) By 1961, when an informal graduate program in computer science was set up by mutual agreement among four departments, Allen was a major figure along with Perlis and myself in pushing its development and then creating a computer science department, involved deeply in decisions about curriculum and the acquisition of equipment.

With Bert Green, then chairman of the psychology department, Allen was instrumental in obtaining the first large, continuing NIMH research grant for cognitive science research in that department. He was a principal figure, initially along with Alan Perlis, in obtaining and renewing the large ARPA grants that provided the core funding for what quickly became one of the nation's leading computer science departments. For the ensuing quarter century or more Allen played a major role in both departments through his research, his teaching, his guidance of graduate students, and his participation in policy.

THE CAMPUS NETWORK

From about 1972 the experience of members of the computer science department with the ARPA network convinced the community that a network of electronic communications was essential not only for the department but for the university. With the department having persuaded the university administration that Carnegie Mellon was in a unique position to offer national leadership in this direction, Allen agreed to serve as chairman of a task force that was appointed to prepare a plan and to educate the campus community about its potential. In February 1982 the task force issued its report, *The Future of Computing at Carnegie Mellon University*. An agreement was reached with IBM for collaboration in designing and installing the system, and the Andrew system, CMU's campus-wide network--one of the first in the nation--came into being. (The Andrews were Andrew Carnegie and Andrew Mellon.)

ARPA

From its beginnings artificial intelligence and simulation of human thinking have been foci of controversy, eliciting disbelief and anger from those who find the idea of a machine thinking either incredible or threatening. Decisions about funding AI research inevitably became enmeshed in this controversy about its worth, and the support by ARPA of computer science in general and AI in particular was periodically under attack throughout a long and stormy history.

A very large slice of Allen's life was spent preparing research proposals and budget defenses for computer science at Carnegie Mellon, as well as participating in ARPA planning exercises and interpreting AI and cognitive science research to the broader scientific community. This, too, is a normal part of institution building in science, but not its pleasantest part. Allen, while resenting the time lost in these duties, never shirked them. However, his belief (and mine) was that propaganda of the deed was more important than propaganda of the word: that in the longer run the fate of AI and cognitive simulation would be determined not by debates with philosophers about what was possible, a priori, but by our success or failure in building programs that demonstrably simulate and thereby provide theoretical explanations for human thought processes. Every possible waking moment was to be reserved for that task.

COGNITIVE SCIENCE AND AAI

Professional organizations are important among the institutions of science, and Allen played his role in them also. It was an honor that he was proud of, but no surprise, that he was elected the first president of the American Association for Artificial Intelligence and received the first Award for Research Excellence from the International Joint Conference on Artificial Intelligence. Editorships, however, were not for him.

<a>ALLEN, THE PERSON

Allen Newell was a memorable person in the most literal meaning of that phrase. I will draw here on my own impressions as recorded in my autobiography (Simon, 1991) and follow these with some comments by others who knew him well.

When I first met AI at RAND in 1952, he was 25 years old and fully qualified for tenure at any university--full of imagination and technique. . . . His energy was prodigious, he was completely dedicated to science, and he had an

unerring instinct for important (and difficult) problems. If these remarks suggest that he was not only bright but brash, they are not misleading.

If imagination and technique make a scientist, we must also add dollars. I learned . . . [from AI] . . . how to position the decimal point in a research proposal. . . . Thinking big has characterized AI's whole research career, not thinking big for bigness' sake, but thinking as big as the task invites. . . .

From our earliest collaborations, AI has kept atrocious working hours. By this I . . . mean . . . that he works at the wrong time of day. . . . He preferred sessions that began at eight in the evening and stretched almost to dawn. I would have done most of my day's work by ten that morning, and by ten in the evening was ready to sleep, and not always able not to.

Perhaps his greatest pleasure . . . is an "emergency" that requires him to stay up all night or two consecutive nights. I recall his euphoria on our visit to March Air Force Base in 1954, when the air exercise extended over a whole weekend, twenty-four hours per day.

Some of these memories are frivolous, but high spirits, good humor, and hard work have characterized my relations with AI from the beginning.

Allen was serious but not solemn. Whimsy and laughter came easily and often to him. Life, sometimes perplexing, was not a plodding march but a vivid drama in which he acted with brilliance and *éclat*, quite aware of the dramatic effects he was producing. This too was obvious early on. The Systems Research Laboratory operated on the grandest scale, its cast an entire Air Force unit. Only Allen and his codirectors could have dreamed up theater on this megabuck scale at a time when behavioral scientists might timidly request \$5,000 or \$10,000 for their research. His forceful qualities and his exuberance impressed themselves on all who met him.

As Cliff Shaw recalled (McCorduck, 1979): "Energy is the thing I remember mainly about working with AI. Energy and brilliance. Long phone calls and long sessions on the teletype were typical. We would have sessions late into the night at AI's home. I felt like I was tagging along behind, trying to get that Johnniac to do what we already knew could be done. And with AI's energy, it was a good thing he had IPL-V, the programming language, as another outlet, so all that energy didn't descend on me."

And much later, from a graduate student: "Allen Newell was not my adviser . . . and was not on my thesis committee. But still Allen shaped my thinking: from him I learned every day more and more what research is. Through him I understood how to work dynamically towards my research goals. . . . Very rapidly, the initial intimidation I felt . . . was transformed into admiration, friendship and respect." That testimonial, given at his memorial service, could be duplicated dozens of times over by his co-workers--from full professors to new graduate students. In his work with students he was patient, his criticism was constructive, he never lost his temper. If he had any faults as a mentor (and who does not?), it was probably in becoming so involved himself in his students' research problems that he sometimes provided them with more structure and more insights than was good for them. They had to work very hard and fast to retain the strategic initiative.

Everything he attended to he attended to with energy and depth--whether it was his current research problem or an inquiry directed to him by a student or a visitor. In fact, it was this inability to address matters superficially that made the diversions weigh so heavily on Allen, taking him from his main research for considerable periods of time. But he handled the diversions with the same cheery enthusiasm and *éclat* as he did the mainstream tasks. It is hard to recall a lackluster Newell performance, whether it be a public address, a conversation in his office, or the analysis of a thinking-aloud protocol.

I have said nothing about Allen's family life or leisure. Noël and Allen with their son Paul formed a close-knit family. As much of his work was done on the computer at home, he was not at all an absentee husband or father, but shared his activities with his family in spite of his marathon work week. The Newells enjoyed entertaining their friends, most of them from CMU or other academic communities. The categories of introvert or extrovert don't quite seem to fit Allen; he worked long hours in his study, but he spent enormous amounts of time with other people--usually engaged in professional tasks.

There was little evidence of, or time for, the simpler kinds of leisure or hobbies unrelated to his work. He traveled fairly often abroad, usually with Noël, but mostly to professional meetings, and only occasionally did he add more than a few vacation days to these trips. At the very end of his life I learned--much to my astonishment, as I had had no inkling of it--that he frequently watched Sunday afternoon (or was it Saturday afternoon?) TV football games, perhaps a bit of fond nostalgia for his high school athletic days.

It is fitting to conclude this account with a selection from Allen Newell's own set of maxims for the dedicated scientist, proposed in his "Desires and Diversions" talk of December 1991, for these maxims describe his own life:

To each scientific life, its own style; and each style defines a life.

Science is in the details.

To work with the results of field X, you must be a professional in X.

There is no substitute for working hard--very hard.

A scientist is a transducer from nature to theory; seek out nature and listen to her.

The scientific problem chooses you; you don't choose it.

New things get their start by evolution or change, not design.

Everything must wait until its time; science is the art of the possible.

Diversions occur, make them count; salvage what is possible for the main goal.

Solve whatever problems must be solved; but do not be seduced by them.

Deep scientific ideas are exceedingly simple; others usually see them as trivial.

and, finally:

Choose a final project to outlast you.

For Allen, Soar was that project.

NOTES

¹ In preparing this account of Allen Newell's life I have drawn heavily on a briefer memorial (Simon, 1993) published in *Artificial Intelligence* and on a more complete one published by John Laird and Paul Rosenbloom (1992) in *AI Magazine*.

Newell's papers are deposited in the Archives of Hunt Library at Carnegie Mellon University, where can also be found the transcripts of lengthy interviews with Newell by Pamela McCorduck, which were used extensively in her *Machines Who Think* (1979), and by Arthur L. Norberg, who interviewed Newell about his activities in connection with ARPA.

² This talk was videotaped and is available by writing to University Video Communications, P.O. Box 5129, Stanford CA 94309.

³ Although, for reasons that are no longer obvious, Cliff Shaw was not a coauthor of this paper; he was a full partner in the entire research effort.

REFERENCES

- Ashby, W. R. 1952. *Design for a Brain*. New York: Wiley.
- Bell, C. G., and A. Newell. 1971. *Computer Structures: Readings and Examples*. New York: McGraw-Hill.
- Bell, C. G., W. Broadley, W. Wulf, A. Newell, C. Pierson, R. Reddy, and S. Rege. 1971. *C.mmp: The CMU Multiminiprocessor Computer: Requirements, Overview of the Structure, Performance, Cost and Schedule*. Technical Report, Computer Science Department, Carnegie Mellon University, Pittsburgh.
- Berkeley, E. C. 1949. *Giant Brains, or Machines That Think*. New York: Wiley.
- Bowden, B. V., ed. 1953. *Faster Than Thought*. New York: Pitman. (Contains Turing's description of a chess-playing program.)
- Card, S., T. P. Moran, and A. Newell. 1983. *The Psychology of Human-Computer Interaction*. Hillsdale, N.J.: Erlbaum.
- Chapman, R. L., J. L. Kennedy, A. Newell, and W. C. Biel. 1959. The systems research laboratory's air defense experiments. *Manage. Sci.* 5:250-69.
- Freeman, P., and A. Newell. 1971. A model for functional reasoning in design. In *Proceedings of the Second International Joint Conference on Artificial Intelligence*. The British Computer Society, London, England, pp. 621-40.
- Kruskal, J. B., Jr., and A. Newell. 1950. *A Model for Organization Theory*. Technical Report LOGS-103. Santa Monica, Calif.: RAND Corporation.
- Laird, J., and A. Newell. 1983. *A Universal Weak Method*. Technical report, Computer Science Department, Carnegie Mellon University, Pittsburgh.
- Laird, J., and P. Rosenbloom. 1992. In pursuit of mind: The research of Allen Newell. *AI Mag.* 13(4):17-45.
- McCorduck, P. 1979. *Machines Who Think*. San Francisco: W. H. Freeman.
- Medress, M. F., F. S. Cooper, J. W. Forgie, C. C. Green, D. H. Klatt, M. H. O'Malley, E. P. Newburg, A. Newell, D. R. Reddy, B. Ritea, J. E. Shoup-Hummel, D. E. Walker, and W. A. Woods. 1977. Speech understanding systems: A report of a steering committee. *SIGART Newslett.* 62:4-8.
- Moore, J., and A. Newell. 1974. How Can Merlin Understand? In *Knowledge and Cognition*, ed. L. Gregg. Hillsdale, N.J.: Erlbaum.
- Newell, A. 1951. *Observations on the Science of Supply*. Technical Report D-926. Santa Monica, Calif.: RAND Corporation.
- Newell, A. 1955. The chess machine: An example of dealing with a complex task by adaptation. In *Proceedings of the 1955 Western Joint Computer Conference*. Institute of Radio Engineers, New York, pp. 101-108. (Also issued as RAND Technical Report P-620.)
- Newell, A., ed. 1961. *Information Processing Language V Manual*. Englewood Cliffs, N.J.: Prentice-Hall.
- Newell, A. 1962. Some problems of basic organization in problem-solving programs. In *Self Organizing Systems*, eds. M. C. Yovits, G. T. Jacobi, and G. D. Goldstein. Washington, D.C.: Spartan.

- Newell, A. 1963. Learning, generality and problem solving. In *Proceedings of the IFIP Congress-62*, pp. 407-12.
- Newell, A. 1965. Limitations of the current stock of ideas for problem solving. In *Conference on Electronic Information Handling*, eds., A. Kent and O. Taulbee. Washington, D.C.: Spartan.
- Newell, A. 1966. On the representation of problems. *Comput. Sci. Res. Rev.*, pp. 45-58.
- Newell, A. 1968. The trip towards flexibility. In *Bio-engineering--An Engineering View*, ed. G. Bugliarello. San Francisco: San Francisco Press.
- Newell, A. 1972. A theoretical exploration of mechanisms for coding the stimulus. In *Coding Processes in Human Memory*, eds. A. W. Melton and E. Martin. Washington, D.C.: Winston.
- Newell, A. 1973. Production systems: Models of control structures. In *Visual Information Processing*, ed. W. C. Chase. New York: Academic Press.
- Newell, A. 1982. The knowledge level. *Artif. Intell.* 18:87-127.
- Newell, A. 1986. Awards for distinguished scientific contributions: 1985. *Am. Psychol.* 41:347-53.
- Newell, A. 1990. *Unified Theories of Cognition*. Cambridge, Mass.: Harvard University Press.
- Newell, A. 1992. Unified theories of cognition and the role of Soar. In *Soar: A Cognitive Architecture in Perspective*, eds. J. A. Michon and A. Anureyk. Dordrecht: Kluwer Academic Publishers.
- Newell, A., and A. V. Baez. 1949. Caustic curves by geometric construction. *Am. Phys.* 29:45-47.
- Newell, A., and G. Ernst. 1965. The search for generality. In *Proceedings of IFIP Congress 65*:195-208.
- Newell, A., and J. B. Kruskal, Jr. 1951. *Formulating Precise Concepts in Organization Theory*. Technical Report RM-619-PR. Santa Monica, Calif.: RAND Corporation.
- Newell, A., and H. A. Simon. 1956. The logic theory machine: A complex information processing system. *IRE Trans. Inf. Theory* IT-2:61-79.
- Newell, A., and H. A. Simon. 1972. *Human Problem Solving*. Englewood Cliffs, N.J.: Prentice-Hall.
- Newell, A., and H. A. Simon. 1976. Computer science as empirical inquiry: Symbols and search. *Commun. Assoc. Comput. Machinery* 19:111-26.
- Newell, A., J. C. Shaw, and H. A. Simon. 1958. Chess-playing programs and the problem of complexity. *IBM J. Res. Develop.* 2:320-25.
- Newell, A., J. C. Shaw, and H. A. Simon. 1960. Report on a general problem solving program. In *Proceedings of the International Conference on Information Processing*. UNESCO, Paris, pp. 256-64.
- Newell, A., D. McCracken, G. Robertson, and R. Akscyn. 1982. ZOG and the *USS Carl Vinson*. *Comput. Sci. Res. Rev.* (Computer Science Department, Carnegie Mellon University, Pittsburgh.)
- Newell, A., J. Barnett, J. Forgie, C. Green, D. Klatt, J. C. R. Licklider, M. Munson, R. Reddy, and W. Wood. 1971. *Speech Understanding Systems: Final Report of a Study Group*. Department of Computer Science, Carnegie Mellon University, Pittsburgh.
- Newell, A., F. S. Cooper, J. W. Forgie, C. C. Green, D. H. Klatt, M. F. Medress, E. P. Neuberg, M. H. O'Malley, D. R. Reddy, B. Ritea, J. E. Shoup, D. E. Walker, and W. A. Woods. 1975. *Considerations for a Follow-on ARPA Research Program for Speech Understanding Systems*. Technical report, Computer Science Department, Carnegie Mellon University, Pittsburgh.

- Polya, G. 1945. *How to Solve It*. Princeton, N.J.: Princeton University Press.
- Polya, G. 1954. *Mathematics and Plausible Reasoning*. Princeton, N.J.: Princeton University Press.
- Rosenbloom, P. S., and A. Newell. 1982. Learning by chunking: Summary of a task and a model. In *Proceedings of AAAI-82 National Conference on Artificial Intelligence*. AAAI, Menlo Park, Calif.
- Shannon, C. E. 1950. Programming a digital computer for playing chess. *Philosoph. Mag.* 41:256-75.
- Siewiorek, D., G. Bell, and A. Newell. 1981. *Computer Structures: Principles and Examples*. New York: McGraw-Hill.
- Simon, H. A. 1993. Allen Newell: The entry into complex information processing. *Artif. Intell.* 59:251-59.
- Simon, H. A. 1996. *Models of My Life*. Cambridge, Mass.: The MIT Press.
- Walter, W. G. 1953. *The Living Brain*. New York: Norton.
- Waterman, D. A., and A. Newell. 1971. Protocol analysis as a task for artificial intelligence. *Artif. Intell.* 2:285-318.