Artificial intelligence research in Japan

Rigas, H.; Booth, T.; Briggs, F.; Murata, T.; Stone, H.S.

IEEE

Journal Name: Computer; (United States); Journal Volume: 18
https://hdl.handle.net/10945/60970

WORKSHOP

Artificial Intelligence Research in Japan

Harriett Rigas, Naval Postgraduate School
Taylor Booth, University of Connecticut
Fayé Briggs, Rice University
Tadao Murata, University of Illinois at Chicago
Harold S. Stone, IBM T.J. Watson Research Center

To develop their "Fifth-Generation" computer systems, the Japanese have had to make some hard choices. Regardless of the outcome, the project has stimulated the industry.

In this article, we describe the results of the three-year initial phase of the Japanese Institute for New Generation Research, called ICOT, as reported in November 1984 at its international conference in Tokyo.

Much publicity surrounded the founding conference in the fall of 1981 at which the phrase "Fifth-Generation Computer Systems" became a worldwide topic. The 1984 conference revealed that ICOT has made progress in developing a technology in Japan for the design and construction of advanced workstations for research in artificial intelligence. The first phase has provided the hardware and software tools for later research and has helped train a new generation of young researchers. Thus, where Japan has been behind in this technology, the initial phase of ICOT activity has brought experience and tools that can diminish or erase the gap. The first phase was intended to be a tool-building phase and was not intended to produce a leap ahead. Later phases of the project are treating the more difficult problems where innovations are essential. We should see concrete results in the next few years.

The stated goals of the 10-year Fifth-Generation Project[1] are rather ambitious for such a short period, and some early decisions on research directions have had to be made to give a focus to the project. The research is proceeding in three phases:

(1) tool-building and initial studies (three years),
(2) construction of prototype parallel machines (four years), and
(3) evaluation and refinement (three years).

The development efforts completed in the first phase have been mainly the design of two prototype machines—a sophisticated workstation known as the Personal Sequential Inference machine and a relational database machine known as the Delta machine. Supporting software systems for PSI are Kernel Language 0 and an operating-systems language built on KL0 called ESP. Explorations in advanced software for the next phase have produced an early version of a parallel knowledge language called KL1 and the Mandala language for manipulating knowledge databases. Several small applications for testing and evaluating software systems have been completed, but no single application of significant size was reported by ICOT.

0018-9162/85/0900-0083$01.00

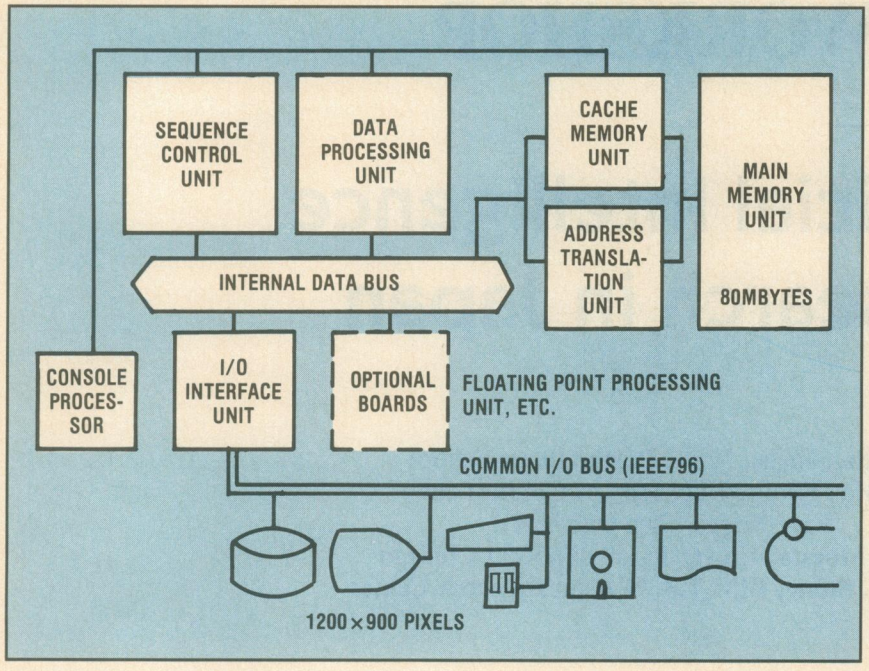


Figure 1. The structure of the PSI machine.[2]

By focusing early on the directions for research, the team hopes to have made the best decisions. They will spend the next seven years prototyping, then perfecting, the approaches, and will have little time to correct their choices. ICOT has committed its resources to develop logic programming as the principal software methodology and parallel hardware, possibly with dataflow or reduction-based architecture, as the underlying machine organization. The risk in these decisions is that so little is known about the application areas that it is far from certain that the chosen software and hardware will be effective for the applications. Outside of ICOT, researchers at this time are exploring a much broader set of possible approaches and have not yet agreed on a small collection of attractive ones. If ICOT has chosen a good set of solutions, it may advance more rapidly by having chosen the solutions early. On the other hand, if ICOT's solutions are less satisfactory than solutions uncovered elsewhere, it will be difficult for ICOT to chart a new course. Research is inherently risky, but ICOT has chosen an approach with higher than average risks. Nevertheless, research in logic programming and parallel architecture is interesting, independent of ICOT's goals. And this research might well lead to uses not foreseen at present, even if it fails to reach the intended goals.

Computers developed at ICOT

The two machines developed at ICOT during the first phase, PSI and Delta, were demonstrated at the Tokyo conference. ICOT researchers specified and designed the functional aspects of the machines. Commercial manufacturers who are members of ICOT did the detailed logic design and implementation.

The architecture of the PSI machine appears in Figure 1. Its purpose is to serve as a testbed for evaluating logic-programming architectures. Consequently, the machine is microprogrammed with writeable control store to enable the research team to explore the effects of different instruction sets and system primitives. The microprogrammed control unit need not be retained in future machines when a suitable instruction set has been engineered.

The major difference between PSI and other workstations is its large main memory, 80M bytes. This much memory is not available on commercial machines today because the cost would be prohibitively high for 1984/1985. But it is clear that the trend in memory costs makes this size memory viable when chips with 1M or 4M bits can be produced in high quantity. Having the large memory now available gives ICOT a tool that can be used to explore memory-intensive applications today, well before low-cost machines are available that make such applications feasible to run. PSI's large memory is undoubtedly the most distinctive aspect of its architecture. NEC, an ICOT industrial member, has indicated that the PSI machine (built by Mitsubishi, another industrial member) influenced their engineers to design a NEC workstation with 320M bytes to be available in late 1985. Therefore, before the ten-year project has run its course, ICOT research has already stimulated new products and development efforts by its industrial members.

Other aspects of PSI's architecture support a sophisticated man-machine interface that is typical of advanced workstations. The bit-mapped display is 1200 by 900 pixels, all points addressable, and the display software accepts keyboard and mouse inputs to control graphic images. While most features of the PSI man-machine interface have been available commercially in systems produced outside of Japan, the PSI software was produced entirely within ICOT, indicating that ICOT is developing expertise in areas where other research communities have been the principal innovators.

The memory organization is somewhat unusual in that the word length is 40 bits, which includes a 32-bit data field and an 8-bit tag field. The tag field can be used to support data typing, garbage collection, and other aspects of high-level languages that are

enhanced by a descriptive field stored with each datum. The KL0 memory system also has two caches, each 4K 40-bit words in size, and an address-translation unit that transforms a 32-bit virtual address into a physical address of 24 bits.

Not shown in Figure 1 are a few internal aspects of PSI that enhance its support of logic programming and the graphics interface. Microcode for PSI is currently written to interpret KL0, which is similar to DEC 10 Prolog. Because stack access is a frequent operation in KL0, one of the two caches is designed specifically for stack operations. Another architectural feature designed to enhance performance is a high-speed register file with 1K 40-bit words. This file holds 16 general registers, and the remainder of the file is organized into buffers and work areas. The file includes, for example, two stack-frame buffers, each with 32 words, that are filled with local data when new clauses are invoked. Other details of the PSI architecture appear in the paper by Taki et al.[2]

In a companion paper, Yokota et al.[3] describe the microprogrammed interpreter for KL0 that runs on PSI. The interpreter has been strongly influenced by Warren's work on Prolog compilers.[4] Like many Prolog compilers, the interpreter generally uses structure sharing to eliminate the inefficiency of copying structures, but uses argument copying during unification. This eliminates some overhead for recalculation of arguments when backtracking occurs, and provides a means for exploiting tail-recursion optimization.

The relational-database machine, Delta,[5,6] is intended to support the use of knowledge bases in artificial intelligence applications. ICOT views a total system as containing many PSI-class machines tied together through a high-speed network to a few Delta-class machines that hold a knowledge base shared among the workstations. A block diagram of the Delta described at the conference appears in Figure 2. It consists of a network-interface processor, control processor, three relational engines, a 128M-byte main memory, a 20G-byte auxiliary memory, and other associated control and interface hardware. Delta has a network interface that serves as its principal input/output port to the external world. The network-interface processor attaches to an internal IEEE-488 bus shared by the relational engines and control processor.

The basic operation of Delta is to respond to relational queries it receives from the network. Queries are transferred to the control processor as they are received. There, the queries are broken into primitives executable by the relational engines. The primitives are high-level relational operations such as SORT, UNION, JOIN, and RESTRICT, which is an unusual design feature of Delta. One or more of the engines can be assigned operations concurrently, and the operations proceed in parallel unless consistency constraints force sequential execution. The relational engines include special hardware for sorting and merging, to enhance the performance of complex operations such as JOIN.

Shibayama et al.[6] describe Delta's transaction-processing functions in some detail. The machine appears to be oriented to high-speed operations on entire relations, and the speed obtained comes from a combination of special hardware (such as the sorting hardware in the relational engines) and a large high-speed memory that serves as a disk cache. Many database researchers have attempted to reduce the amount of data fetched from auxiliary memory so that the actual number of operations required to process a query

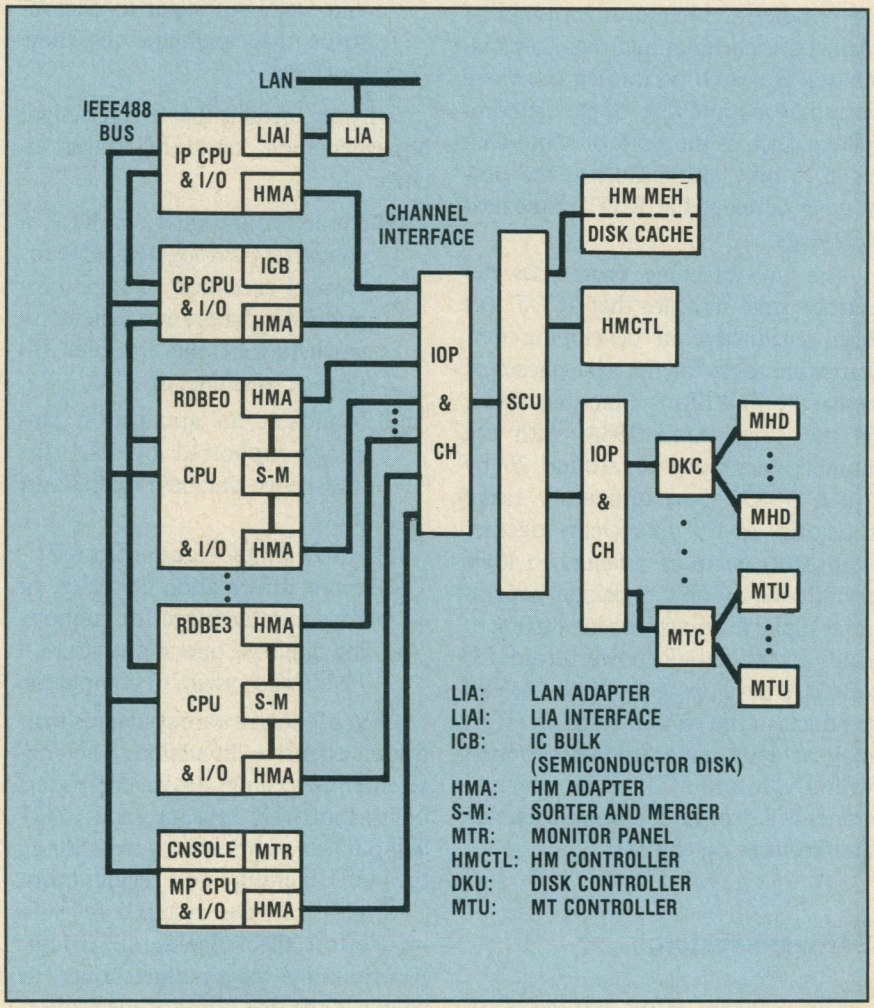


Figure 2. The Delta machine architecture.[6]

is much smaller than the number of operations needed to examine a full relation. A survey of such techniques appears in Jarke and Koch,[7] and Tanaka's proposed database machine[8] is compatible with such schemes. The ICOT papers suggest that the present implementation of Delta software operates on entire relations, and the papers do not give sufficient detail to determine what kind of optimization is possible on Delta to reduce auxiliary-memory access.

Both prototype machines PSI and Delta are principally serial computers. Parallelism on PSI is limited mostly to conventional mechanisms for overlapping I/O and computation. Delta has a few independent processors that provide a means for studying parallel execution, but parallelism is very limited and can increase performance by at most a small multiplier. Nevertheless, ICOT has pressed forward with studies of parallel architectures, notably of reduction machines and dataflow machines. Research performed elsewhere in Japan on prototypes of parallel computers, such as the work of Moto-Oka et al.,[9] may also influence the prototypes developed at ICOT in the next few years.

The two machine prototypes described here indicate that ICOT has been productive in developing new hardware tools for the next phases of research. Six PSI machines have been in use since April 1984, with the number increasing to around 20 by April 1985. The unusually large memories of PSI and Delta together with other features oriented to logic programming and relational operations make the machines interesting to study and potentially powerful aids for software development. While the end products of the 10-year research effort may actually have little in common with Delta and PSI, these two machines will strongly influence whatever architectures do emerge.

Software systems

The software effort has been devoted in part to research studies in logic programming and in part to the actual development of software for PSI and Delta. The most visible contributions have come from the prototype software that has been created in the last three years, but the work in theoretical foundations may prove to be crucial as well in the coming years.

The following software systems were presented in papers and demonstrations at the conference:

(1) Kernel Language 0, or KL0, the Prolog derivative used as the machine language for the PSI,

(2) Extended Self-Contained Prolog, or ESP, an object-oriented extension of KL0 used for systems programming on the PSI,

(3) Sequential Inference-Machine Programming and Operating System, or SIMPOS, the operating system for the PSI, and

(4) Other support software for PSI and Delta sufficient to demonstrate the capabilities of these machines.

Software systems that are in design, implementation, or testing stages include

(1) Kernel Language 1, or KL1, a concurrent-Prolog derivative for parallel execution of logic programming for the next generation workstation, the Parallel Inference Machine, or PIM,

(2) Mandala, an application language supported by KL1 for knowledge database operations, and

(3) Knowledge Acquisition-Oriented Information Supplier, or Kaiser, a language for supporting database operations across a PSI/Delta system of computers.

Each of the software systems is worthy of extensive discussion. The conference proceedings has several papers on the software systems, and ICOT will publish additional information in the public literature on a regular basis.

Apart from the systems development effort, theoretical studies in logic programming have matured over the course of the first phase of the project. Shapiro's Concurrent Prolog[10] was the basis of initial proposals for KL1. ICOT's research in concurrency may reach fruition in the next several years in the form of a much evolved KL1 system running on PIM. There are still many practical obstacles to overcome in developing PIM and associated programming systems. ICOT's current research on KL1 and PIM is devoted to the development of working prototypes.

Among the papers presented on theoretical foundations was one by Yasuura of the University of Kyoto that shows that the unification operation, a key primitive operation for logic programming languages, has a parallel complexity equal to its serial complexity.[11] Dwork et al. made a similar discovery.[12] The result reveals that in the worst case a unification operation can be sped up by a parallel machine by at most a constant amount no matter how many processors are used for the unification. This suggests that unification is inherently serial, and casts doubt on the use of parallelism to speed up unification. Actually such a conclusion is unduly negative because the complexity result is a comparison of worst cases only. Average behavior may be quite different and more meaningful. Moreover, parallelism is not limited to speeding up individual unification operations; many unifications can be computed in parallel as well. Nevertheless, Yasuura and Dwork et al. give some insight into the parallel execution of the unification operation.

Architecture research

The performance goals for the 10-year project call for machines capable of roughly 10,000 to 100,000 times the performance of 1982 machines with throughput measured in terms of logical inferences per second. Since advances in basic devices might contribute only a factor of 10 to 100 in that time frame, by necessity ICOT has devoted considerable attention to parallel machines. The first phase of the research effort has narrowed the study to a few different architectures,

including the reduction architecture and dataflow architecture for inference engines and parallel implementation of relational database operations. Delta is a first step in the latter area in that it can support small-scale concurrency with its multiple relational-database engines.

For parallel inferencing, at least three forms of parallel execution are being examined:

(1) AND-parallelism,

(2) OR-parallelism, and

(3) stream parallelism.

AND-parallelism refers to following two or more paths concurrently, all of which must succeed in a consistent way in order for the total computation to succeed. The challenge is to make sure that multiple concurrent computations exchange information efficiently so that the separate computations continue to be consistent as they progress. There is a danger in following multiple paths because if any one path fails, then the work in progress by other processors must be abandoned as wasted effort. Although parallel execution may keep many processors busy, the wasted effort diminishes the net computational speedup. Wasted effort must be controlled, and one can easily construct pathological examples in which wasted work significantly exceeds useful work when no effort is made to direct processors to useful work.

OR-parallelism is somewhat easier to implement than AND-parallelism because it permits independent computational paths to be executed in parallel. If any one of these paths succeeds, then the mutual goal succeeds. Moreover, it is not necessary to exchange information in the course of a parallel execution to assure consistency. But like AND-parallelism, OR-parallelism does lead to wasted effort if some computational paths are terminated when another path reaches a goal successfully. This occurs when many possible solutions to a subproblem exist, but only one is required for the next goal. If all possible solutions must be found, OR-parallelism improves performance by finding the set of solutions faster. When an OR-parallel execution has the potential for wasted work, then, as for AND-parallelism, it is essential to control execution.

Stream parallelism, the third technique, refers to a mode in which a computation can be structured as a pipeline. In this case, an initial process produces a continuous stream of candidate data, each of which goes through a succession of processing stages. Parallel computation can be performed on the stream by using separate processors for different stages of the computation. Each processor operates on a different datum, and data flows from stage to stage as in a pipeline computer. In more elaborate implementations, stream parallelism can accommodate the merging of two or more streams and the internal generation of two or more streams. With stream parallelism, if an intermediate goal is reached, then the data in process by earlier stages are abandoned, which limits the amount of wasted effort to the computation required to fill the buffers of processors whose computations are terminated. One natural way of obtaining stream parallelism during execution of AND-parallel constructs is to introduce read-only variables on clauses to force sequential execution within one instance of an evaluation, while permitting distinct instances to be executed in parallel. Shapiro's Concurrent Prolog[10] and Clark and Gregory's PARLOG[13] both support stream parallelism in this fashion in an environment designed for high concurrency.

The challenge is to make sure that multiple concurrent computations exchange information efficiently so that the separate computations continue to be consistent as they progress.

Very little is known today about the relative effectiveness of these approaches. ICOT will be building prototypes and measuring performance in the coming years. Meanwhile, the parallel inference-engine architecture under study at Tokyo University[9] is providing simulation data for an OR-parallel approach. They report a speedup of 170 for a simulated 256-processor system executing the eight-queens problem, which is a notable result. However, other problems exhausted the exploitable parallelism and attained only much lower speedups. These results suggest that problem structure and choice of algorithm have a dramatic effect on speedup for this particular OR-parallel design. The research questions remain open concerning how to identify which problems are well suited and which are poorly suited to OR-parallelism execution, and how to design efficient OR-parallel algorithms that yield the best possible problem speedup.

In the area of parallel database machines, ICOT plans to first gain experience with Delta and then identify how to expand its capabilities. The work on Delta was influenced in part by the work of Tanaka at the University of Hokkaido whose invited paper[8] described a massively parallel database architecture that is somewhat more sophisticated than the Delta prototype. His sorting and searching engines that are central to this proposed architecture have been running in prototype forms, but the total system is still a paper design.

Applications

The principal motivation for the ICOT research effort is to create a technology for new applications, primarily applications related to artificial intelligence and expert systems. One of the themes of the conference papers was applications. Among the demonstrations of the PSI were rule-based systems written in KL0 that performed some nontrivial tasks. For example, one program analyzed a one-voice musical melody and prepared a four-voice harmony for that melody. While the program itself is a rather modest accomplishment, it does demonstrate that the PSI hardware and

software function today as an integrated system for developing applications.

Because of its extensive effort devoted to machine design and implementation of systems software, ICOT itself has not devoted significant energy to applications; rather, the applications reported at the conference were developed throughout the world, though laboratories in Japan were well represented. Among the areas where significant progress would have a major impact on future computers is design automation. Papers on expert-system techniques applied to design automation were delivered by Fujita et al. of Tokyo University,[14] Maruyama et al. of Fujitsu,[15] Mori et al. of NEC,[16] and Poe of Apple.[17] The first three papers treat logic synthesis and verification, while Poe examines microcode synthesis.

The general thrust of the work in design automation is that the complexity of designs is growing tremendously, spurred by the ever-growing density achievable in VLSI fabrication. When automation was first introduced in the design process, it aided tremendously in dealing with complexity and basic bookkeeping, but most automated-design systems still require substantial human intervention in routine, low-level design tasks. Expert-system techniques have the potential for eliminating the human element at the lowest levels of design. For example, Mori et al.[16] treat the wire-routing problem. Automatic layout-programs based on maze-running algorithms can produce wiring layouts that contain nearly all specified conductors of fairly complex multilevel board and chip designs. But rarely do these techniques succeed in routing 100 percent of the wires. The NEC researchers describe how they have used a rule-based approach to route the last few conductors automatically. The rules describe possible ways of altering a given layout to create room for additional conductors in specified regions.

Other themes in the application area were knowledge representation, intelligent information retrieval, and natural language understanding. Although ICOT researchers were not well represented in this area of conference activity, a number of application demonstrations on ICOT systems by non-ICOT researchers indicate that ICOT has good working relations with external researchers who can provide applications experience to ICOT.

Research activities in Japan

Although ICOT is the most publicized computer research group in Japan, its focused research is not representative of the general breadth of activity. Universities have studied parallel architectures, for example, and have working prototypes of a great variety of parallel systems. Hoshino's array computer at Tsukuba, Tanaka's database machine at Hokkaido, and Moto-Oka's PIE machine at Tokyo are representative of the scope of the activity. The research work at the universities has relatively little student involvement, and the students who do participate are Master's level more often than not. This is the case because university production of PhD's is very low; most students find that a Master's degree is sufficient for launching a career in Japan. A similar situation exists in the United States today, except that the pool of PhD students in the US is many times larger than that of Japan, and consequently, university research projects in the US tend to have a greater proportion of student involvement.

One artificial intelligence system combines extended Prolog and a frame-based graphics system to create a novel system oriented to high-level computer-aided design.

Publicly funded laboratories are another extension of the Japanese research community. The Electrotechnical Laboratory in Tsukuba is one such well-known laboratory. Work at ETL is broadly based. Work in computer systems today includes research in expert systems and parallel dataflow architectures, both of which are complementary to work now being done at ICOT. Kazuhiro Fuchi, the founder and director of ICOT, came originally from ETL.

The publicly-owned Nippon Telephone and Telegraph Company also sponsors its own research laboratories. The laboratory developments are not made directly into products, because NTT does not manufacture products for sale. Rather, NTT does exploratory research and advanced development to create the technology for other companies to use in their manufacture of products for sale to NTT. In a sense, NTT is acting for the public good much like a public laboratory like ETL, except that the main motivation for research is to advance the state of communication systems.

Like many other private and public companies throughout the world, NTT has been investigating the use of artificial intelligence techniques for dealing with present and future problems. Ogawa et al. of NTT's laboratory at Musashino, for example, describe KRINE, an environment for knowledge representation and inferencing. This system combines extended Prolog and a frame-based graphics system to create a novel system oriented to high-level computer-aided design. Graphics aspects of KRINE support displaying and modifying designs, while its Prolog aspects provide an environment for incorporating inferencing into design algorithms and graphic display.

NTT's role in research is bound to change as it changes from a public company to a private company, which is scheduled for 1985, but the deeper effects of this change on research at NTT are not known today.

Industry contributes a substantial amount to research activity in Japan, mainly because the majority of the technological work force is housed within the companies. Each of the companies that are industrial members of the ICOT project maintains its own research program. The research programs may well encompass projects that support and extend work at

ICOT, as, for example, NEC's announced work on an advanced version of a PSI machine. Where ICOT's research is focused in particular directions, industrial members of ICOT are free to pursue other directions. The papers at the conference in November suggest that industrial researchers are paying a good more attention to practical applications of artificial intelligence than ICOT has. Industrial researchers may well produce totally different architectures and systems from those being studied at ICOT, although it is much more likely for the laboratories to develop ideas based on ICOT's work.

Because the industrial research effort is privately funded, published output is more controlled, and the research is less exposed to public view than is the research at ICOT. The research effort is rather extensive, and is strongly influential in charting the course of technology in Japan. Consider, for example, recent offerings of high-end supercomputers from Fujitsu and Hitachi, soon to be joined with an offering from NEC. These computers enter a market formerly supplied largely by Cray and Control Data. Entry in this market is an unusual event, and it is remarkable to see three new entrants in the course of two years.

One effect of the wide publicity garnered by ICOT is to diminish the publicity given to developments elsewhere in Japan, many of which are worthy of attention. In reviewing research achievements in Japan, it is essential to look beyond the work at ICOT to the universities, to the public laboratories, and to industry to obtain an accurate picture of technological development.

The great publicity given to ICOT in the last three years has resulted in a much greater awareness of the field of artificial intelligence. AI research programs in many countries have been established since ICOT's founding, funded both publicly and privately, with the work at ICOT providing a significant stimulus for starting such programs. Our comments here suggest that ICOT has spent significant energy in closing a technological gap and now is attempting to move the frontiers of artificial intelligence in directions of its choosing. But meanwhile that effort has stimulated an effort many times larger, including activity elsewhere in Japan, and in the US, Great Britain, and Western Europe. The research frontier indeed appears to be moving, but in many different directions worldwide, and no single approach has yet emerged as distinctly superior to all others.

Having developed advanced prototypes for inference workstations and database machines, ICOT currently has a unique set of tools to use in its next phase of research. In the next phase of activity, ICOT plans to use these tools to make their first plunge into highly-parallel prototype architectures and software systems. The research challenges that lie ahead in the next few years are many and difficult, and will require far more innovation than has been necessary to complete the prototypes delivered in the first phase of research. Just how successful ICOT might be in the next phase will become clear in 1988 at the next ICOT symposium. Whether or not ICOT makes substantial progress towards its goals by then, the tremendous growth in activity in the area spurred at least in part by ICOT may well result in remarkable achievements. So ICOT may be responsible at least indirectly for setting the stage for advances in artificial intelligence, in addition to whatever contributions it makes directly through its own efforts.

Acknowledgments

The material in this report was developed by the authors during a visit to Japan in conjunction with the 1974 conference. Harriett Rigas headed the delegation of authors, and the National Science Foundation sponsored the visit. We thank Bernard Chern of the National Science Foundation for making this trip possible. We also express our gratitude to our several hosts in Japan for meeting with the delegation for technical discussions. We give special thanks to Makoto Nagao and Shuzo Yajima of the University of Kyoto, Tsutomo Hoshino of the University of Tsukuba, Tosiyasu Kunii of the University of Tokyo, Mario Tokoro and Hideo Aiso of Keio University, Saitoshi Goto of NEC Corporation, and to the researchers at ICOT for their hospitality and for opening their laboratories to the delegation.

References

1. T. Moto-oka et al., "Challenge for Knowledge Information-Processing Systems," *Proc. Int'l. Conf. Fifth-Generation Computer Systems,* North-Holland, Amsterdam, 1982.
2. K. Taki et al., "Hardware Design and Implementation of the Personal Sequential Inference Machine (PSI)," *Proc. Int'l. Conf. Fifth-Generation Systems,* Tokyo, 1984, pp. 308-409.
3. M. Yokota et al., "A Microprogrammed Interpreter for the Personal Sequential Inference Machine," *Proc. Int'l. Conf. Fifth-Generation Systems,* Tokyo, 1984, pp. 410-418.
4. D.H.D. Warren, "Implementing Prolog—Compiling Predicate-Logic Programs," D.A.I. Research Rpt. No. 39-40, Dept. of Artificial Intelligence, University of Edinburgh, Scotland, 1977.
5. H. Sakai et al., "Design and Implementation of the Relational Database Engine," *Proc. Int'l. Conf. Fifth-Generation Systems,* Tokyo, 1984, pp. 419-426.
6. S. Shibayama et al., "Query Processing Flow on RDBM Delta's Functionally-Distributed Architecture," *Proc. Int'l. Conf. Fifth-Generation Systems,* Tokyo, 1984, pp. 427-435.
7. M. Jarke and J. Koch, "Query Optimization in Database Systems," *ACM Computing Surveys,* Vol. 16, No. 2, June 1984, pp. 111-152.
8. Y. Tanaka, "MPDC: Massive Parallel Architecture for Very Large Databases," *Proc. Int'l. Conf. Fifth-Generation Systems,* Tokyo, 1984, pp. 113-137.
9. T. Moto-Oka et al., "The Architecture of a Parallel Inference Engine—PIE," *Proc. Int'l. Conf. Fifth-Generation Systems,* Tokyo, 1984, pp. 479-488.

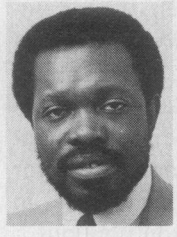
10. E.Y. Shapiro, "A Subset of Concurrent Prolog and its Interpreter," ICOT Report TM-0003, ICOT, Tokyo, Nov. 1982.
11. H. Yasuura, "On Parallel Computational Complexity of Unification," *Proc. Int'l. Conf. Fifth-Generation Systems,* Tokyo, 1984, pp. 235-243.
12. C. Dwork, P.C. Kanellakis, and J.C. Mitchell, "On the Sequential Nature of Unification," *Journal of Logic Programming,* Vol. 1, No. 1, June 1984.
13. K.L. Clark and S. Gregory, "Parallel Programming in Logic," Research Rpt. DOC 84/4, Dept. of Computing, Imperial College of Science and Technology, April 1984.
14. M. Fujita et al., "Specifying Hardware in Temporal Logic and Efficient Synthesis of State-Diagrams Using Prolog," *Proc. Int'l. Conf. Fifth-Generation Systems,* Tokyo, 1984, pp. 572-581.
15. F. Maruyama et al., "Prolog-Based Expert System for Logic Design" *Proc. Int'l. Conf. Fifth-Generation Systems,* Tokyo, 1984, pp. 563-571.
16. H. Mori et al., "Knowledge-Based VLSI Routing System—WIREX" *Proc. Int'l. Conf. Fifth-Generation Systems,* Tokyo, 1984, pp. 383-388.
17. M.D. Poe, "Control of Heuristic Search in a Prolog-Based Microcode Synthesis Expert System," *Proc. Int'l. Conf. Fifth-Generation Systems,* Tokyo, 1984, pp. 589-595.
18. Y. Ogawa et al., "Knowledge Representation and Inference Environment: KRINE—An Approach to Integration of Frame, Prolog, and Graphics," *Proc. Int'l. Conf. Fifth-Generation Systems,* Tokyo, 1984, pp. 643-651.

Harriett B. Rigas is chairman of the Department of Electrical and Computer Engineering at the Naval Postgraduate School in Monterey, California. She obtained a BSc in electrical engineering from Queen's University in Canada in 1956 and an MSEE and PhD in electrical engineering from the University of Kansas in 1959 and 1963, respectively. After two years at the Lockheed Missile and Space Company in Sunnyvale, she joined Washington State University, where she remained until 1984 and served as chairman from 1980 to 1984.

Rigas is recipient of the 1982 Society of Women Engineers Achievement Award. She is an IEEE Fellow.

Taylor Booth is a professor of computer science and engineering and director of the Computer Applications and Research Center at the University of Connecticut. His research interests include the analysis and design of high-performance software and computer systems. He has authored or coauthored three books and numerous papers in the computer field. He is the former editor-in-chief of the *IEEE Transactions on Computers.* He has been a member of the IEEE Board of Governors and has served as secretary, vice president for education, and first vice president of the IEEE CS.

He was awarded the Terman Award in 1972, the IEEE Centennial Medal in 1984, and is listed in *Who's Who in America.* He received his PhD from the University of Connecticut in 1962. He is a fellow of the IEEE and a member of the ACM and ASEE.



Fayé A. Briggs is an associate professor in the Department of Electrical and Computer Engineering at Rice University. His research interests include parallel computer architecture and performance evaluation of multiprocessing computer systems. He was formerly an assistant professor of computer engineering and the systems manager of Advanced Automation and Research Laboratory at Purdue University. He is the coauthor of *Computer Architecture and Parallel Processing.* He has also served as a consultant to IBM T. J. Watson Research Center, Exxon Production Research Company, and Texas Instruments.

Briggs is on the editorial board of the *Journal of Parallel and Distributed Computing.* Since 1983, he has been a Distinguished Visitor of the IEEE Computer Society. He also served as the track chairman of computer architecture and hardware at the 1984 NCC. He received the PhD degree from the University of Illinois in electrical engineering.

Tadao Murata is a professor of electrical engineering and computer science at the University of Illinois at Chicago. His current research interests include theory and applications of Petri nets, concurrent computer systems, and new generation computer architectures. He received the MS and PhD degrees from the University of Illinois at Urbana, in 1964 and 1966, respectively, and is an IEEE Fellow.

Harold S. Stone is the manager of advanced architecture studies at IBM T.J. Watson Research Center. He has been a faculty member at the University of Massachusetts and Stanford University and has held visiting faculty appointments at institutions throughout the world. He is the author, coauthor, or editor of six textbooks and over 60 technical publications. As a consulting editor to Addison-Wesley, McGraw-Hill, and University Microfilms, he has produced more than 70 titles in all areas of computer science and engineering. His research contributions have been primarily in computer architecture and digital systems design.

Stone received a PhD in electrical engineering in 1963 from the University of California at Berkeley. He is a member of IEEE and ACM, and has served as technical editor of *Computer* and as a governing board member of the IEEE CS.

Questions about this article can be directed to Harriett B. Rigas, Dept. of Electrical and Computer Engineering, Code 62, Naval Postgraduate School, Monterey, CA 93943.