Report on ICOT

Mark E. Stickel
June 9, 1992

With the generous sponsorship of the U.S. National Science Foundation, I had the privilege of visiting the ICOT Research Center and participating in its research program for nine months during September-November 1990, March-May 1991, and November 1991-February 1992. It was a wonderful personal and professional experience. The people at ICOT were unfailingly kind and friendly. I would especially like to thank the people I worked most closely with: Masayuki Fujita and Ryuzo Hasegawa. I had many interesting technical and personal discussions with them and saw in them, and their colleagues, the time, energy, interest, and intelligence they invested in their research. Fujita and Hasegawa shared my love of theorem proving research. I would also like to thank Koichi Furukawa, whom I met before coming to ICOT, who stimulated my interest in visiting ICOT, and with whom I also had many technical and personal discussions while at ICOT. He and Dr. Fuchi created the excellent research climate at ICOT and fostered the work on theorem proving there. I would like to thank Kazuhide Iwata for his friendship and help with the daily details of living in Tokyo, including arranging for an apartment. I had good personal and professional interactions with too many other people at ICOT to name here.

I was briefed on the research activities of the various laboratories at ICOT and found much high quality research that was of interest to me: constraint logic programming, knowledge representation, natural language processing, parallel software and hardware, and, of course, theorem proving. In the end, though I would have enjoyed working on any number of activities at ICOT, I concentrated on theorem proving, the area of strongest personal interest and an area where I thought I could contribute the most while also demanding the least time of ICOT researchers to educate me in what they were doing.

On various occasions, I lectured on the Prolog Technology Theorem Prover (PTTP), cost-based abductive inference, equality theorem proving, theory resolution, upside-down meta-interpretation of model elimination, and theorem proving in general. While at ICOT, I researched and wrote about upside-down meta-interpretation of model elimination, a unit-resulting extension of PTTP, and function and relation matching rules for building in theories. I discussed ICOT's MGTP, the earliest versions of which had already been written. I discussed the value and importance of term indexing to improve

efficiency of theorem proving and described Argonne's approach to theorem proving. I performed many experiments with theorems ICOT was working on using Argonne's OTTER and my own PTTP. I implemented discrimination-net-based term indexing in KL1; this was used in some versions of MGTP. Aspects of the Argonne approach that I described and praised to them also found their way into versions of MGTP.

A focus of theorem proving research at ICOT was to prove a set of theorems offered by Argonne's Ross Overbeek as challenge problems to test the performance of theorem proving programs. These are difficult problems solved by few programs. During and between my visits to ICOT, I saw steady progress being made on solving these problems. They were first solved by use of many heuristics, then without heuristics but with very poor parallel speedup, and finally solved with near-linear speedups without heuristics. A proof "look-ahead" capability often allows the ICOT prover to solve problems after generating many fewer clauses than Argonne's OTTER.

These problems were solved by a "nonground" MGTP designed to solve such non-range-restricted Horn problems. The other, original "ground" MGTP is designed for range-restricted non-Horn problems. Ground MGTP is based on Manthey and Bry's SATCHMO theorem prover and is basically a hyperresolution theorem prover that performs case-splitting on non-unit, positive derived clauses. Case splitting is feasible because the range restriction ensures that derived clauses are ground, so there is no problem with variable sharing between cases. Ground MGTP is especially well suited to implementation in KL1 on PIMs. The range restriction implies that whenever a pair of terms is unified, at least one of those terms will be ground (variable free). This permits the efficient implementation of theorem-proving variables by KL1 logical variables and the use of KL1's one-way, single assignment unification. By contrast, nonground MGTP required a unification algorithm, with theorem-proving variables represented by integers and substitutions represented by vectors indexed by these integers, be written in KL1, resulting in a costly slowdown. Case-splitting in ground MGTP provides lots of work for many processing elements, with low communication requirements, thus making it easy to achieve high speedup factors. In contrast, the successful achievement of high speedup factors for the nonground MGTP required much experimentation and refinement of work distribution schemes.

Ground MGTP is still a niche theorem prover, well-suited only for range-restricted, non-Horn problems. It is motivated by the case-splitting possibilities of non-Horn problems, so it offers nothing extra for Horn problems. Although any problem that is not initially range-restricted (every variable of a positive literal of a clause must also appear in a negative literal of the clause) can be translated into one that is by adding a "dom" predicate that inductively defines ground terms of the domain and qualifying non-range-restricted clauses with "dom" literals, this is usually not very effective. Nevertheless, range-restricted non-Horn problems appear to be a useful niche. Evidence of this comes in the form of ICOT's recent solution of an open problem in mathemat-

ics. MGTP thus joins the very short list of theorem provers that have solved open problems. The contribution of ICOT's parallel hardware to this proof is noteworthy as well. One of these open problems was solved in 3 hours on a 256-processor PIM; on a single processor of the same power, this would have required waiting a month for the solution. This represents a qualitative difference in the theorem proving process that makes otherwise nearly unthinkable tasks doable. Inoue and others have also been doing excellent work in demonstrating the usefulness of MGTP-style reasoning for nonmonotonic reasoning, diagnosis, etc. Many AI reasoning problems seem naturally formulatable for execution by ground MGTP. Upside-down meta-interpretation permits the bottom-up MGTP to be imbued with goal-directedness.

Outside of theorem proving, I think ICOT's contributions are many. ICOT's scientific contributions, particularly in the area of logic programming languages, practice, and theory, are competitive with that of other research institutions around the world. ICOT is an internationally recognized research center. Through foreign visitors coming to ICOT, ICOT researchers visiting overseas, ICOT's organization and participation in conferences and workshops, publication of research results in technical reports, conference proceedings, and journals, often in English, ICOT and Japan are participating strongly in the international computer science community. ICOT's making the software developed in the Fifth Generation Project freely available is an extension of ICOT's deliberate policy of openness and is a noteworthy and generous contribution to the scientific community.

I think ICOT has been very successful in training its researchers in logic programming, parallel processing, and the methods and values of computer science research. Perhaps they can bring about greater commercial use of logic programming when they return to their companies. Still, the barriers to new programming methodologies in industry seem high. With my long experience programming in LISP as well as Prolog, industry's failure to recognize the value of alternative methods has long been obvious and disappointing. Considerations other than technical merit often determine what is successful, such as MS-DOS and C.

ICOT's objectives seem fundamentally right. Parallel processing is the right way to provide lots of computing power cost-effectively. MIMD architectures are more easily used for a variety of applications than more restrictive computational approaches. A focus on symbolic computing applications is a needed counterpoint to the usual emphasis on supercomputing for numeric, scientific applications. The KL1 programming language is a major accomplishment of ICOT. It is an elegant parallel logic programming language that facilitates writing parallel programs while easily avoiding synchronization errors. Writing PIMOS and earlier operating systems entirely in logic programming languages is an massive demonstration of the suitability of such languages for low-level systems programming as well as high level applications programming.

Full exploration of the capabilities and limitations of a concurrent logic programming approach demanded that everything be written in KL1 as a research methodology. After writing out as much information and developing as many programming techniques as we can this way, I think that acceptance of KL1 in the marketplace will be enhanced if KL1 procedures can be combined with procedures in other languages. This could save the cost of rewriting in KL1 procedures already available in other languages. Also, sometimes performance can be significantly improved by rewriting small portions of systems in a lower-level language. The performance loss of the heavily used unification algorithm written in KL1 in nonground MGTP suggests a case in which performance could be improved substantially.

ICOT also created experimental hardware to support parallel symbolic processing. However, the wave of language-specific processors seems to have crested and past. In the LISP world as well as the Prolog and logic programming world, there now seems to be little interest in special processors. While language-specific processors can certainly deliver superior performance compared to general-purpose processors, less money and resources are available for their development than for general-purpose processors. Development costs of specialized processors must be spread over a much smaller market, often rendering them uncompetitively expensive. The revenues generated by a mass-marketed general-purpose processor can provide funds to improve its performance even on tasks for which it somewhat ill-suited, enough to ultimately become competitive with specialized processors.

The lack of commercial appeal of ICOT's prototype Parallel Inference Machines (PIMs) is a direct result of the decision to use specialized processors. In this respect, the hardware group was asked to play a supporting role in the project by providing hardware designed around ICOT's software research effort instead of designing machines with wider appeal by using more standard processors. ICOT succeeded in building machines with hundreds of powerful processors and achieved the goal of building a machine delivering 100 megalips of performance. No more suitable machine for ICOT's work is available: the specialized processors do provide a performance gain for KL1 over standard processors, and other large MIMD processors aren't really quite available yet. PIMs and the earlier Multi-PSI machines provided the necessary testbed for ICOT's research, providing high performance, reliability, and availability.

I think the widespread propagation of the technology that ICOT has developed depends on porting it to commercially popular architectures. KL1 should be ported to standard processors. Standard, commercially available large multiprocessors don't exist yet, but ICOT's system and application software should be ported to run on them when they do. Besides providing the software ICOT developed, ICOT should find some means of instructing others in the programming methodologies they used to write huge systems using KL1.

The inception of ICOT was accompanied by great expectations. ICOT certainly failed to solve "the AI problem" and thus be viewed as an unqualified success by the world press. But neither has anyone else, and ICOT has contributed as much as other research centers. In its research approach, ICOT was always willing to build things: applications, languages, operating systems, processors, multiprocessors. They did not restrict themselves to developing paper theories, but realized them in hardware and software. Implementation is a good test of value of ideas, and I think ICOT's willingness to experiment with the technologies they devised is very healthy.

I hope that ICOT continues. Establishing a research center with an international reputation is no small task. The investment to develop the research center, to establish a core of researchers and managers, a set of operating procedures, and a culture, has been made and should be preserved. The PIM multiprocessors have only recently been completed, so there has been little opportunity to experiment with or evaluate them yet. More effort is required to propagate ICOT's ideas and software to the world.