

# ICOT Evaluation Report

Evan Tick  
Department of Computer Science  
University of Oregon  
Eugene, OR 97403 USA

June 16, 1992

## 1 Introduction

This report summarizes my views of the Fifth Generation Project (FGCS) conducted at ICOT over the period of 1982–1992. My participation is somewhat unique because I was both an ICOT visitor in February 1987 and then a recipient of the first NSF-ICOT Visitors Program grant, from September 1987–September 1988. I joined the University of Tokyo, in the Research Center for Advanced Science and Technology (RCAST) at that time, with a visiting chair in Information Science donated by the CSK Corp. Thus over the period of 1986–1989 I had an “insider’s view” of the FGCS Project. My area of research concerned performance evaluation of parallel logic programming paradigms [12, 15, 8, 11, 9, 10, 17], primarily working with M. Sato. I also worked on multiprocessor cache protocols with A. Goto and A. Matsumoto [3] and compile time estimation of task granularity in concurrent languages [7, 13]. During my stay at ICOT I had the opportunity to begin writing a book describing then state-of-the-art approaches to parallelization of logic programs [14].

Furthermore, my collaboration with ICOT researchers continued after my return to the U.S. During that period, the primary research was in the area of parallel garbage collection (with A. Imai [5, 4, 6]) and continued shared interest in distributed parallel algorithms, specifically for the best path problem (with N. Ichiyoshi [16, 1, 2]). (I included a bibliography of my own work associated with ICOT in order to convey the impact and importance on my own line of research. Of course, ICOT members have published extensively and produced results in numerous areas.)

My experience as a Stanford post-doctoral research associate, working full-time at ICOT in 1988, was very rewarding. I had previously worked at IBM Yorktown Heights (with T. Agerwala and D. DeGroot), and at SRI Menlo Park (with D. H. D. Warren), and the ICOT research environment and academic comraderie was on par with these institutions. Summaries of the FGCS Project successes and failures by most foreign researchers tend to categorize the abstract vision (of knowledge engineering and a focus on logic programming) as a great success and the lack of commercially competitive hardware as the main failure. I would like to comment on more subtle successes that fewer observers had a chance to evaluate. These successes, as

alluded to by Dr. Fuchi in some of his remarks in the Evaluation Workshop, involve the training of a generation of computer scientists.

## 2 A New Generation

In these preliminary remarks, I will not go into the numerous details and implications of this “side effect” of the FGCS Project (I leave that detailed analysis for a journal article I am preparing on the subject). I will summarize the main points as I see them from the vantage point of working shoulder-to-shoulder with ICOT members.

1. increased communication culture — ICOT infrastructure was unique for Japanese research groups in the early 1980's in that it supplied researchers with various communication channels that normally did not exist in the corporate culture:
  - company-to-company interaction engendered by the cooperative efforts of engineers from several companies.
  - electronic mail, increasing international as well as local information flow.
  - company-to-university interaction engendered by the Working Groups (initiated at the inception of the project). A related point is PhD production, discussed below. I know best that Prof. Tanaka of the University of Tokyo has a close research relationship to ICOT, working on design of inference multiprocessors and languages throughout these past ten years. Other professors have also influenced and are influenced by ICOT research.
  - Japan-to-international research community interaction engendered by the high value placed on the publication and presentation of research results.
2. post-graduate education — ICOT served as a substitute for OJT (“on-the-job training”), and in doing so, graduated a generation of engineer/managers well educated in advanced areas of computer science and better able to manage their own groups in the future. The latter point applies to both the management of engineering groups as well as political management, learned by a close relationship with MITI.
3. “corporate culture explosion” — I know of no other words to aptly describe the movement away from the culture of lifetime employment. I believe that ICOT coincided with greater forces within Japan causing this revolution; however, the revolution was certainly felt within the FGCS Project. Several, not a few, ICOT members switched their affiliations between companies and from companies to universities. If there was ever an industry to foster such a movement, it would certainly be a high-technology area such as computer engineering, therefore this should not come as a surprise. However, I think it did catch some of the companies by surprise.

4. PhD generation — ICOT generated an abnormally high (compared to comparable research organizations within Japan, which is a difficult comparison to make) percentage of PhDs from work conducted. Furthermore, it generated several professors, going to most of the major Japanese universities, as well as others.

Complete analysis of these four points could take many pages, so I will only give my (unsubstantiated) opinions here. I think all these natural results of the “market forces” acting on young ICOT researchers are positive. Increased communication between engineers, managers, professors and students will lead to more rapid progress in developing basic research ideas into successful commercial products. The question remains in my mind as to whether a National Project of this magnitude is necessary to create these human networks each generation, or if this first network will propagate itself without help from another project? It is reasonable to assume a mixed success, i.e., the networks will weaken with age, but will remain in place. Thus in the future, it may not require such a grand-scale project to strengthen ties. For example, current ICOT graduates, understanding the importance of free and flexible discussion of results at national conferences, will increase the participation of the researchers in their care, thus enabling the next generation to form their own friendships and working relationships.

### 3 About Technology

I will exploit this opportunity to discuss the validity or commercialization of the processing technologies developed by ICOT, specifically the idea of building a special-purpose multiprocessor to execute a fine-grain concurrent language. This seems to be the main concern of the press, and perhaps the key point upon which ICOT is being evaluated. One could criticize ICOT for attempting to naively leapfrog “fourth generation” RISC-based microprocessor technologies, which continue yearly to grow in performance. Ten years ago, Japanese companies did not have experience developing microprocessor architectures, much less second-generation RISC designs (superscalar), nor MIMD multiprocessor designs. Building the various PIM machines gave the hardware manufactures some limited experience in microprocessor design, although presumably this experience could have been had with a more conventional target.

On one level, however, the unique experience that *was* attained, i.e., that of fabricating tagged symbolic architectures, contains much of the structure needed to tackle the problem from the bottom-up, as being done by conventional multiprocessor vendors. It is not surprising that operating systems are now developing light-weight threads, and that languages such as object-oriented Smalltalk and tuple-based Linda form the cores of recent distributed processing efforts. My contention is that these efforts are climbing from the bottom-up, whereas ICOT had a top-down approach to the same problem (of massively parallel symbolic computation). Furthermore, because I know firsthand of the quick responsiveness of Japanese research and development in this area, I have little doubt that these two methods will be bridged. If the performance gap is bridged, the key question is *who will be in the better position?*

The top-down approach has advantage of programming and application experience in concurrent and symbolic, high-level languages. The bottom-up approach has the advantage of using imperative languages that evolved slowly, thus retaining market share. There is no clear answer to this question, but let me rephrase it in terms of two specific technologies: wormhole-routed distributed networks and concurrent constraint languages. I believe both required significant intellectual efforts to conceptualize, design, implement, and apply in real systems. The former represents a bottom-up technology and the latter a top-down technology. Bottom-up technologies are easier to introduce into designs, e.g., PIM/m incorporates wormhole routing (and can execute GDCC, a constraint language), whereas the Intel machines do not yet have implementations of constraint languages. Perhaps GDCC can be ported to general-purpose multiprocessors, but that is *not* the issue. Where GDCC came from, and where it is going, can only be determined from the foundation of the research expertise gained in its development. This is of course true about routing technologies, but again, bottom-up technologies are more easily imported (and more easily sold — they translate more directly to FLOPS).

## 4 Conclusions

I would like to finish this essay with a prescription for guaranteed success at generating the positive “social” results of the FGCS experience. Much discussion at the Evaluation Workshop concerned the issue of software. I think the distribution and availability of the great body of work is valuable, but not the main issue. The main issue to making the FGCS Project a “success” is to guarantee the high-level of computer science research initiated by ICOT. Shapiro suggested giving grants to foreign institutes for collaborative efforts. A better idea, in my view, would be to create a permanent Japanese Institute of Computer Science, which could accept international visitors, company trainees, and hire directly from the universities. There may be few proponents of this idea, primarily because of the game-theory effect that it appears to be in no individual’s best interest, i.e., not the universities (who have their own institutes), the national labs, or the companies. However, taken in total, it would in fact benefit Japan because a truly first-rate Institute, with an international reputation, would produce all the beneficial effects of ICOT, without the pressure of producing advanced technology products in limited time periods. The Institute would allow a unique opportunity for fresh university graduates to tackle advanced problems in computer science with the support of industrial technologies. For example, next-generation CPU and network designs are best produced with combined resources, allowing the companies to develop their own current-generation designs.

## References

- [1] P. Adamson and E. Tick. Greedy Partitioned Algorithms for the Shortest-Path Problem. *International Journal of Parallel Programming*, 20(4), August 1991.

- [2] P. Adamson and E. Tick. Parallel Algorithms for the Single-Source Shortest-Path Problem. In *International Conference on Parallel Processing*, Penn State, August 1992.
- [3] A. Goto, A. Matsumoto, and E. Tick. Design and Performance of a Coherent Cache for Parallel Logic Programming Architectures. In *International Symposium on Computer Architecture*, pages 25–33. Jerusalem, IEEE Computer Society, May 1989.
- [4] A. Imai and E. Tick. A Shared-Memory Multiprocessor Garbage Collector and its Evaluation for Committed-Choice Logic Programs. In *Symposium on Parallel and Distributed Processing*, pages 870–877. Dallas, IEEE Computer Society, December 1991.
- [5] A. Imai and E. Tick. Evaluation of Parallel Copying Garbage Collection on a Shared-Memory Multiprocessor. In *Proceedings Japanese Symposium of Parallel Processing*. Tokyo, May 1991.
- [6] A. Imai and E. Tick. Evaluation of Parallel Copying Garbage Collection on a Shared-Memory Multiprocessor. *IEEE Transactions on Parallel and Distributed Computing*, 1992.
- [7] E. Tick. Compile-Time Granularity Analysis of Parallel Logic Programming Languages. In *International Conference on Fifth Generation Computer Systems*, pages 994–1000, Tokyo, November 1988. ICOT.
- [8] E. Tick. Performance of Parallel Logic Programming Architectures. Technical Report TR-421, ICOT, Tokyo, September 1988.
- [9] E. Tick. A Performance Comparison of AND- and OR-Parallel Logic Programming Architectures. In *International Conference on Logic Programming*, pages 452–470. Lisbon, MIT Press, June 1989.
- [10] E. Tick. Comparing Two Parallel Logic-Programming Architectures. *IEEE Software*, 6(4), July 1989.
- [11] E. Tick. Parallel Logic Programming on Shared-Memory Multiprocessors: A Tale of N-Queens. In *Proceedings Japanese Symposium of Parallel Processing*. Atami, February 1989.
- [12] E. Tick. A Performance Comparison of Shared-Memory OR- and AND-Parallel Logic Programming Architectures for a Common Benchmark. *Journal of Information Processing*, 13(1), 1990.
- [13] E. Tick. Compile-Time Granularity Analysis of Parallel Logic Programming Languages. *New Generation Computing*, 7(2):325–337, January 1990.
- [14] E. Tick. *Parallel Logic Programming*. MIT Press, Cambridge MA, 1991.
- [15] E. Tick and J. A. Crammond. Comparison of Two Shared-Memory Emulators for Flat Committed-Choice Logic Programs. In *International Conference on Parallel Processing*, volume 2, pages 236–242, Penn State, August 1990.

- [16] E. Tick and N. Ichiyoshi. Programming Techniques for Efficiently Exploiting Parallelism in Logic Programming Languages. In *SIGPLAN Symposium on Principles and Practices of Parallel Programming*, pages 31–39, Seattle, March 1990. ACM Press.
- [17] E. Tick and K. Susaki. TOESP: A Prolog Benchmark. Technical Memo TM-451, ICOT, 1-4-28 Mita, Minato-ku Tokyo 108, Japan, January 1988.