Report of a stay at ICOT

Gérard Huet

April 22, 1988

Introduction

I visited ICOT and a few other Japanese institutions from April 4, 1988, to April 22, 1988. The report of my stay in Japan has two parts: a technical note "Experiments with GHC prototypes", and a non-technical general report, the present document. I first give a detailed diary of my activities.

1 A diary of my activities in Japon

1.1 April 4, 1988

I arrived at Narita around 11am, got a bus at 12, arrived at Tokyo Grand Hotel at 2pm, shaved and had a quick lunch. Takayama san came and picked me up at the hotel, and we arrived at ICOT around 3.

1.1.1 Administrative business

Met with Iwata san, who handled the administrative business of my stay. We briefly discussed ICOT organization, and I was introduced to the French guest researchers Mrs Philippe Devienne and Yvon Autret. I was given a desk on the "mountain side" of the lab, with a VT100 terminal. Dr Furukawa and Dr Fuchi dropped by to greet me.

1.1.2 Getting at work: connecting to computers

I get hrlp from Murakami san to set-up my computer environment. There are two SUNS 3-280, ICOT31, and ICOT32. I shall use ICOT31. The standard editor seems to be GNU emacs. TeX is not installed on the SUN ICOT31, only on ICOT32, and they do not use NFS, and thus I have to use the DEC20 for text processing. I use ftp to transfer files between the 2 machines.

1.2 April 5, 1988

1.2.1 Software installation

This morning, with the help of Murakami san, I was able to load my SUN tape. Successful installation of CAML V2-5 and of the Calculus of Constructions.

1.2.2 How to run CAML at ICOT

CAML is now installed on the SUN-3 ICOT31. If you have `/usr/local/bin` in your PATH, just type-in `caml`. Ask me for CAML documentation, which consists of the CAML Primer and the CAML Reference Manual. The system files are installed in `/usr/local/caml`.

1.2.3 CAML demonstrations

In the afternoon, demonstration to K. Sakai, Y. Takayama and colleagues of CAML features. I demonstrate in particular a purely applicative Knuth-Bendix completion prototype in CAML.

1.2.4 Welcome/farewell party

Most members of Lab 1, as well as several other researchers from other labs joined in a farewell party to Dr. J. Tanaka, who is returning to his company Fujitsu. The party was also for welcoming Mr. Ishizaka, who is joining ICOT's 1st laboratory from Fujitsu, and myself. Dr. Furukawa gave a nice speech, the atmosphere was very friendly and the traditional Japanese food excellent.

1.3 April 6, 1988

1.3.1 More software

In the morning, preparation of a CAML prolog system prototype.

1.3.2 Meeting with K. Mukai

Had an interesting lunch with K. Mukai, who is leaving to-morrow for Europe. He works on type theory and its applications to linguistics. He has interests in Barwise's situation semantics, Martin-Lof's intuitionistic theory of types, and Girard's linear logic. His work relates to the current research of Plotkin, Rounds, Ait-Kaci, and to the work on record types in polymorphic languages.

He has devised an extension of PROLOG with record structures, for linguistic purposes. His records are implemented as balanced binary trees, keeping the fields in sorted order. Record extension is done by tree merge. K. Makai says there are still algorithmic problems on how to implement best these structures, but the performance of the current implementation is acceptable.

1.3.3 Demos of METIS

S. Ohsuga made a demo of his equational system METIS. The system, written in PROLOG, runs on the DEC20. It implements various term-rewriting methods:

- Knuth-Bendix completion with rpo ordering, extensions lexico, multiset, and kachinuki
- interactive mode with menu selection of possible ordering choices
- inductionless induction with Fribourg's optimization
- unfailing completion of Hsiang and Rusinovitch

The system has a pleasant user interface with two windows, one for the current state of the theory, the other one for commands and messages. It is possible to save the state of completions. There is no associative-commutative treatment in this version. Examples shown were combinatory problems (synthesis of I from S and K, synthesis of fixpoint combinator from M and B), and a completion of groups without identity with unfailing completion. Unfailing completion is significantly slower than the ordinary completion, due to the great number of critical pairs generated. Proofs with inequalities are possible with this method.

The selection criterion for the next equation to process is a measure on $M = N$ as follows. Let m (resp.n) be the number of function symbols in M (resp. N), u (resp. v) the number of variables of M (resp. N). The ordering is on $< \{m, n\}, \{u, v\} >$. The selection strategy is not programmable. Critical pairs are reduced when produced. It is possible to undo the choice of ordering, but only one step.

1.3.4 CAL

K. Sakai gave me a demonstration of the CAL interpreter. This is a language that mixes logic programming with constraint satisfaction decision algorithms. He showed me examples with an implementation of Gröbner bases, which permits to solve non-linear equational constraints over C. Examples are geometry problems (e.g. using Pythagorus theorem to constraint lengths), maximisation of functions using Lagrange's method. The language looks elegant, and the implementation (in DEC20 Prolog) reasonably efficient. It is also possible to solve boolean constraints (using Boolean Gröbner basis), but at present one cannot mix both kinds of constraints. This work relates to the current research of Jaffar-Lassez, Colmerauer, Dincbas and Buchberger.

1.3.5 Meeting with K. Ueda

I met briefly K. Ueda in the evening. I discussed with him the relationship between category theory, lambda-calculus, and intuitionistic logic. We argued about various possibilities for merging the functional and logic programming paradigms. I demonstrated my little PROLOG prototype written in CAML. We discussed about the possibility of extending it to GHC.

1.4 April 7, 1988

This morning, the cherry trees are blooming.

1.4.1 Talk preparation

In the morning, I prepare my slides for next Monday's talk at the Logic Programming Conference 88.

1.4.2 PROLOG in CAML demo

I demonstrate to Takayama san my PROLOG interpreters in CAML. We discuss of constructive logic, and of the respective rigour of mathematics and computer science. My own point of view is that computer programming is more precise, and thus rigorous, than standard mathematics practice, but this is contrary to usual social conventions.

1.4.3 GHC

I am trying to understand the semantics of GHC, and I have trouble understanding what is a proper definition of deadlock. By discussing with K. Ueda I understand that current implementations of GHC do not allow user-defined predicates in the guards, which furthermore are evaluated sequentially left to right. I study K. Ueda's user's guide to his DEC20 GHC compiler and try a few examples.

1.5 April 8, 1988

Last night, it rained, and there is still snow in the street. The air is fairly cold. It looks like the Hanami will have to be canceled...

1.5.1 DHC

In the morning, I start working on a simplification of GHC, that I call DHC. The idea is to keep just the equality special predicate, and in this special case guards are un-necessary.

Later in the day, K. Sakai gives me a copy of his new paper on Boolean Gröbner bases.

1.6 April 11, 1988

Today, I continue working on a CAML implementation of DHC. Nice welcome lunch with Hrishige, Fuchi, Kurozumi, Furikawa, Hanegawa, Sakai and Takayama.

1.7 April 12, 1988

I the morning I give an invited lecture on "An overview of the Calculus of Constructions" at the Logic Programming Conference'88.

In the afternoon, I finish my prototype of DHC and test it on a few examples.

1.8 April 13, 1988

Today, I have a demonstration by Fujita and Sakai of the CAP-LA system, which runs on the PSI machine.

The system does proof checking and some theorem proving in linear algebra. The user interface goes through a structured editor. A syntax of proofs described in natural notation, called PDL, is provided. The system looks well designed, with enough generality to be applicable to other mathematical fields. I am shown a complete proof of the fact that the determinant of the transposed of a matrix is equal to the determinant of the matrix.

The PSI machine (now PSI2, built by Mitsubishi), is a personal workstation with 80Megabytes of memory, which is quite large by present standards. It runs the object-oriented logical formalism ESP and the operating system SIMPOS. The bit-map display manages windows with satisfactory performance.

1.9 April 14, 1988

Today, I work on extending my DHC interpreter to full GHC, with evaluable arithmetic predicates.

1.10 April 15, 1988

In the morning, I have a demo of the CIL system of natural language understanding. CIL interprets sentences of situation semantics, with some kind of call-by-need logical formalism which I did not really grasp. The system runs on the PSI machine.

In the afternoon, I visit the NTT labs in Musashino. I meet Dr. Katsuji Tsukamoto, Executive Manager of the Software Research Laboratory, who presents the various research activities. The researchers I meet are Shigeki Goto (Realizability interpretations), Yoshihito Toyama (term rewriting systems), Satoshi Ono (strictness analysis, programming environments), Mizuhito Ogawa (abstract interpretations of functional programs), and Junnosuke Yamada (inductionless induction).

I have a demonstration of the Ellis machine. This is the third generation of this LISP machine, now produced industrially by OKI Corp. It runs the "New Unified Environment" NUE which comprises the language TAO (combining LISP, PROLOG and object-oriented programming), the interactive editor Zen, and a window system. The current configuration is 16Mbytes, with large micro-programmable space of 64k of 64 bits words. The operating system does not seem to be completely reliable yet, and I was shown the demo of a LISP interpreter, the compiler does not seem to be operational yet. Nevertheless, the performances are quite impressive: 0.5 s for recursive fact(600) in LISP, 11kLISP estimated PROLOG, i.e. about 1.3 factor over Symbolics.

I also have a quick demo of NTT's data-flow machine, which is in a less advanced state (hand wiring).

Y. Toyama recently solved an important problem concerning the termination of left-linear term rewriting systems.

1.11 April 16, 1988

Hanami at Hotel Pacific in Shinagawa, with various colleagues of ICOT.

1.12 April 18, 1988

I finish debugging and testing my GHC interpreter.

I meet Kaoru Yoshida, from the 4th laboratory, who works on a parallel object-oriented language called A'UM. The current prototype implementation is in KL1. The language seems well suited to data-flow computation.

1.13 April 19, 1988

Departure for Sendai with Takayama san. We are greeted in Sendai by Pr. Sato, and we visit his lab after lunch. I meet his new research assistant Yukiyoshi Kameyama, who is interested in constructive mathematics. I install CAML without difficulty on their SUNS, and do a demo of the CAML environment.

Pr. Sato is now working on a new formalism called SST (Symbolic Set Theory), for which he has a realizability interpretation for generating programs in an untyped LISP-like language called sst. This work is still in progress, and no written documentation is available on either SST or sst.

1.14 April 20, 1988

In the morning, we meet with Pr. Ito, and we visit his laboratory. Tohoku University seems to be extremely well equipped for teaching, with spacious computer labs (80 terminals/workstations). The research equipment comprises a Symbolics, 2 Ellis machines, 1 SUN/Toshiba (which runs an unorthodox version of the operating system), many SONY workstations (this looks like an attractive workstation running Berkeley Unix, cheaper and slower than a SUN-3, but with 8 Mbytes of memory).

I am shown a demo of a parallel LISP machine. The machine connects 8 68000 procesors to a common memory with a cross-bar switch. An extra 68000 is used for overall control. The machine runs a version of LISP with explicit parallelism primitives.

In the afternoon, I give a talk on the realizability method of Christine Paulin-Mohring for the Calculus of Constructions. Then Y. Takayama presents a talk on his proof extractor for QJ (Pr. Sato's previous logic), and S. Goto (who arrived from NTT) presents his non-standard realizability interpretation (which gives a notion a proof extraction well-suited to call-by-need computations on potentially infinite streams).

1.15 April 21, 1988

More discussions in a private seminar between Prs Ito and Sato, S. Goto, Y. Takayama, and myself, on constructive mathematics, onsen, program extraction from proofs, hoya, and other aspects of computer science and Japanese "art de vivre".

Return to Tokyo in the afternoon. I write my trip report in the evening.

1.16 April 22, 1988

This is my last day here. In the morning, I give a last demo showing CAML and my implementations of GHC to S. Goto and others.

Conclusion

ICOT is a very good research laboratory, with many first-class researchers who work on important problems. The lab has already produced an impressive amount of languages, provers, machine architectures.

ICOT has a nice work atmosphere. This was my first experience of sharing such a large office, and this proved not to be too hard to adjust to this environment, which certainly promotes sharing and discussions. However, the noise may be distracting at times.

It is obvious that a 2 weeks visit is barely enough to meet people, understand the problems, and maybe start some collaborative work. I regretted not to have more time to discuss with Murakami san of proving GHC programs. I also had no time to meet the 4th laboratory researchers.

Curriculum Vitae

1 Personal Data

Name: Huet Gérard.

Birth Date: July 7th 1947

Nationality: French

Family status. Married, 2 children

Personal Address:
7, rue de Naples.
78150 Rocquencourt
Phone: (1) 39 55 39 96

Professional Address:
INRIA
Domaine de Voluceau, Rocquencourt B.P. 105

78153 Le Chesnay Cedex
Phone: (1) 39 63 54 60
UUCP: mcvax!inria!huet

2 Studies

June 64	Baccalauréat Mathématiques Elémentaires
Sept 64 - June 66	Mathématiques Supérieures & Mathématiques Spéciales Lycée St Louis, Paris
Sept 66 - June 69	Ecole Nationale Supérieure de l'Aéronautique, Paris.
June 69	Diplôme d'Ingénieur Civil de l'Aéronautique.
June 69	Maitrise es Sciences, Informatique, Université de Paris.
Sept 69 - Sept 72	Case Western Reserve University, Cleveland, Ohio
Jan 71	M.S. Computing and Information Sciences Dept, CWRU
Aug 72	Ph. D. CWRU
Sept 76	Docteur d'Etat es Mathématiques, Université Paris VII.

3 Professional Experience

3.1 Job Career

June 70-Aug. 72	Research Assistant, Computing and Information Sciences, CWRU, Cleveland Ohio
Oct 72-March 73	Researcher, IRIA-Laboria, Rocquencourt, France.
Since April 73	Research Engineer, IRIA-Laboria, Rocquencourt, France.
Jan. 74-Aug. 79	Project Leader, Projet Conception et Réalisation d'Outils d'Aide à la Programmation, IRIA.
Sept. 79-July 80	Computer Scientist, Computer Science Laboratory, SRI International, Menlo Park, Ca.
Aug. 80-Aug. 85	Project Leader, Projet Formel, INRIA, Rocquencourt, France.
Sept. 85-May 86	Invited Professor, Computer Science Department, Carnegie-Mellon University.
June 86	Consultant, Digital Corporation, Systems Research Center, Palo Alto, California.
June 86-Now	Directeur de Recherches de 1ère catégorie. Inria. Project Leader, Projet Formel.

3.2 Language Experience

French native, English fluent spoken and written.

3.3 Programming experience

Familiar with operating systems Dec10, Tenex, Multics, Unix, Symbolics, ... Experienced Programmer in Fortran, Pascal, AlgolW, Lisp, ML... Project leader experience in software development for interactive programming environments, symbolic computation and automated deduction.

3.4 Teaching experience

Various graduate courses taught, at Paris VII from 1974 to 1976, at Paris XI

(Orsay) from 1976 to 1978, at Paris VII from 1982 to now. Undergraduate courses in computer science taught at Paris XI from 1976 to 1978. Various specialized courses in summer schools Professor, Ecole d'Intelligence Artificielle, Vigneu, France, July 1985. Graduate course on Formal Structures for Computation and Deduction, Carnegie-Mellon University, 1986. Professor, International Summer School on Logic of Programming and Calculi of Discrete design, Marktoberdorf, Germany, August 1986.

3.4 Editorial responsabilities

Member of the Editorial Board, New Generation Computing Journal (Ohmsha & Springer-Verlag). Advisory Editor, Studies in Proof Theory Lecture Notes, Bibliopolis. Editor, Research Notes in Theoretical Computer Science (Pitman).

4 Research Activity

My long-range research interest is automated deduction, and its application to the development of reliable software. I made fundamental contributions to the field of automated deduction in the areas of higher-order theorem proving, unification theory and mechanization of equational logic by canonical simplification methods. More recently, I am developing proof methods for a higher order typed lambda calculus well suited for the development of constructive mathematics.

On the practical side, I have been working for 10 years in various projects of programming environments development based on symbolic computation tools. The Mentor programming environment (project started jointly with G. Kahn in 1974) provides users with a uniform set of tools centered on a structured editor for abstract syntax trees. The KB system (designed and implemented jointly with J.M. Hullot) manipulates algebraic specifications by various equational proof methods centered around canonical simplication techniques. The Semant system (designed and implemented jointly with G. Cousineau) manipulates formal definitions of programming languages given in the denotational semantics style. This last system uses as its meta-language an improved version of the ML language developed originally for the LCF proof assistant by R. Milner and its colleagues at the University of Edinburgh. This extended ML was developed in collaboration with L. Paulson from University of Cambridge. A new proof assistant for the Theory of Constructions is under development in ML.

The theoretical and the practical interests are now in agreement. It is clear that nothing short of very powerful formalisms for constructive mathematics is adequate for the development of modular programming languages and of programming environments suited for the development of verifiably reliable software. The Theory of Constructions, developed jointly with Th. Coquand, gives a uniform formalism in which to develop higher order reasoning. It basically augments de Bruijn's Automath typed lambda-calculi with the powerful second-order types of Girard. The resulting language, in which Intuitionistic Type Theory can be developed in a consistent manner, is the ideal linguistic medium for Natural Deduction. For Constructive Mathematics, it gives hope to realize the programme set forth by Leibniz, Frege, Brouwer, Heyting and Bishop. For Computer Science, it clears the way for the harmonious development of programs decorated with their specifications, in a powerful higher order functional language in which types are the propositions of the underlying logic.