REPORT ON VISIT TO ICOT

Robert Kowalski
Department of Computing,
Imperial College,
London, UK

July 1990

I visited ICOT during the week of 2-7 July 1990. I gave four talks: one entitled "English as a (Logic) Programming Language" on 2 July; talks on abduction and on rules and exceptions in logic programming on 6 July; and a talk on amalgamating object level and metalevel in logic programming on 7 July. In addition I arranged to give a talk on 8 July to the ICOT supported working group on legal reasoning.

I had discussions on non-monotonic reasoning with Jun Arima, Katsumi Inoue, and Ken Satoh. Our discussions centred around the motivation for developing non-monotonic reasoning for full predicate calculus. Having worked for many years developing theorem-provers for full predicate calculus, and having failed to find significant applications, I suggested that it might be sufficient to extend logic programs, which do have practical applications, rather than to develop general theories for the full predicate calculus.

Ken Satoh showed me a representation of a (soft) constraint satisfaction problem formulated using disjunction in the full predicate calculus. We

also considered a formulation of the same problem as a constraint logic program. We seemed to agree that it would be very interesting to try to establish some kind of general correspondence between alternative representations, one using disjunctions and the other using Horn clauses. At a first approximation it seemed that one formulation uses disjunction to reason explicitly with the completion, the other reasons with Horn clauses where the completion is implicit.

We also discussed several other possible uses for non Horn logic. I suggested in particular that we need non Horn logic for integrity constraints or to reason about incomplete deductive databases.

Although many researchers throughout the world are working on full predicate logic, few researchers are applying it. Most applications can be formulated in the much simpler logic programming form (or in natural extensions of logic programming form, including integrity constraints or "real" negation, for example). Donald Loveland recently confessed to me that he was having much difficulty finding applications for his near Horn logic programming system. It has occurred to me that this might be related to the fact that in intuitionistic logic a disjunction A or B is provable if and only if A is provable or B is provable. Perhaps we do not need non Horn logic for the same reason that intuitionistic logic might be adequate and natural for knowledge representation.

I had a related discussion with Masayuki Fujita, who showed me a theorem-prover implemented in the Edinburgh logical framework. The theorem-prover uses disjunctive reasoning to synthesise logic programs as extract terms from proofs of specifications. It is also possible, however, to

derive logic programs from specifications using "symbolic execution" (fold-unfold techniques), without reasoning explicitly with disjunction. I believe it would be of practical value and of theoretical interest to understand better the relationship between techniques which use full predicate calculus and techniques which use extensions of logic programming form.

This theme reoccurred in discussion with Professor Yoshino and Professor Haraguchi concerning the formalisation of statute law. Many legal scholars have advocated the logical analysis and representation of statute law. Until very recently, full propositional logic or predicate calculus have been the preferred formalisms for such representation. I believe it is possible to show, by analysis of many examples, that simple extensions of logic programming form are adequate and appropriate for representing statute law. In fact (and this is the most important point) restricting expression to (extended) logic programming form actually simplifies and clarifies the law.

Professor Yoshino proposed taking the example of the United Nations convention on the international sale of goods as a common example for international research and collaboration in the field of building expert systems for legal reasoning. I strongly support this suggestion and personally intend to use this example in my own work. I suggested that it might be possible to obtain United Nations support (from UNESCO, for example) to hold a workshop to discuss and compare alternative formalisations of the convention.

Professor Yoshino and Professor Haraguchi are also interested in

implementing case based reasoning in legal domains. Professor Haraguchi argued that this is necessary in legal reasoning not only to generate analogies between old cases and new ones, but also to generate analogies between only rules and new cases.

I was shown a demonstration of a case-based legal reasoning system running in KL1 on the multi-PSI machine by Masaki Hoshida and Masato Ishikawa. The system was described as a production rule system; however, it soon became apparent that the production system in this case implements forward reasoning using rules generated from the arguments used in previous cases. This approach to case-based reasoning is similar to work by Uri Schild, previously a Ph.D. student at Imperial College, and is similar to the approach taken in the UNDP sponsored KBCS project in New Delhi.

I was interested in the description of how the case-based reasoning system was executed on the multi-PSI machine. I was particularly interested in the way the rules derived from different cases were distributed among the different processors. This suggested to me the idea that the multi-PSI machine might be used as a kind of database machine. In the example of case-based reasoning, in particular, it seems especially interesting to consider the trade-off between holding copies of the same rules and data in all the processors compared with partitioning them among the different processors. It might be useful, for example, to hold isa-hierarchy rules in all the processors, while distributing the rules derived from previous cases among the different processors.

Because the case-based reasoning system is implemented in KL1, I was interested to obtain information about KL1. I was pleased to discover from

Kazunori Ueda that KL1 is essentially GHC with annotations to control parallel execution. This should make KL1 much easier to use than if it were less closely related to GHC. It seems to me that it might also be useful to describe with annotations the way in which the initial program and/or database is distributed among the processors.

Kazumasa Yokota explained to me his work on deductive and object-oriented databases. His language is very elegant and powerful. It seems to compare very well with some related work in the ESPRIT Basic Research Action "Computational Logic", for which I am coordinator. This related work by our partners in Kaiserslautern and Rome is on structured typed languages, which have arisen from the KLONE language. Other work with similar objectives, done at ECRC, using linear logic for its semantics, was reported at ICLP 90 in Jerusalem.

Katsumi Nitta gave me an overview of the work of the Seventh Research Laboratory. We also discussed in detail his previous work on representing procedures in patent law. I found his explanation of the behaviour of his object-oriented, logic-like language very attractive, but I was not happy with its lack of pure logical semantics. It would be very interesting and very useful to find a purely logical reconstruction of his system. I liked his treatment of deontic modalities (permission, prohibition, obligation) as indicating the legality of events. I have recently begun to think about representing prohibitions and obligations as integrity constraints. It seems to me that such an approach might provide a "rational reconstruction" of Nitta's work. I hope to have time to explore this idea further.

I was pleased to meet with Koichi Furukawa during his busy schedule. He gave me an overview of the work at ICOT and an indication of some of the interconnections between different areas. I was very interested in the biological applications being discussed with the National Institute of Health. I agree with Dr. Furukawa that this is an area of application where the combination of symbolic computation, learning and large-scale parallelism can make an important contribution.

I have visited ICOT before, for very short visits, on three previous occasions, the last being two and a half years ago. I was greatly impressed by the range and depth of work now being carried out. Within a relatively short period of approximately eight years, ICOT has developed into a research laboratory which rivals the best Computer Science laboratories in the world. It seems inconceivable to me that all this might come to an end in two years time. I hope for the sake of Computer Science in Japan that some way can be found to maintain the momentum which has been created.

I was extremely well cared for during my visit. Jun Arima made sure that my visit was as pleasant and well organised as possible. Dr. Kazuhide Iwata arranged all the administrative aspects of my visit most helpfully and efficiently. Ken Satoh showed me the attractions of Kamakura. Mr. Hiroichi Hiroshige, Dr. Kazuhiro Fuchi, and Dr. Koichi Furukawa showed me great friendship and hospitality.

ROBERT KOWALSKI

Department of Computing
Imperial College of Science and Technology
London

Bob Kowalski studied at the University of Bridgeport (B.A. Mathematics 1963). Stanford University (M.Sc. Mathematics 1966) and the University of Edinburgh (Ph.D. Computer Science 1970).

From 1967 to 1975 he was a Research Fellow in the Department of Computational Logic at the University of Edinburgh. He came to Imperial College in 1975 as Reader in Theory of Computing. In October 1982 he was appointed Professor of Computational Logic.

His early research was in the field of automated deduction. This led in 1972 to the development with Alain Colmerauer at the University of Aix-Marseille of the use of formal logic as a programming language. Kowalski's research provided the theoretical framework, while Colmerauer's gave rise to the programming language PROLOG.

Kowalski is now Head of the Logic Programming Group at Imperial College. The Group is investigating the implementation and application of both PROLOG and a parallel logic programming language, PARLOG. The principle applications are in the areas of expert systems, deductive databases, and the formalization of legislation.