GUEST EDITOR'S INTRODUCTION
Special Issue: Applications of Logic Programming

L. STERLING

In the guest editor’s introduction to the previous applications special issue of the Journal of Logic Programming (Volume 8, Numbers 1 & 2, 1990), the question was posed “Are applications being written using logic programming tools?” Seven years later, the answer is a clear and unequivocal yes. There is a steadily growing literature of solid applications written in logic programming languages, published in the proceedings of the three conferences on Practical Applications of Prolog, for example, and in the Prolog 1000 database.

This special issue of the Journal of Logic Programming is intended as a snapshot of that growing literature. The papers included here are revised versions of papers presented at the Second International Conference on Practical Applications of Prolog (PAP’94) held in London in April, 1994. For the conference, 44 papers were accepted from over 90 submitted. Of those, 18 were considered for the special issue, and 7 were selected, coincidentally from 7 different countries. Many of the authors of the papers are newcomers to the logic programming community or outsiders—evidence of the evolution of the Prolog user community and the broad applicability of logic programming. Although several of the papers describe prototypes, there also were many examples of systems successfully deployed in industry. Let me give a brief overview of the accepted papers.

The first paper in the special issue is Computer Support for Protocol-Based Treatment of Cancer by Peter Hammond and Marek Sergot from the United Kingdom. They describe OaSiS, a decision support system written in MacProlog. The work represents an evolution of the use of Prolog to develop expert systems, both in the scope of the system and the attempt to incorporate deontic logic.

The second paper is The NRL Protocol Analyzer: An Overview by Catherine Meadows from the U.S. Naval Research Labs. She describes a prototype tool to analyze the security of cryptographic protocols based on equational unification. It has been implemented both in Quintus Prolog and SWI Prolog, and took advan-
tage of Prolog's built-in unification and backtracking, and facility for rapid prototyping.

Prolog for Structured Character Description and Font Design by Martin Dürst is the next paper. Its application domain is a little different—analyzing structural character descriptions for East Asian ideograms. The system has been written in Quintus Prolog and ECLiPSe Prolog.

The next two papers describe tools built on top of Prolog. A Tool for Developing Interactive Configuration Applications by Tomas Axling and Seif Haridi at SICS, Sweden, describes OBELICS, a tool implemented in SICStus Objects, an object package on top of SICStus Prolog, that integrates a problem-solving method for interactive configuration by selection and domain knowledge modeling. SICStus Objects also has been used in the development of IDEA, an intelligent data retrieval system. The environment is described in Intelligent Data Retrieval in Prolog: An Illuminating Idea by Cristina Ruggieri, Mirko Snacanassi, Nenna Dore, Francesco Russo, and Ugo Manfredi from DS Logics, Italy.

A mainstay application area for Prolog has been hardware design. The paper of Ulrich Bieker and Andreas Neumann entitled Using Logic Programming and Coroutining for Electronic CAD describes an elegant simulator of a very high level hardware description language. Parts of the system could benefit also from constraint technology.

The final paper addresses constraints directly. The Practical Applications of Prolog (PAP) conferences also have accepted constraint logic programming papers. Constraint Logic Programming for Examination Timetabling by P. Boizumault, Y. Delon, and L. Peridy from France describes an application for timetabling exams using constraints.

I cannot resist speculating on the future of applications, within logic programming, in general, and Prolog, in particular. There is good news and bad news. The good news is that the existence of successful applications based on logic programming is now well established. Indeed I do not foresee a need for any other special issues of the journal on applications. Of course, individual application papers can be submitted at any time to be considered for publication in JLP. There is finally a standard for Prolog and Prolog has entered the mainstream of languages.

Unfortunately, along with entering the mainstream comes the need to face the realities of life: It will be a struggle to survive. The future of Prolog for practical applications is by no means ensured. There is concern about the long-term viability of the Prolog vendors and a question as to who will maintain and support the various Prolog implementations. It should be an interesting next five years.

Thanks to the Editor-in-Chief, Maurice Bruynooghe, for supporting a special issue on applications. Thanks to all the reviewers of the papers and to the reviewers for PAP'94. Thanks also to the Department of Computer Engineering and Science at Case Western Reserve University for its support.