Introduction to the Special Section

Bal, H.; Belkhouche, B.; Soffa, M.L.

published in
IEEE Transactions on Software Engineering
1995

document version
Publisher's PDF, also known as Version of record

Introduction to the Special Section

Henri E. Bal, Boumediene Belkhouche, and Mary Lou Soffa

I. Introduction

If we consider Zuse's Plankalkül the first programming language, research on computer languages is about half a century old now. During this period, languages have always been in the mainstream of computer science research. The focus of the research has perhaps shifted somewhat from efficiency (1950s) to expressivity (1960s), portability (1970s), and handling complexity (1980s), but the interest in languages never declined. At present, language designers and implementors are faced with new challenges, for example, exploiting novel (parallel and distributed) architectures and supporting new programming methodologies like object-oriented programming.

The IEEE Computer Society International Conference on Computer Languages (ICCL) brings together people working on many different aspects of programming languages, specification languages, and other kinds of computer languages. ICCL has a broad scope covering theoretical and practical work, general-purpose languages and application-specific languages, and language design as well as implementation. ICCL '94 is the fifth of the series of bi-annual conferences sponsored by the IEEE Computer Society Technical Committee on Computer Languages (TCCL). As in the previous meetings, ICCL'94 strived to further its commitment to quality and diversity. The broad scope and the international character of ICCL are confirmed once again by the diverse submissions from all parts of the globe.

One hundred and thirteen papers were submitted to ICCL'94, from 24 different countries. Twenty five papers were selected for presentation at the conference. In addition, the program contained two invited talks and two panel sessions. In the first invited talk, Robert Dewar addressed language evaluation, and in the second invited talk, Patrick Cousot addressed higher order abstraction.

The papers in this issue cover the best software engineering-related work presented at ICCL'94. The initial selection was done by the ICCL'94 program committee and conference chairman, in cooperation with TSE. Each selected ICCL'94 paper was improved substantially by the authors and then resubmitted to TSE. The papers were again formally refereed. The four accepted papers deal with issues ranging from language design and usage to implementation-oriented aspects.

The first paper, "Reflections on Metaprogramming," evaluates a reflective programming technique that some object-oriented languages use to enhance flexibility. With this technique, called metaprogramming, part of the semantics of a language is represented in objects. The implementation of these objects can be modified by the programmer, so the semantics can be tailored to the needs of an application. The Common Lisp Object System (CLOS) supports this technique through its metaobject protocol. The authors have used this protocol to extend CLOS with support for persistence, which was important for implementing a CAD system. The paper describes their experience with metaprogramming in CLOS.

The second paper, "A*: A Language for Implementing Language Processors," describes a new language for creating language processing tools. A* is as easy to use as Awk, but it generates tools for analyzing programs written in a variety of source languages. A* programs operate on parse trees, whereas Awk programs operate on record-based input streams. A* has been used for generating various tools for languages such as C and LOTOS. The paper describes the motivation, design, and usage of A*.

The third paper is "SPiCE: A System for Translating Smalltalk Programs Into a C Environment." The authors note that Smalltalk-80 is a very productive language for prototyping, but that delivering applications in this language is hard for two reasons: Smalltalk programs cannot run in isolation from the Smalltalk environment, and they cannot interoperate with code written in other languages. These problems are solved through a system called SPiCE, which translates Smalltalk into portable and interoperable C code. Since Smalltalk and C have widely different execution and storage models, the translation is difficult. The SPiCE system is able to achieve its goal, however, and has been used successfully for compiling several large Smalltalk applications.

The fourth and last paper, "Region Analysis: A Parallel Elimination Method for Data Flow Analysis," discusses how to effectively parallelize the process of data flow analysis. A key idea is to partition the flow graph in such a way that a sufficient number of parallel tasks is created that can be balanced evenly among the processors. The technique used to do the partitioning is called region analysis. This technique is compared with parallel Allen-Cocke interval analysis and is shown (through measurements on a shared-memory multiprocessor) to result in better performance.

We would like to thank everybody who has been involved with ICCL. Specifically, we would like to thank the program committee and the numerous referees for ICCL'94, the Computer Society staff, and Jean-Paul Bahsoun. We are grateful to Nancy Leveson for providing us with the opportunity to edit

Manuscript received September 1993; revised March 1995.

H.E. Bal is with Vrije Universiteit of Amsterdam; e-mail: bal@cs.vu.nl.

B. Belkhouche is with the Department of Computer Science, 301 Stanley Thomas Hall, Tulane University, New Orleans, LA 70118-5674; e-mail: bb@cs.tulane.edu.

M.L. Soffa is with the Faculty of Arts and Sciences, 910 Cathedral of Learning, University of Pittsburgh, Pittsburgh, PA 15260; e-mail: soffa@cs.pitt.edu.

IEEECS Log Number S95049.

this special issue. We thank Lee Damsky, who did a wonderful job in coordinating the efforts to make this special issue a success.

Henri E. Bal received an MSc in mathematics from the Delft University of Technology in 1982 and a PhD in computer science from the Vrije Universiteit (Free University), Amsterdam, in 1989. His research interests include programming languages, parallel and distributed programming, and compilers. At present, Dr. Bal is senior lecturer (equivalent to associate professor in the United States) at the Vrije Universiteit of Amsterdam. He is head of a research group on parallel programming. The research of this group is based on the Orca parallel language, which has been implemented on Amoeba, Unix, and several multicomputers. Dr. Bal is author of *Programming Distributed Systems* (Prentice Hall, 1991) and coauthor (with Dick Grune) of *Programming Language Essentials* (Addison-Wesley, 1994). He was program chairman of the IEEE Computer Society 1994 International Conference on Computer Languages.

Boumediene Belkhouche is currently an associate professor of computer science at Tulane University. His research interests include programming languages, formal semantics of concurrency, and software engineering.

Mary Lou Soffa received her PhD in computer science from the University of Pittsburgh in 1977. Since that time, she has been a faculty member at the University of Pittsburgh and is currently a professor in the Computer Science Department. Since 1991, she has also been serving as the dean of graduate studies in arts and sciences at the University of Pittsburgh. Her research interests include language implementation, parallelizing compilers, program analysis, and software tools. She currently serves on the editorial boards of *IEEE Transactions on Software Engineering*, *ACM Transactions on Programming Languages and Systems*, *International Journal of Parallel Programming*, and *Computer Languages*. She also serves as member-at-large for SIGSOFT and vice chair for conferences for SIGPLAN.