ELSEVIER

Future Generation Computer Systems 16 (2000) v–vi

FGCS
Future Generation Computer Systems

Guest Editorial

Principles of abstract machines

An abstract machine is a device used mainly by compiler writers and computer architects to bridge the semantic gap between a high level (programming) language and a low level (computer) architecture. Two widely known examples are the Java virtual machine (for Java) and the Warren abstract machine (for Prolog). Whether the devices are called abstract machines or virtual machines appears to be a matter of taste and background.

Abstract machines are needed because the differences are so great between the features offered by a high level language (such as the creation of an object in Java, or the unification of two terms in Prolog) and the facilities provided by a computer (such as adding the contents of two registers). Using an abstract machine brings the two levels together by interposing one or more intermediate levels. An abstract machine both hides details of the level below and exposes details of the level above.

The challenge for research in abstract machines is the development of scientific methods for deciding at what level an abstract machine is needed and what it should hide and expose.

On the theoretical side, significant progress has been made recently using the technique of formal refinement. This enables one to systematically derive a more detailed version of an abstract machine from a less detailed one, whilst preserving correctness. This effectively pushes the machine down in the hierarchy.

On the practical side there are still major difficulties. In particular it is difficult to control or predict the efficiency of a translation from the high level language to the low level machine.

This issue describes the state-of-the-art in the theory and the practice of abstract machines. It contains an overview paper and a selection of six papers presented at the first international workshop on *Principles of Abstract Machines* (WPAM). The workshop was held in Pisa, Italy, on 19 September 1998 in conjunction with the SAS and PLILP/ALP conferences.

The overview paper by Diehl et al. presents a detailed annotated bibliography of research on abstract machines.

The first regular paper describes how abstract machines can be made less abstract by refinement. Cabestre et al. start from the operational semantics of a simple language (for evaluating arithmetic expressions) and derive a compiler and an abstract machine for the language.

Abstract machines are a general tool. The next two papers show how abstract machines are deployed in areas other than code generation. Brown and Manandhar show how choosing appropriate data structures helps to improve the performance of an abstract machine designed for parsing natural languages. Walton et al. show how abstract machines can be designed to allow for run-time module replacement (in the context of the functional language Dynamic ML).

Efficiency is the main driver in the following paper. Demoen and Sagonas describe an abstract machine to support tabling in an implementation of Prolog. Tabling causes previously evaluated goals to be remembered for subsequent use.

PII: S0167-739X(99)00087-4

Pragmatic issues are discussed by the authors of the last two papers. Diehl describes visualisation tools to help abstract machine designers animate the behaviour of their machines. Bertelsen provides a careful analysis of the Java virtual machine showing how modelling existing designs helps to expose problems in a design.

On behalf of the WPAM programme committee we thank all those who submitted papers. We thank Michael Hanus, Michael Franz and Peter van Roy for their careful work in the reviewing and selection process.

S. Diehl[a], P.H. Hartel[b], P. Sestoft[c]
[a]*Universität des Saarlandes, Postfach 15 11 50*
D-66041 Saarbrücken, Germany
E-mail address: diehl@cs.uni-sb.de (S. Diehl)
[b]*Department of Electronics and Computer Science*
University of Southampton
Southampton SO17 1BJ, UK
[c]*The Royal Veterinary and Agricultural University*
Department of Mathematics and Physics
Thorvaldsensvej 40, 5 sal
DK-1871 Frederiksberg C, Denmark