DYNAMICS AND PRAGMATICS FOR HIGH PERFORMANCE CONCURRENCY

the abstract of a thesis submitted to
The University of Kent at Canterbury
in the subject of computer science
for the degree
of doctor of philosophy

By
Frederick R. M. Barnes
June 2003

This thesis is concerned with support at all levels for building highly concurrent and dynamic parallel processing systems. The CSP model of concurrency, as (largely) embodied in the occam programming language is used due to its simplicity, expressiveness, architecture-independent nature, and potential for high performance. Additionally, occam provides guarantees regarding freedom from aliasing and race-hazard error.

This thesis addresses one of the grand challenges of present day computer science: providing a software technology that offers the dynamic flexibility and performance of mainstream object oriented environments with the level of safety, formal analysis, modularity and lightweight concurrency offered by CSP/occam. Two approaches to this challenge are possible: do something to make the mainstream languages (e.g. Java, C++) safe, or make occam dynamic — without compromising its existing good properties. This thesis follows the latter route.

The first part of this thesis concentrates on enhancing the occam language and run-time system, on a commodity platform (IBM PC) running the freely available Linux operating system. After a brief introduction to the various components of the KRoC occam system, additions and extensions to the occam programming language and supporting run-time system are examined. These provide a greater degree of programming flexibility in occam (for example, by adding support for dynamic allocation, mobile semantics and dynamic network construction), without compromising the safety of programs which use them. Benchmarks are reported that demonstrate significant improvements in performance (for example, channel communication in tens of nano-seconds).

The second part concentrates on improving the level of interaction between occam programs and the OS environment. Providing easy access to sockets and networking, for example.

This thesis concludes with a discussion of the work presented herein, with consideration given to parallels with object-oriented languages. Also described are details of ongoing and potential future research. The modified language grammar, details of new compiler generated code, and miscellany are provided in the appendices.