THE MONADS PROJECT STAGE 2: HARDWARE DESIGNED TO SUPPORT SOFTWARE ENGINEERING TECHNIQUES§

J Leslie Keedy, David Abramson, John Rosenberg & David M Rowe

Department of Computer Science
Monash University
CLAYTON VIC 3168

The paper first describes the main aim of the MONADS project, which is to develop and investigate techniques for improving the quality of software engineering for large systems, and the main technique under investigation, the information-hiding principle. Then follows a brief overview of current hardware and software research to support information-hiding systems. The purpose of the paper is to introduce and provide background information relevant to five companion papers.

§ See also other papers by these authors individually.

1. INTRODUCTION

At the Eight Australian Computer Conference in Canberra in 1978 the MONADS team presented a series of papers describing various aspects of the first MONADS Operating System (Keedy, 1978; Rosenberg and Keedy, 1978; Ramamohanarao and Keedy, 1978; Georgiades, Richards and Keedy, 1978; Richards and Keedy, 1978). This was an experimental operating system designed to control a Hewlett Packard HP2100A computer system which had been modified at Monash University to support additional addressing modes, registers and a virtual memory (Wallace, 1978). Since that time the MONADS project team has expanded quite considerably and much new work has been done, with the emphasis changing such that a substantial amount of effort is currently being devoted to the design of processor architecture and hardware, as well as supporting software. At this conference we shall be presenting a further series of papers describing some of the advances which have been made, particularly in the area of hardware. The purpose of this present paper is to introduce the more detailed technical papers and to place them in the wider context of the MONADS project (Keedy, 1981).

2. THE BASIC AIM OF THE MONADS PROJECT

Despite the wide range of its detailed activities (e.g. operating systems, compilers, language design, computer architecture, hardware design), the MONADS project has a single overriding basic aim: to investigate and develop techniques and methods for improving the specification, design and implementation of large software systems. The need for improvements in software engineering principles and practice is widely acknowledged and the inferior quality of most existing software systems of all types (e.g. application systems, operating systems, real-time systems, database systems) is freely admitted (Naur, Randell and Buxton, 1976). Some of the symptoms of the software problem are high development costs, long delivery delays, delivered systems which are riddled with errors and exorbitant maintenance costs. The MONADS team, in common with other researchers, considers that the fundamental reason for this state of affairs is that at present software systems are not well structured. By following the example of other engineering disciplines and even of natural systems, we believe that systems should be composed of stable and relatively independent components which have minimal interactions with each other (Keedy and Rosenberg, 1981; Rosenberg and Keedy, 1981a). This can be achieved in practice by rigorously applying the information-hiding principle.

3. THE INFORMATION-HIDING PRINCIPLE

The basic idea behind the information-hiding principle is that systems should be decomposed into modules in such a way that the designers and implementors of individual modules need as little information as possible about the implementation details of other modules (Parnas, 1971). Modules are therefore specified in terms of procedural interfaces (with one or more entrypoint procedures per module), since this allows information about algorithms, data structure representations and the use of other modules to be entirely hidden from a calling module (Parnas, 1972). Although this sounds straightforward it leads to a quite different view of modularity from the approaches in common use. If rigorously applied to all software units the use of free-standing data structures is outlawed, and so a quite different view of files emerges, for example (Keedy and Richards, 1982). In a companion paper (Keedy, 1982) one of us presents an extended example showing how a set of general purpose and special purpose modules can be made to cooperate in practice. The example was chosen because it provides illustrations of modules which in conventional systems would be managed in a variety of different ways: user programs, program modules, subroutine libraries, files, data abstractions, operating system modules and even hardware modules. It becomes evident from a study of this example that the adoption of this approach to system design, despite its basic simplicity, cannot straightforwardly be used in currently available computing environments. The problem is that existing operating systems are not designed to provide the basic support necessary and that existing computer designs cannot support either the modular approach or the required operating systems in an efficient or elegant way.

4. CURRENT HARDWARE RESEARCH

The MONADS team's current ideas about how information-hiding systems should be designed have been strongly influenced by the work undertaken in building an operating system in stage one of the project. It was only in the course of that research that we realised, for example, that files could be integrated into a uniform information-hiding systems model. We eventually recognised that a major consequence of this and of other ideas e.g. a uniform approach to process structuring (Ramamohanarao, 1980; Keedy and Ramamohanarao, 1979) was that we would need new hardware and a new operating system in order to demonstrate the full potential of information-hiding systems.

Given, on the one hand, a lack of funds, and, on the other hand, experience in modifying a Hewlett Packard HP2100A for use with the MONADS I operating system (Wallace, 1978), the project team accepted Abramson's suggestion in early 1979 that as part of his Ph.D. research he should undertake far more extensive modifications to a second HP2100A. It was realised from the outset that there were some basic restrictions in the HP2100 which could not be removed by the planned modifications, and for this reason Abramson's new hardware, which is known as MONADS II,was regarded as a pilot project which would be capable of testing the feasibility of our software engineering methods and would serve as a testbed for Abramson's ideas about how a large uniform virtual memory, one of the requirements of our new hardware, could be implemented. The virtual memory technique itself (Abramson, 1981) and the software support for it (Rosenberg and Keedy, 1981b) soon proved to be feasible. In the course of the MONADS II work Abramson has also devised a model for the design of hardware to support a flexible and powerful basic addressing and protection structure for modular software. This is the subject of a companion paper (Abramson, 1982).

In 1981 funds became available to commence work on a new and much larger computer system, MONADS III, which is not based on an existing computer. Another companion paper (Keedy, Rosenberg, Abramson and Rowe,1982) discusses the relationship between MONADS II and MONADS III in detail and explains how a single portable operating system will be used for both. Although many of the MONADS II ideas have been reused in MONADS III, in some respects the two systems are quite different. For example the basic configuration of MONADS III makes more use of distributed processing units (Rosenberg, Rowe and Keedy, 1982) and in a companion paper there is a description of how the problems of connecting the various units are solved (Rowe, 1982). In the final companion paper, the aims and nature of the MONADS III instruction set are described (Rosenberg, 1982).

5. CURRENT SOFTWARE RESEARCH

Both the MONADS II and MONADS III systems make provision for a substantial amount of microcode which will be used not only for the basic instruction set but also for a set of "system management instructions". Many of

these have functions which are quite unlike instructions supplied in other computers. Together with the memory management and addressing hardware they provide the special mechanisms needed to support information-hiding systems in a flexible and efficient way. A provisional implementation of these instructions has been completed by a research student as part of a hardware kernel for the MONADS II system; this is based on the principles proposed by Rosenberg for MONADS I (Rosenberg, 1979; Rosenberg and Keedy, 1978) but it differs considerably in detailed approach. At present another student is developing the remaining modules of an operating system. This will be portable between MONADS II and MONADS III and will provide users of the two systems with the required support for developing application systems according to the information-hiding principle.

The operating system modules are being written in Pascal and a somewhat modified version of Modula-2. Compilers for these languages have been developed in Pascal. These compilers are themselves decomposed into information-hiding modules and provide an example of how the information-hiding principle can be used both to define sharable general purpose modules and to improve software portability (Rosenberg and Keedy, 1981a).

The fine details of the relationships between system management instructions, hardware kernel, operating system and compilers are still being studied, and the results so far are promising. It is likely that some of these results will be submitted to the organisers of the next Australian Computer Conference.

ACKNOWLEDGEMENTS. The authors acknowledge with thanks the work of other members of the MONADS team who have contributed to the design and implementation of the MONADS II and MONADS III hardware and software, especially Peter Dawson, Mark Evered, Mark Halpern, Ed. Gehringer, Glenda Patterson, Kotigiri Ramamohanarao, Ian Richards, John Thomson, Brian Wallis and John Wells.

We also gratefully acknowledge the funding provided by the Australian Research Grants Scheme (grants F77/15337 and F80/15191), by the Monash Special Research Fund (grant SC18/79) and by the Australian Computer Research Board (grant SB3/11).

REFERENCES

Abramson, D.(1981) "Hardware Management of a Large Virtual Memory", Proc. 4th Australian Computer Science Conference, Brisbane (Australian Computer Science Communications 3, 1, pp. 1-13).

Abramson, D.(1982) "Hardware for Capability Based Addressing", Proc. 9th Australian Computer Conference, Hobart.

Georgiades, A., Richards, I. and Keedy, J.L.(1978) "A File System for the MONADS Operating System", Proc. 8th Australian Computer Conference, Canberra,pp. 547-558.

Keedy, J.L.(1978) "The MONADS Operating System", Proc. 8th Australian Computer Conference, Canberra, pp. 903-910.

Keedy, J.L.(1981) "A Progress Report on the MONADS Project" Australian

Computer Science Communications, 3, 2, pp. 270-277.

Keedy, J.L.(1982) "The MONADS View of Software Modules", Proc. 9th Australian Computer Conference, Hobart.

Keedy, J.L. and Ramamohanarao, K.(1979) "A Job Management Model for In-Process Systems", MONADS Report No. 7, Monash University.

Keedy, J.L. and Richards, I.(1982) "A Software Engineering View of Files", Australian Computer Journal, 14, 2.

Keedy, J.L. and Rosenberg, J.(1981) "Information Hiding - A Key to Successful Software Engineering", Proc. Conference on Computers in Engineering, 1981, Institution of Engineers, Australia, Publication No. 81/8, pp. 1-5.

Keedy, J.L., Rosenberg, J., Abramson D. and Rowe, D.M.(1982) "A Comparison of the MONADS II and III Computer Systems", Proc. 9th Australian Computer Conference, Hobart.

Naur, P., Randell, B. and Buxton, J.N.(1976) "Software Engineering Concepts and Techniques", Petrocelli/Charter, New York.

Parnas, D.L.(1971) "Information Distribution Aspects of Design Methodology", Proc. 5th World Computer Congress, IFIP-71, pp. 339-344.

Parnas, D.L.(1972) "On the Criteria to be Used in Decomposing Systems into Modules", Comm. ACM, 15, 12, pp. 1053-1058.

Ramamohanarao, K.(1980) "A New Model for Job Management Systems", Ph.D. Thesis, Monash University.

Ramamohanarao, K. and Keedy, J.L.(1978) "Job Management in the MONADS Operating System", Proc. 8th Australian Computer Conference, Canberra, pp. 1476-1488.

Richards, I. and Keedy, J.L.(1978) "Subsystem Management in the MONADS Operating System", Proc. 8th Australian Computer Conference, Canberra, pp. 1520-1529.

Rosenberg, J.(1979) "The Concept of a Hardware Kernel and its Implementation on a Minicomputer", Ph.D. Thesis, Dept. of Computer Science, Monash University.

Rosenberg, J.(1982) "The MONADS Series III Instruction Set", Proc. 9th Australian Computer Conference, Hobart.

Rosenberg, J. and Keedy, J.L.(1978) "The MONADS Hardware Kernel", Proc. 8th Australian Computer Conference, Canberra, pp. 1542-1552.

Rosenberg, J. and Keedy, J.L.(1981a) "Information Hiding - A Case Study", Proc. Conference on Computers in Engineering, 1981, Institution of Engineers, Australia, Publication No. 81/8, pp. 6-9.

Rosenberg, J. and Keedy, J.L.(1981b) "Software Management of a Large Virtual Memory", Proc. 4th Australian Computer Science Conference, Brisbane, pp. 173-181.

Rosenberg, J., Rowe, D.M. and Keedy, J.L.(1982) "An Overview of the MONADS

Series III Architecture", Proc. 5th Australian Computer Science Conference, Perth, pp. 58-67.

Rowe, D.M.(1982) "MONADS III Inter-Unit Communication" Proc. 9th Australian Computer Conference, Hobart.

Wallace, C.S.(1978) "Memory and Addressing Extensions to a HP2100A" Proc. 8th Australian Computer Conference, pp. 1796-1811, Canberra.

Proceedings of the

Ninth Australian Computer Conference