

systematically transformed into commitments, and how commitments are influenced by the constraints of the execution environment.

The logical architecture of the pump control system embodies design commitments about object classes, interfaces and logical connectivity. The physical architecture embodies commitments about object replication, object location and the use of RPC for communication. These commitments were made through systematic refinement of the logical architecture, treating the non-functional projections in turn. This process also made explicit the interactions between projections.

The physical architecture also spells out obligations on the implementation and on the execution environment. Obligations on the implementation include

- formal specification and verification of the pump controller and environment monitor;
- implementation of NMR and standby techniques;
- code execution within prescribed periods and deadlines.

Obligations on the execution environment include

- provision of replicated processors for the pump controller and environment monitor;
- provision of a reliable RPC mechanism with variable time-out facility and local call optimisation.

Development of the physical architecture included arguments that if these obligations are met then the non-functional requirements of the system will be satisfied. Such an assurance, obtained before detailed design or implementation starts, is one of the major benefits of using TARDIS. It stands in contrast to the prevailing practice of designing a system to meet its functional requirements and then trying to meet the non-functional requirements through testing and tuning.

Acknowledgements

The authors wish to thank Andy Wellings and John McDermid for their contributions to the ideas developed in this paper. Andrew Lister also gratefully acknowledges the support of a SERC (UK) Visiting Fellowship.

REFERENCES

1. G. Booch, *Software Engineering with Ada* (2nd edition). The Benjamin/Cummings Publishing Company, Inc. (1986).
2. A. Burns and A. J. Wellings, *Real-time Systems and their Programming Languages*, Addison Wesley (November 1989).
3. A. Burns and A. M. Lister, An architectural framework for timely and reliable distributed information systems (TARDIS): description and case study. YCS. 140, Department of Computer Science, University of York (1990).
4. H. Kopetz, Design principles for fault tolerant real time systems. *MARS Report*, Institut für Technische Informatik, 8/85/2 (1985).
5. J. Kramer, J. Magee, M. S. Sloman and A. M. Lister, CONIC: an integrated approach to distributed computer control systems. *IEE Proceedings (Part E)* **180** (1), 1–10 (Jan. 1983).
6. J. Kramer and J. Magee, A model for change management. In *Proceedings of the IEEE Distributed Computing Systems in the '90s Conference* (September 1988).
7. J. C. Laprie, Dependability: a unified concept for reliable computing and fault tolerance, pp. 1–28. In *Resilient Computer Systems* (ed. T. Anderson). Collins and Wiley (1989).
8. B. Meyer, Reusability: the case for object-oriented design. *IEEE Software* **4** (2), 50–64 (March 1987).
9. B. Meyer, *Object-Oriented Software Construction*. Prentice-Hall, Inc, Englewood Cliffs, New Jersey (March 1988).
10. B. Randell, P. A. Lee and P. C. Treleaven, Reliability issues in computing system design. *ACM Computing Surveys* **10** (2), 123–165 (June 1978).
11. S. K. Shrivastava, L. Mancini and B. Randell, On the duality of fault tolerant structures, pp. 19–37. In *Lecture Notes in Computer Science*. Springer-Verlag (1987).
12. M. Sloman and J. Kramer, *Distributed Systems and Computer Networks*. Prentice-Hall (1987).
13. A. Vrchoticky and W. Schütz (eds.), *Real-Time Systems (Specific Closed Workshop)*, *ESPRIT PDCS Workshop Report W3*. Institut für Technische Informatik, Technische Universität Wien, Vienna (January 1990).

Announcement

11–15 May 1992

Fourteenth International Conference on Software Engineering, Melbourne, Australia

Call for Papers

Please submit 6 copies of full paper or experience report (3000–6000 words), tutorial proposal, panel session or tools exhibit proposal to Professor A. Y. Montgomery, General Chair, Department of Computer Science, Royal Melbourne Institute of Technology,

P.O. Box 2476V, Melbourne 3001, Victoria, Australia. Phone (61)-3-660 2943; Fax (61)-3-662-1617. E-mail aym@goanna.cs.rmit.oz.au.

Deadline: **6 September 1991**

The Conference

The fourteenth International Conference on Software Engineering will be held at the World Congress Centre, Melbourne, Australia. It is sponsored by IEEE, the British Computer

Society, ACM SIGSOFT in conjunction with the Australian Computer Society, the Institution of Engineers Australia, and the Institution of Radio and Electronic Engineers Australia. Enquiries to Professor A. Y. Montgomery, General Chair, Department of Computer Science, Royal Melbourne Institute of Technology, P.O. Box 2476V, Melbourne 3001, Victoria, Australia. Phone (61)-3-660-2493; Fax (61)-3-662-1617. E-mail aym@goanna.cs.rmit.oz.au.