Unified Hardware and Software Models for Smart System Design

Ravi Kapadia
Department of CS, Vanderbilt University
Box 1679, Sta B
Nashville TN 37235
email: ravi@vuse.vanderbilt.edu

"Smart" electromechanical systems (e.g., washing machines, car anti-skid braking systems and reprographic machines, i.e., printers and photocopiers) are machines with intelligent, embedded computer control. Conventional embedded controllers performed a preprogrammed set of actions, but as machines have become more versatile, controllers are being designed to generate and evaluate different plans to perform specified tasks, search for optimal actions, and react to changing environments. Traditionally, hardware design in embedded systems has preceded software design resulting in sub-optimal designs. As the use of embedded software proliferates, integrated hardware and software design becomes critical.

Like conventional design, smart system design explores the space of possible solutions to develop one that satisfies functional and performance specifications and that optimizes desired criteria such as speed, cost, and power consumption. Smart system design is a difficult problem because decisions must be made about allocating functionality between hardware and software such that the combined design performs correctly and optimizes desired criteria. Furthermore, anytime algorithms must be designed where both planning and execution time affect overall performance. Finally, because there are dependencies between the hardware and the software, the exploration of the design space must consider both the hardware and the software. Adopting an integrated approach over the present, sequential methodology, will improve quality and reduce the time taken to design complex, smart systems.

A key to exploring the design space effectively, is to develop a unified representation of hardware and software at suitable levels of abstraction. [1] has developed discrete hardware models for scheduling sheets in reprographic machines. We propose a unified, hybrid modeling scheme that integrates discrete hardware and software models to aid in the task of generating and evaluating design alternatives for smart systems.

We focus on the design of system hardware and control software (primarily, scheduling discrete operations) in the reprographic machine domain. Given alternative hardware configurations, computational resources (e.g., memory, processor speed) and algorithmic strategies (e.g., search control heuristics), the design task is to optimize parameter values (e.g., speed of components, capacities of trays), and to match hardware configurations, algorithmic strategies, and computational resources to satisfy functional specifications and optimize performance, cost, and machine wear.

[3] adapts the Environment Relationship (ER) Net [2] framework (an extension of Petri nets, that represent time explicitly and associate properties with tokens) to model the sub-system that transports sheets of paper in the reprographic machine. The next step is to represent algorithmic strategies, computational resources, and hardware-software interactions, and, finally, to incorporate the models in the architecture for automated design optimization presented in [4].

References

[1] M. Fromherz and B. Carlsson. Optimal Incremental and Anytime Scheduling. In Proc. Workshop on Constraint Languages/Systems and their Use in Problem Modeling at ILPS'94, 45–59. Nov. 1994.

[2] C. Ghezzi, D. Mandrioli, S. Morasca, and M. Pezze. A Unified High-level Petri Net Formalism for Time Critical Systems. In IEEE Trans. on Software Engg., 17(2):160–172, Feb. 1991.

[3] R. Kapadia, G. Biswas, and M. Fromherz. Hybrid Modeling for Smart System Design. In Proc. 10th Florida AI Research Symposium, FL, May 1997.

[4] R. Kapadia and M. Fromherz. Design Optimization with Uncertain Application Knowledge In Proc. of IEA-AIE'97, 10th Intl. Conf. on Industrial and Engg. Applications of AI and Expert Systems, Atlanta, June 1997.