

SOFTWARE ISSUES IN CRITICAL TRANSPORTATION SYSTEMS

Principal Investigator: **Dr. Mats Heimdahl**,
Associate Professor

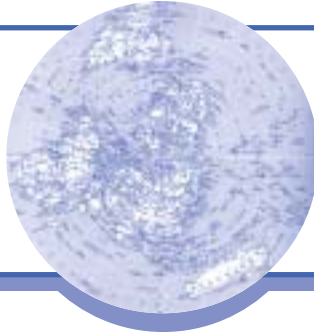
Dept. of Computer Science and Engineering

University of Minnesota • Twin Cities

Phone: 612-625-2068

E-mail: heimd002@umn.edu

Web: <http://umn.edu/home/heimd002>



Why This Research is Needed

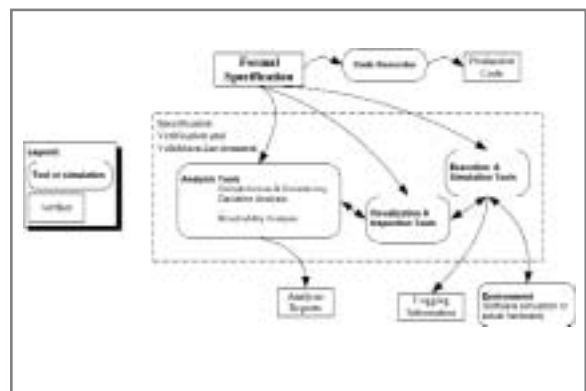
Vehicles are increasingly reliant on software-based control systems to manage functions such as throttle control and braking. These “drive-by-wire” systems are analogous to the computer-driven control systems of newer military and commercial aircraft. In these critical systems, failure during operation can have catastrophic results. Engineers designing software for critical systems need tools that allow them not only to guard against the potentially life-threatening consequences of software malfunction, but avoid the high costs of reworking critical software late in the development process.

Research Objective

To gain a formal understanding of the issues involved in critical software systems design, and to develop tools which will allow software engineers to build these systems safely and efficiently.

Methodology

The Critical Systems Research Group, established by Heimdahl, focused its early work on the development of easy-to-use formal modeling languages suitable for the modeling of embedded control applications, as well as on the application of special purpose static analysis techniques. This work provided the foundation for their work on a requirements engineering environment, called NIMBUS, that allows both static and dynamic evaluation of formal requirements models. The NIMBUS environment has formed the testbed for the group’s continued research into requirements modeling. The capabilities of NIMBUS allows a modeling technique which the group terms “specification-based prototyping”



Professor Mats Heimdahl (top) with students xx and xxx; an overview of the NIMBUS architecture.

that allows an engineer to use the formal requirements model as a prototype for dynamic evaluation purposes (including hardware-in-the-loop simulation).

While continuing to explore formal modeling and model validation, Heimdahl’s group has investigated how a formal foundation can be leveraged to reduce software development costs and increase software quality.

Professor Heimdahl participated in the development of a set of formal criteria describing desirable properties of requirements specifications for embedded software systems. He then pursued the development of automated procedures to analyze formal models for some criteria. His work focused on the analysis of models expressed in the Requirements State Machine Language (RSML), a high-level graphical requirements language suitable for the specification of embedded systems.

Research Impacts

Several groups around the country, have developed requirements checklists based on the formal criteria developed in part by Heimdahl and his team. In an experiment, researchers at the Jet Propulsion Laboratory determined that the formal criteria could identify 150 of the 192 safety-critical errors detected in late system testing of the Galileo and Voyager spacecraft.

What's Next

The Critical Systems Group plans to continue their work in formal software development. In particular, they are interested in reducing development cost through the reuse of formal specification fragments within and across product families. They are also interested in the potential to leverage the capabilities of NIMBUS to decrease the high costs associated with testing and improve static analysis through the inclusion of model checking and theorem proving techniques. In the following sections, the students involved in these efforts outline their research projects.

Related Publications

M.P.E. Heimdahl and B.J. Czerny. On the Analytical Power Needed When Analyzing State-Based Requirements: An Experience Report. *Science of Computer Programming*, Vol-36, Issue 1, pp. 65-96 (January 2000).

J.M. Thompson, M.P.E. Heimdahl, D.M. Erickson, Structuring Formal Control Systems Specifications for Reuse: Surviving Hardware Changes, *Fifth NASA Langley Formal Methods Workshop*, June, 2000.

M.S. Jaffe, N.G. Leveson, M.P.E. Heimdahl, B.E. Melhart. Software Requirements Analysis for Real-Time Process-Control Systems. *IEEE Transactions on Software Engineering*, 17(3): 241-258, March 1991