

N89-16343

RT_BUILD : An Expert Programmer for Implementing and Simulating Ada Real-Time Control Software

*Larry L. Lehman**
*Steve Houtchens***
*Massoud Navab***
Sunil C. Shah†

Integrated Systems Inc.

101 University Ave.

Palo Alto, CA 94301

(415)-853-8400

Abstract

RT_BUILD™ is an expert control system programmer that creates real-time Ada™ code from block-diagram descriptions of control systems. Since RT_BUILD embodies substantial knowledge about the implementation of real-time control systems, it can perform many, if not most of the functions normally performed by human real-time programmers. Though much basic research has been done in automatic programming, RT_BUILD appears to be the first application of this research to an important problem in flight control system development. In particular, RT_BUILD was designed to directly increase productivity and reliability for control implementations of large complex systems.

RT_BUILD Capabilities

RT_BUILD implements control systems designed with the MATRIX_X™ control design package. Control systems are specified as nonlinear, multi-rate, discrete-time block-diagrams in the interactive graphical environment of MATRIX_X's SYSTEM_BUILD™ module. RT_BUILD accesses the SYSTEM_BUILD data-base to create an exact implementation of the design.

The code produced by RT_BUILD includes all specified dynamic compensation and control logic, a real-time application level executive, and generic software interfaces to hardware such as sensors, actuators, and displays.

* *Manager, Software Development*

** *Member, Technical Staff*

† *Manager, Real-Time Systems*

The efficient executive performs the task scheduling and interrupt handling required to implement real-time multi-rate controllers. Initialization/termination and exception handling functions are also included.

The software elements for an application are automatically assembled from a modular library of Ada control functions. User supplied control or interface functions (such as specific hardware drivers or existing Ada control algorithms) can also be included from user libraries.

The real-time code produced by RT_BUILD is highly optimized for speed. Further optimization (speed or memory) is normally performed by optimizing compilers. The net result is an automatic implementation of the design whose real-time performance is extremely difficult to surpass.

RT_BUILD can be used at the final stage of the control engineering cycle to generate real-time code for most control problems including: aircraft/spacecraft control, robotics, process control, servo control, and any other real-time Ada application. Earlier in the engineering cycle, RT_BUILD can be used to develop off-line simulations or real-time hardware-in-the-loop simulations. Since RT_BUILD can be used to rapidly estimate real-time processing requirements, implementation considerations and processor constraints can be considered much earlier in the control development cycle.

RT_BUILD Design Goals

RT_BUILD was designed to address the requirements of flight software. The following aspects distinguish flight software from real-time software used in prototype testing.

- (i) Flight software is used over a long time period. It must be very easy to maintain, update, and verify.
- (ii) Processing power and memory are at a premium. They must be used efficiently.
- (iii) The real-time software must be capable of handling a variety of input-output mediums.
- (iv) The software must handle multi-processor implementations, since most flight control systems must use many processors.
- (v) The software must accurately implement the designed control systems since costs of making errors can be very high.

The paper will describe how recent developments in computer-aided engineering have been applied to meet these requirements in RT-BUILD.

RT-BUILD Design Approach

Real-time software is generated around the concept of a modular, reconfigurable periodic scheduler and associated computation modules that can coexist with other foreground computing tasks (e.g., interrupt service routines) as well lower priority background processes. Device independent interfaces are incorporated in this design to isolate hardware dependencies. The Ada Periodic Scheduler (APS) is designed to provide periodic pre-emptive priority-based execution of tasks. The scheduler as well as the rest of the software is written in Ada.

The detailed paper will show various details of the design procedures and software structure as well as the role Ada capabilities play in this automatic code generation.

A Multi-Disciplinary Technology

The paper will discuss the multi-disciplinary technology required to develop an integrated set of Computer-Aided Control Engineering (CACE) tools which include automatic code generation capabilities.

While current tools provide a tremendous improvement in flight control systems development, areas where further research is being conducted will be covered in the detailed paper and the presentation.

TM RT-BUILD, MATRIX_x, and SYSTEM-BUILD are trademarks of Integrated Systems Inc., Ada is a trademark of the U.S. Department of Defense (Ada Joint Program Office).