
Introducing a Target-Based Approach to Rapid Prototyping of ECUs

FEBRUARY, 1997

Abstract

This paper presents a target-based approach to Rapid Prototyping of Electronic Control Units (ECUs). With this approach, the constraints of actual production ECUs are introduced early in the development cycle. This reduces the cost impact of software defects and fosters a more quality-assured software engineering process. This paper also presents ADvantage, a Computer Aided Control System Design (CACSD) toolset. ADvantage is used to implement the target-based approach to Rapid Prototyping of ECUs (RPE). ADvantage also fully supports other methods of rapid prototyping such as the idealized environment RPE approach often used today.

1. Introduction

Controls engineers commonly use RPE to develop and assess control strategies and to provide software engineers with controller algorithm requirements. RPE is usually performed using graphical controller design packages that automatically generate code for execution on portable computers. The computers simulate the control laws, and interface directly to the actual (or emulated) plant — often an automobile driven on a test track. Because it executes floating-point code on generic processors, this approach is sometimes termed “idealized RPE.”

There is a growing recognition within the automotive industry that while RPE in an idealized environment is better than *earlier methods* (described below), it is still not the ideal prototyping paradigm.

The main problem with this approach is that controller algorithms developed in an idealized environment behave differently when executed in the constrained production environment. Constraints such as fixed-point arithmetic, RTOS scheduling overhead, and CPU performance are considered by many RPE practitioners as too important to ignore.

Another area of concern is that the controller algorithms supplied by controls engineers using idealized RPE often undergo significant changes during target implementation by the software engineers. The changes include *conversions* such as floating-point to fixed-point, Laplace to Z-domain to discrete equations, and *optimizations* for less memory and faster execution. In many cases, these changes either are not documented or are not made known to the controls engineer.

But for all its faults, the advantages of rapid, automated development far outweigh the benefits of earlier prototyping methods, which did execute the code on the target. This is because the *earlier methods* required software engineers to convert the controller algorithms by hand to target-specific assembly code. While accurate, this was a painfully slow and error-prone procedure.

2. The Throwaway Prototype Model

In a paper presented at the 1996 IEEE CACSD Symposium [1], Delphi Chassis Systems concluded that:

Rapid prototyping is not efficient for making minor modifications to existing production control systems or for specifying requirements for production intent fixed-point processors late in the development cycle.

On the positive side, Delphi found that RPE shortens development cycles by using an automated approach (i.e., Rapid Prototyping) and that RPE is appropriate for new concept evaluation or feasibility studies.

It is generally agreed that RPE facilitates the process of deriving initial software requirements by allowing controls engineers to quickly build and assess working prototypes. Unfortunately, after the prototype assessment is finished, the prototype models are not used for subsequent ECU software development steps, such as fixed-point requirement specification, design, coding, testing, and maintenance. In essence, the *idealized RPE* models are thrown away early in the development cycle.

3. The Evolutionary Prototype Model

A more efficient RPE paradigm would still provide an idealized RPE capability, but it would then allow for rapid implementation and execution of fixed-point code on the target CPU. Either the actual or a functionally-equivalent RTOS would be used. The target-based RPE models could be reused and refined as the ECU development cycle evolves.

Evolutionary Rapid Prototyping is an iterative software development method that uses the

target-based RPE model as the first design iteration. During the last few years, the Evolutionary Rapid Prototyping model has gained much support within the software engineering community at large. Refer to the *Proceedings of the International Workshop on Rapid System Prototyping*, sponsored by IEEE Computer Society, for recent advancements in Rapid Prototyping technology.

4. The Accuracy Question

Another concern with idealized RPE is that the prototyping model will behave differently than the production unit, but how differently? In their paper describing a Virtual Engine RPE approach [2], Toyota Motor Corporation identified discrepancies between engine data calculated by the production CPU and engine data calculated by the prototype CPU. Toyota attributed these "slight differences" to floating versus fixed-point implementations, poor control-flow models, and inadequate interrupt capability. Target-based RPE greatly reduces these differences and, in some cases, eliminates them.

5. Expanding Engineering Roles

In the future, market forces will require more sophisticated controller algorithms with enhanced functionality. At the same time, the market will resist added costs due to increases in processor size and speed. Thus, controls and software engineers will need to work more closely together from project inception, as they estimate word size effects and the computational delay associated with each new control strategy. These estimates will require accurate timing and memory usage studies and truncation/round-off error analyses. It is at this point, that target-based RPE will no longer be merely optional.

6. The ADvantage Solution

In an often cited paper [3], Barker et al., presented a framework model for a CACSD working environment. The authors repeatedly stressed the importance of an open system architecture:

Open systems allow [users] to choose the products with the best price/performance features knowing that they will work as expected. With a closed proprietary system it is the developer who has control over the system development, so the user is ... locked into a single supplier.

Applied Dynamics International (ADI) has developed a CACSD environment, ADvantage, which embraces many aspects of the framework model mentioned above. In particular, ADvantage is based on many published standards and provides automated links, or bridges, to popular commercial CACSD packages.

The ADvantage open architecture benefits RPE engineers in many ways. It allows engineers to:

- maintain their existing model investments as they migrate to a target-based RPE method
- develop their own custom RPE environment by mixing products from four of the top CACSD vendors and more than twelve major hardware suppliers
- easily incorporate an unlimited variety of standards-based I/O devices and software models, either developed in-house or provided by other commercial vendors

ADvantage adheres to the following published standards:

- ANSI C and FORTRAN 77 language standards
- IP Module standard (ANSI/VITA 4-1995)
- VMEbus hardware standard (including the recent high bandwidth VME64 addendum)

The ADvantage Integrated Development Environment (IDE) is the software manifestation of the ADvantage CACSD framework. The ADvantage IDE provides a common project platform and integrated tool links, as shown in Figure 1.

Figure 1. The ADvantage IDE

7. ADvantage Software Architecture

The ADvantage software architecture uses a host/target approach rather than a self-hosted method. With host/target architecture, the kernel needed for a target processor is small and can easily be ported to any processor or microcontroller. In such a system, it is also easier to ensure deterministic behavior and give users meticulous control of the code execution, should they desire it.

Figure 2 shows the target software architecture that executes on each processor within the Applied Dynamics Real-Time Station (AD RTS). The targets include floating-point CPUs such as PowerPC 604 and fixed-point microcontrollers such as MC68332. Thus, both idealized and target-based RPE are supported.

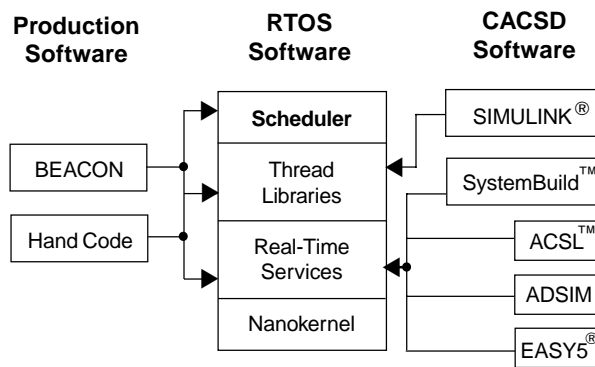


Figure 2. Target Software Architecture

7.1. CACSD Software

The CACSD software interface shown in Figure 2 is based on the source code output from each tool's automatic code generator (e.g., ISI's AutoCode™ and The MathWorks' Real-Time Workshop™). With the ADvantage IDE, models from different CACSD packages are easily connected for multiple processor execution (e.g., a SIMULINK plant model output can be

“sensed” by a controller developed in SystemBuild).

7.2. BEACON Software

BEACON differs from most CACSD tools in that its design intent was to automate the development and testing of software for embedded microcontrollers, and not to generate code for simulation on general-purpose CPUs. BEACON can be used for both idealized and target-based RPE because it generates either floating or fixed-point software for every block in its palette. But target-based RPE requires much more than efficient fixed-point capability. For example, software developers need, and BEACON provides, fine-grain control over the code generation with little intervention from an under-the-scenes sorter (i.e., no surprises). Software developers also need a diverse control-flow capability; BEACON's control-flow palette contains blocks for sequential flow, conditional/predicate logic, and repetition. One of BEACON's most important features is that it produces diagrams that can be shared and reviewed by controls and software engineers as precise, unambiguous software requirements and design specification. As the Software Engineering Institute and many others believe, software reviews are one of the most important parts of any software development process [4].

7.3. RTexec Software

RTexec is ADvantage's instrumented real-time kernel. RTexec provides standard real-time libraries, device drivers, and multithread facilities. A set of basic thread libraries is provided to let programmers develop their own schedule schemes. In addition, an RTexec scheduler is available that performs Rate Monotonic Scheduling and allows for event-triggered tasks.

The priority of any task, whether rate or event based, can be set by the programmer. RTextec also has mechanisms for mutual exclusion, exception handling, and thread profiling.

7.4. Host-based Software

The host environment is the platform for the CACSD packages, BEACON, and ADvantage IDE. The host environment also provides a full-featured suite of software support tools.

- Interact for source-level debugging (such as breakpointing, single stepping and display of the function call stack).
- SIMpanel for an animation, calibration and data acquisition environment based on the popular software from DataViews.
- Online help for quick information access.

These support tools are very flexible and can be used to perform a wide range of analyses. The SIMpanel display that profiles thread timing statistics (Figure 3) is just one example.



Figure 3. Thread Profiling

8. ADvantage Hardware Architecture

The ADvantage RPE hardware component is the AD RTS family of products, and in particular, the portable In-Vehicle System (IVS). The AD RTS has a VMEbus that provides a multiple processing, distributed execution environment via Single Board Computers (SBC). These SBCs include:

- User Interface Processor (UIP) for host/target communication.
- Compute Engines (CEs) for executing plant models or idealized RPE controllers.
- Parallel Intelligent Resources (PIRs) for high and low speed I/O on a localized bus.
- Target ECU (an IP Module) for target-based RPE and I/O servicing.

The target ECU device contains the target microcontroller on an IP Module. Dual Ported Memory (DPM) provides an interface between the microcontroller and the IP Module Bus (IP Bus). Information on the IP Bus is then available to any other board within the AD RTS. For example, the User Interface Board (UIB) can be used to send/receive data between the microcontroller and host, and thus the RPE engineer. In another case, a compute engine may be used to simulate a plant's response to the controller signals for simulation studies. See Figure 4.

In addition to the DPM, the target microcontroller also has access to local SRAM and PROM, currently 256KB each, but subject to change.

This isolates the target microcontroller from the IP Bus. Thus, the IP Bus (and the entire AD RTS system) cannot interfere with the target microcontroller during execution except in a controlled manner via the DPM.

The I/O between the external hardware (or test vehicle) and AD RTS is possible in many forms including custom pod (such as DPM), special purpose communication buses (including J1850 or CAN), or directly from the target ECU using its own I/O facilities.

Using I/O directly from the target ECU is preferred when performing target-based RPE. The other I/O methods are most applicable when performing idealized RPE. If these I/O options are not adequate, ADI's Systems Integration Services is available to provide custom I/O solutions.

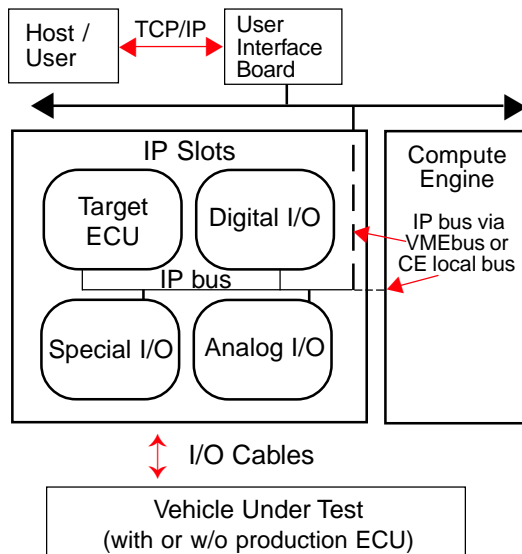


Figure 4. Hardware Architecture

9. Conclusions

Target-based rapid prototyping is likely to become the preferred approach for the next generation of RPE. ADI has developed a product line, ADvantage, which provides a target-based capability while preserving and maintaining a customer's existing RPE modeling investments. The AD RTS also supports multiple processors and high bandwidth communications such that a multitude of real-time analyses can be performed within the same environment. Analyses can range from low-speed idealized RPE to high-fidelity, asynchronous engine simulation for hardware-in-the-loop testing.

10. References

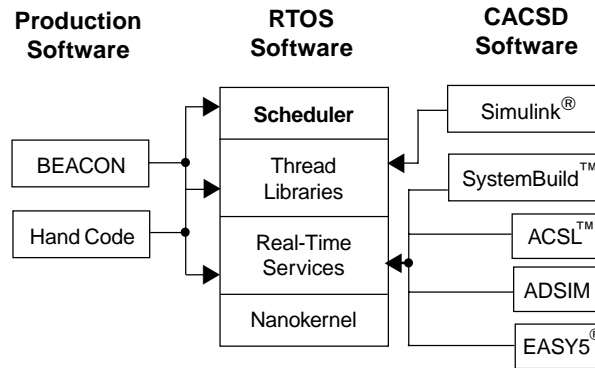
- [1] M. R. DePoyster, J. F. Hoying, K.N. Majeed, "Rapid Prototyping of Chassis Control Systems," Proceedings of the 1996 IEEE International Symposium on Computer-Aided Control System Design.
- [2] A. Kimura, I. Maeda, "Development of Engine Control System Using Real Time Simulator," Proceedings of the 1996 IEEE International Symposium on Computer-Aided Control System Design.
- [3] H.A. Barker, M. Chen, P.W. Grant, C.P. Jobling, P. Townsend, "Open Architecture for Computer-Aided Control Engineering," IEEE Control Systems, April 1993.
- [4] M. Paulk, et al., "Capability Maturity Model for Software," Software Engineering Institute, Carnegie Mellon University, Pittsburgh, PA, 1993.

11. About ADI

Applied Dynamics International provides hardware, software, and engineering services for embedded software developers and control system engineers. Since 1957, ADI real-time computer systems have been used extensively in advanced control system applications in the aerospace, defense, training, and automotive industries. For more information on the complete ADI product family, please contact us at:

Applied Dynamics International, Inc.
 3800 Stone School Road
 Ann Arbor, MI 48108-2499
 USA
 Telephone: (313) 973-1300
 Facsimile: (313) 668-0012
 E-Mail: adinfo@adi.com
<http://www.adi.com>

Applied Dynamics International Limited
 Oxford House, Oxford Street
 Wellingborough
 Northamptonshire NN8 4HG
 United Kingdom
 Telephone: 44-1933-272666
 Facsimile: 44-1933-273024
 E-Mail: adinfo@appdyn.uucp@britain.eu.net
<http://www.adi.com>



11. About ADI

Applied Dynamics International provides hardware, software, and engineering services for embedded software developers and control system engineers. Since 1957, ADI real-time computer systems have been used extensively in advanced control system applications in the aerospace, defense, training, and automotive industries. For more information on the complete ADI product family, please call us at (313) 973-1300 or visit our web site: <http://www.adi.com>.

Applied Dynamics International, Inc.
 3800 Stone School Road
 Ann Arbor, MI 48108-2499
 USA
 Telephone: (313) 973-1300
 Facsimile: (313) 668-0012
 E-Mail: adinfo@adi.com
<http://www.adi.com>

Applied Dynamics International Limited
 Oxford House, Oxford Street
 Wellingborough
 Northamptonshire NN8 4HG
 United Kingdom
 Telephone: 44-1933-272666
 Facsimile: 44-1933-273024
 E-Mail: adinfo%appdyn.uucp@britain.eu.net
<http://www.adi.com>