Reducing Maintenance Costs Through the Application of Modern Software Architecture Principles

Christine Hulse and
Scott Edgerton
United Defense, LP
Minneapolis, Minnesota

Michael Ubnoske
Architecture Technology
Minneapolis, Minnesota

Louis Vazquez
Department of the Army
OPM Crusader
Picatinny Arsenal,
New Jersey

Abstract

Large software programs are usually long lived and continually evolve. Substantial maintenance effort is often extended by engineers trying to understand the software prior to making changes. To successfully evolve the software, a thorough understanding of the architect's intentions about software organization is required. Software maintenance costs can be reduced significantly if the software architecture is well defined, clearly documented, and creates an environment that promotes design consistency through the use of guidelines and design patterns. Building a maintainable system depends upon the consistent application of these architectural practices. This paper describes the application of modern software architecture methods to achieve a maintainable implementation of a large, distributed, real-time, embedded software system.

Introduction

The Crusader Field Artillery System is the U.S. Army's next generation 155-mm self-propelled howitzer (SPH) and its companion resupply vehicle (RSV). The Crusader is a software-intensive system that is being designed to last well into the next century. Ada95 is the implementation language for all software development on the Crusader program. Modern software architecture methods are being applied to capture the general principles used throughout the design and to provide a guide for further development of the software. The architecture is being modeled using the Unified Modeling Language (UML) to bring together the design vision of all of the key stakeholders. Our software architecture is a high level model of the software contained inside a Crusader self-propelled howitzer vehicle or a resupply vehicle.

The Crusader software has been designed with the following architectural objectives:

- Support for an object-oriented design
- Support for distributed objects
- Support for implicit invocation
- Support for the Joint Technical Architecture - Army
- Resilience to change
- Support for meeting hard real-time requirements
- Support for domain reuse.

Documenting the software architecture in a rigorous way has many advantages for maintenance. Software maintainers must possess a good understanding of the existing code and a thorough understanding of the designer's intentions about software organization. Substantial maintenace effort is expended trying to understand the software in preparation for making changes. This maintenance effort can be reduced significantly if the software architecture is documented clearly and explicitly.

This paper describes our experiences in developing an object-oriented software architecture for the Crusader system using modern software architecture methods.