

JOVIAL - TRW - BMEWS - J3

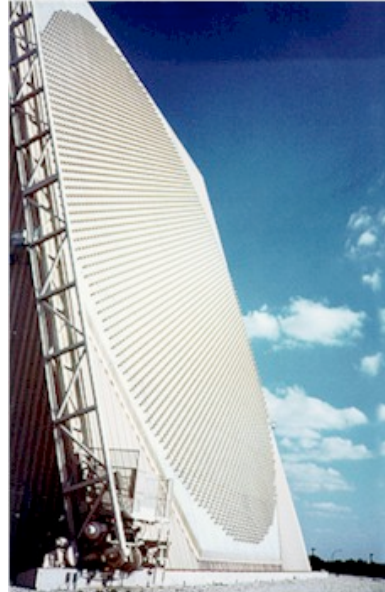
TRW

Ballistic Missile Early Warning System (BMEWS)

J3 JOVIAL to C++

History: The Ballistic Missile Early Warning System (BMEWS) radar system provides intercontinental and sea-launched ballistic missile detection, early warning and attack assessment capabilities. These early warning radar systems were being upgraded to improve target detection and mid-course target tracking for the hit-to-kill Exoatmospheric Kill Vehicle (KEV) of the Missile Defense Agency's (MDA) ballistic missile defense (BMD). The Upgraded Early Warning Radar System (UEWRS) will allow earlier detection, closer tracking, and more intercept opportunities per attacking warhead.

BMEWS was written in J3 JOVIAL, which required migration to modern C++ and Java. To support the National Missile Defense Deployment Readiness Program, TRW contracted with TSRI to demonstrate the feasibility of modernizing BMEWS using automated transformation technologies rather than resorting to a manual rewrite.



Challenge: Modernizing the radar system's software into C++ and Java significantly reduced operation costs, allowed target platform independence, increased processing capacity, and provided mission redundancy. J3 JOVIAL is a dated legacy software language that was widely used for many high performance mission critical embedded applications developed during the 1970s, 80s and 90s. Modernization of JOVIAL into C++ is a highly complex problem. For example, JOVIAL saves space in physical memory by using overlays to swap program segments, data fields are sized to odd bit lengths, and the JOVIAL macro definition mechanism is incompatible with that of C and C++. The 60-bit Cyber and 16-bit 1750 processors on which older JOVIAL systems run use word lengths and bit ordering that are inconsistent with modern processors. JOVIAL is a powerful real-time language used for mission critical applications with complex data structures, dynamic program restructuring, real-time hardware interrupt support and multi-level fault handling. Preservation of computation state during real-time interrupts and fault condition was required in the upgraded system.

Results: TRW undertook this research study to gain a better understanding of the schedule, cost, and risk mitigation benefits of using a highly automated approach to modernizing a major mission critical information system written in a real-time high-performance legacy language. TRW initially presumed transformation results with no better than 80% automation. However, by the end of this project, TRW software engineers, running TSRI tools at their own sites near Cheyenne Mountain in Colorado Springs, were transforming the BMEWS JOVIAL code in into C++ at a 99% level of automation. This collaboration between TRW and TSRI also demonstrated the capability to map between a high-level Rational Rose UML model of the UEWRS system and UML documentation generated automatically for the BMEWS system. BMEWS code was automatically lifted into data element tables, control flow graphs, state machine diagrams, state transition tables and class models. System-wide re-factoring operations also demonstrated the capability to automatically eliminate redundancy from derived C++ code. This technology demonstration project established the initial framework for TSRI's automated tools and services in providing automated assessment, transformation and re-factoring. TSRI's J3 JOVIAL to C++ transformation capability was later enhanced to include J73 JOVIAL. These tools and services have since been employed on other JOVIAL modernization

projects, including MILSTAR Command & Control System - Consolidation (CCS-C), and the F-16 Block 30 Up Front Control System.