

# COBOL - NGIT - REMIS

Northrop Grumman Information Technologies (NGIT)  
Reliability & Maintainability Information System (REMIS)

Tandem COBOL85 and Tandem SQL to C++ and Oracle9i

**History:** The Reliability & Maintainability Information System (REMIS) is a key component of the Air Force Depot Maintenance System. REMIS consisted of 3.1 million lines of source code written in COBOL85, TAL and C. The system runs on a Tandem Symmetric Multiprocessor with NonStop SQL/MP DDL and a Tandem database. Northrop Grumman IT (NGIT) maintained REMIS at Wright-Patterson AFB for the Air Force Material System Group (MSG). NGIT undertook Increment-1 of the REMIS migration into the GCSS-AF Integration Framework (IF) to demonstrate the feasibility and assess the

benefits, (economic, schedule, and qualitative), of using automated transformation to create a GCSS-AF Resident version of the REMIS Application. NGIT contracted with TSRI to support Increment-1 of the REMIS migration to the GCSS-AF, supporting creation of the IF Resident REMIS Application.



**Challenges:** To achieve compliance with GCSS-IF, the IF Resident REMIS Application was comprised of a web-enabled user interface developed from J2EE Java technology interfacing with re-factored C++ components. NGIT manually converted the existing REMIS presentation layer source code into web-enabled pages. TSRI used its automated technology to convert the existing REMIS database access layer source code into object oriented C++. A new Graphical User Interface (GUI), written as Java components, was separately developed by Northrop Grumman to interface with C++ components via a JAVA/C++ API developed jointly by TSRI and NGIT. The technical effort provided by TSRI included a four-phase code conversion process, consisting of Automated Assessment, Automated Transformation, Automated Re-Factoring, and Semi-Automated Re-Factoring.

**Assessment:** During automated generation of "As Is" documentation, structure charts, control flow graphs, state machine models, data flow diagrams, and data element tables were generated to GCSS documentation standards for roughly 300,000 lines of COBOL85. The documentation was delivered as a large web document consisting of several million web pages with scalable, high fidelity vector graphics providing easy accessibility by any number of NGIT or Air Force users.

**Transformation:** TSRI used the eVolution 2000 Toolset to convert existing REMIS source code into object oriented C++ updating the Oracle 9i database. The COBOL to C++ conversion process included transforming embedded Tandem SQL into Oracle SQL for Pro\*C and automated formation of C++ methods from COBOL programs and paragraphs. The target system required a 5-tier web architecture, which was achieved by the automatic creation of distinct classes separating the business code from the data access code and by introducing database and system support API layers integrated through a thread spooler to spawn programs as needed. The business logic and database logic interleaved, but were easily distinguishable because of the C++ method invocations that were defined for operations specific to each of these layers. The transformation of REMIS COBOL85, TAL into C++ was accomplished with 100% automation. TSRI designed a custom "Pooling" system to provide multiple instances of each derived C++ program supporting web-enablement and allowing the system to be scaled for any number of

users. All programs were turned into objects and made "thread safe". Large programs in the business logic layer required pre-initialization to reduce load times. The "pooling" mechanism component improved load times of program instantiation by caching programs in advance for instant user availability and by resetting program data elements to their original state before returning used programs to the pool.

Boost multi-threading libraries were used to achieve cross platform compatibility, necessary because the application layers were distributed onto separate processes and distributed geographically.

**Re-Factoring:** Several forms of re-factoring operations were used to complete the REMIS modernization. Fully automated re-factoring was employed in the consolidation of "identical" code derived from COBOL copybooks and in the detection/elimination of "dead" code. An additional form of automated re-factoring was applied to automatically derive common base classes from classes containing similar methods to superimpose a hierarchical class structure upon the nested classes initially derived by transformation from copybooks or introduced for other program structures. "Similar" code reduction was accomplished by semi-automated re-factoring. Cluster indices were generated to support semi-automated re-factoring operations used by the customer to consolidate similar classes. NGIT engineers participated in defining re-factoring specifications semi-automatically for "similar" data structure consolidation through a remote interface into TSRI's operation center. With VNC allowing remote access to TSRI re-factoring tools across the web, NGIT personnel were presented with clusters of C++ structures that TSRI tools had determined were sufficiently "similar" for consolidation. An NGIT engineer selected candidates for consolidation and specified the names to be used in unifying these similar data structure elements (aliases). The re-factoring specification was generated from this interaction and applied during a re-factoring phase to the transformed REMIS system, automatically consolidating the C++ classes. This reduced the line count in the re-factored code by 30% and demonstrated the ease and effectiveness of semi-automated re-factoring.

**NGIT & TSRI cooperation:** To reduce the effort associated with testing and to facilitate integration with the Java front-end under development by NGIT, a "Matterhorn" User-Interface Emulator running on the Apache web-server interfaced with the original GUI still running on the TANDEM mainframe. This interim mechanism allowed presentation tier components developed by NGIT to be integrated and tested with the business logic and database access layer generated by TSRI's automated modernization processes even though these components were running at sites 2500 miles apart on different schedules. The automated modernization of the business logic layer of REMIS Increment-1 was completed months before the Java GUI redevelopment was completed.

**Results:** TSRI successfully performed the assessment, transformation and re-factoring on the 300,000 line COBOL subsystem of increment-1. TSRI delivered the code in two increments. NGIT was particularly pleased by the smooth delivery of both the transformed and the re-factored versions of the REMIS C++ code. They were also pleased with the unexpectedly strong improvements in performance and maintainability that resulted from the re-factoring process. After a month of acceptance testing only 1 flaw was detected in the 300,000 lines of transformed code. No flaws at all were found in the 200,000 lines of re-factored code. TSRI was selected for Northrop Grumman's "2002 Small Business of the Year Award" as a result of these efforts.