

COBOL - NEA - Administrative Systems

National Endowment for the Arts (NEA)

NEA Administrative Systems

Federal Systems Integration & Management Center (FEDSIM)

WANG VS COBOL to C++

History: The National Endowments for the Arts (NEA) Grants Management System (GMS), Financial Management System (FMIS), and Automated Panel Bank System (APBS) were unique one-of-a-kind 27-year old legacy systems written in Wang VS COBOL, running on a WANG VS Operating System, and using a WANG VS DMS database. Despite their advanced age these NEA systems were high quality systems with no commercial off-the-shelf equivalents. The systems provided administrative support, grant application, financial management, and congressional reporting for the NEA, an agency managing nearly \$200 million in federal money. After WANG filed for bankruptcy in 1992 and was acquired by Getronics in 1999, platform obsolescence and escalating operating costs became major drivers to modernize the NEA's mission critical legacy systems. The only two practical options for modernizing the NEA legacy systems were manual rewrites or an automated transformation. The NEA target platform was determined by a gift to the NEA from the Bill and Melinda Gates foundation of Intel computers and Windows software. Before approaching TSRI, the NEA had contracted with another company, which attempted a manual rewrite. This manual effort lasted 2 years, cost \$1.6 million and ended in total failure. The GSA - Federal Systems Integration & Management Center (FEDSIM) engaged TSRI to provide automated modernization services to the NEA.



Challenge: The NEA's administrative system consisted of three major subsystems FMIS, GMS and APBS totaling 656,000 lines of VS Wang COBOL. These systems used the Indexed Sequential Access Method (ISAM) features of the VS Data Management System (DMS) and made extensive use of VS OS subroutines. The NEA legacy systems included twelve hundred 3270 green screens, roughly 900 data tables with 10,000 data elements, and an annual data volume of 4.5 gigabytes. Automated modernization of the NEA administrative systems was a major technological challenge as well as a project management challenge. TSRI provided modernization as well as system integration and deployment services. The NEA provided acceptance testing, deployment testing, and overall program management. FEDSIM provided contract management services.

Modernization required a seamless transition into a modern platform-independent C++ system using Microsoft SQL Server. The NEA required that all system enhancements and modernizations meet Congressional reporting mandates. Prior to deployment, the new Windows NT C++ system was tested in parallel with the legacy WANG system through end-of month, end-of quarter and end-of-fiscal year processing cycles.

The modernization was undertaken in multiple phases. Phase-1 included the assessment, transformation, implementation and documentation of all three major NEA subsystems. These transformations were accomplished during March and April 2000, with functional testing commencing the second week of May. During this time frame an ODBC interface was introduced into the database tier to provide access to Microsoft SQL Server and the DMS database was converted into a MS SQL Server.

All three subsystems were integrated in June and parallel testing commenced in July 2000. This tight deployment schedule was dictated by the urgent need to replace the Wang system, which was experiencing intermittent hardware failures. The project was further complicated by the need to modify and integrate roughly 40,000 lines of COBOL in the FMIS system. This modification was made in parallel with the modernization project and required retrofitting before parallel testing could commence.

Another significant challenge was the requirement to convert the monolithic mainframe system with 3270 terminals into a 2-tier / thick client system. The transformed applications were required to run as dedicated desktop Windows applications using a centralized MS SQL server database. The business logic, data manipulation and user interface logic of the thick client applications were separated into distinct API layers.

Creation of the user interface layer required automatic generation of Microsoft Foundation Class (MFC) User Interface (UI) from the WANG VS COBOL screen data headers originally designed for a 3270 terminals. This entailed converting the original embedded terminal control processes into an event driven MFC UI application so that separate threads could be introduced for the business process layer and the user interface layers. This allowed the original business logic to be synchronized to almost any form of external user interface. The user interface layer was designed to allow runtime selection of Windows MFC UI, Netscape Browser UI, or the Unix CURSES CRT packages. This allowed device and platform independence for the user interface tier. The MFC UI front-end main thread used native Windows messaging and standard Win32 thread synchronization objects.

Creation of the database interface layer presented additional challenges. The new 2-tier thick client architecture replaced a monolithic mainframe application. Console terminals controlled by a thin data stream and a centralized database collocated with the VS DMS database, were replaced with multiple independent desk top applications with network access to a shared centralized database in the new C++ system. The network traffic to support MS SQL server database access proved to be several orders of magnitude greater than the network traffic of the WANG system. To reduce the impact of the high volume of data transmissions across the network, TSRI re-factored the data access queries to move data access logic from the client to SQL server.

The initial testing in early May went smoothly with two of the three major systems declared ready for parallel testing by the 3rd week of May. Testing was a lengthy process because 1200 screens and an extensive library of test suites needed to be exercised. Early in the testing process the modernized GMS and APBS systems were pronounced clean, but the FMIS system encountered problems. These problems were ultimately traced to flaws unwittingly introduced by another contractor into the final source code update.

During subsequent months of parallel testing the NEA was able to communicate problems to TSRI developers, who were quickly able to resolve bugs by means of an interface layer play/replay capability built into the C++ version of the NEA system. This capability allowed the NEA testers to record a session and send the record of all tester inputs to TSRI when a problem report was posted. TSRI developers could then replay the session to rapidly localize and correct problems.

Phase-2 was a follow-on contract for re-factoring or re-engineering the modern C++ system. This re-factoring provided improvements in several areas; it enhanced performance, simplified future maintenance, and insured that the three administrative systems would work in-concert. In a final follow-on contract, TSRI continued to act as a system integrator, making all NEA requested system enhancements designed to carry the system for the next decade.

Results: The three phases of the NEA modernization project and major upgrade were completed in less than two years. TSRI successfully met all goals and mandates, achieving full implementation of the modernized system on time. Of special note was the fact that midway through parallel testing the NEA encountered a WANG hardware system failure that forced an early switchover to the modernized C++ system. Had TSRI not worked so quickly and with such precision, the entire NEA organization would have been shut down for an unspecified period of time.