U.S. AIR FORCE

U.S. Air Force ILS-S Re-Platform Standard Base Supply System (SBSS) Modernization Case Study

COBOL to Java Documentation, Modernization & Refactoring using Automated AI-Based Tools, Object Management Group (OMG) standards, and Automated Testing & Integration Support.

The Software Revolution, Inc. (TSRI) – 11410 NE 122nd Way Suite 105, Kirkland, WA 98034-6931
Tel: (425) 284-2770 Fax: (425) 284-2785 – www.tsri.com

Executive Case Study

The United States Air Force Integrated Logistics Systems-Supply (ILS-S) is the definition of a mission critical system. It is used by over 18,000 users at 260 U.S. Air Force, Air Force Reserve, and Air National Guard fixed operating bases and forward operating locations, to provide daily supply and equipment support for wartime flying missions. The Air Force needed a low-risk solution to modernize the 50-year old Standard Base Supply System (SBSS, a key portion of the ILS-S system) to drive down operating costs, while retaining all of its functionality.

The ILS-S Software Modification Re-platform (ILS-S SM-RP) was the culmination of 16 years of continuous innovation, collaboration and singular commitment to the success of the program by both the Air Force and the supporting contracting team. This was an exceptional team effort on the part of the Air Force's program office, NTT DATA Services' technical lead, the database conversion team, TSRI's code conversion team, and the ARRAY/NTT DATA Services integration and testing teams. For this effort, the ILS-S Re-platform program executed the right acquisition strategy, selected a team with a proven history of software delivery excellence that offered a development team with the domain knowledge, Agile-infused software development processes, and automated code-conversion tools to provide the Air Force with the highest confidence of successful program delivery and performance.

This case study does not attempt to provide a comprehensive history of the success of this entire program; rather, it focuses on *the role of the automated code conversion tool selected to support the modernization effort*. Follow on, companion case studies will address the other factors that contributed to the program's success.

The Software Revolution, Inc. (TSRI), Array Information Technologies, and NTT DATA Services, the ILS-S incumbent for over 16 years, offered extensive past performance and domain knowledge on similar projects for the Air Force, and were awarded the contract. TSRI used its automated modernization capability to document and transform all 1.3 million lines of COBOL and C code to modern Java code, as well as refactor the modernized code to improve maintainability and performance. The ARRAY team, led by NTT DATA Services' solution architect, Paul Saladna, worked with TSRI to complete testing, integration and deployment of the modernized system and to help manage the customer relationship and project expectations. The transformation achieved functional equivalence, and was transformed with greater than 99% automation, significantly lowering risk and schedule.

The project was considered highly successful by the stakeholders, and as of May 2017 is deployed on-time and without difficulty at initial bases Langley and Fairchild. Full roll out is now proceeding ahead of schedule.

Detailed Case Study

In 2014, The Software Revolution, Inc. (TSRI) was contracted by Array Information Technology, Inc. (ARRAY) under the NETCENTS-2 contract vehicle to help modernize the Integrated Logistics Systems-Supply (ILS-S). The ILS-S is the quintessence of a

mission critical system. It is used by over 20,000 users at 280 U.S. Air Force, Air Force Reserve, and Air National Guard fixed operating bases and forward operating locations, to provide daily supply and equipment support for wartime flying missions. Part of the ILS-S was a modern Java/Oracle® application, but the Standard Base Supply System (SBSS, an aged but key portion of the ILS-S system) consisted of a legacy back-end system running on Unisys® mainframe computers. This latter system provides the core business logic and authoritative data for base-level management of supplies and equipment, ensuring their availability for warfighting missions.
The core requirements defining the goals of this project included that the modernized SBSS application needed to exhibit performance and functionality equivalent to or better than the legacy system. Also, because SBSS is a widely used, critical production system, there was little to no tolerance for code freeze. After 50 years of development and maintenance, the code was in need of comprehensive documentation to elucidate the system's structure and processes.

Team ARRAY started developing the ILS-S SM-RP solution by reviewing options including a total rewrite, a re-architecture solution, a COBOL to Micro Focus COBOL migration solution, and an automated code conversion. The total rewrite and re-architecture solutions failed to meet the program time constraints, had historically low success rates and were too costly. The COBOL to Micro Focus COBOL migration solution was a stopgap measure that failed to reach the Red Hat Enterprise Linux (RHEL)/Java/Oracle architectural future state. Team ARRAY selected the automated code conversion solution because it has the highest probability for success and the lowest cost.

The team’s next step was to select the right code conversion tool. Team ARRAY initially reviewed 12 code conversion tool vendors. In October 2013, Team ARRAY conducted a competition with four down-selected tool vendors: Semantic Designs, Dell, EvolveWare and TSRI. Each vendor submitted a written proposal outlining their technical approach, project schedule, risks/mitigation strategy, total cost, expected level of effort from the government and past performance for similar size/scope/technology projects. Team ARRAY selected TSRI based on the following:

- TSRI’s offered an automated code roll tool capable of converting 95% of the legacy SBSS code, minimizing cost and allowing Team ARRAY to focus on overall conversion quality instead of manual code conversion; it also provided the highest probability of meeting the 25-month schedule.
- TSRI offered an automatically generated blueprint capability that provided direct 1-to-1 COBOL to Java traceability and accountability required to confirm all of the software capability in the “as-as” system would be in the “to-be” system and automatically identify any Unisys operating system gaps as a result.
- For Program Executive Officer (PEO) Air Force Business and Enterprise Systems (BES) Reliability and Maintainability Information System (REMIS), TSRI successfully converted 300,000 COBOL lines of code into object-oriented C++ and Java/JEE and was named “Small Business of the Year” by the prime contractor as a result.

- The TSRI solution had the lowest total price of the tools assessed.

Team ARRAY's goal from the outset was to assist PEO BES successfully launch a high confidence information technology (IT) program. To enhance the probability of success required due diligence and commitment by both government and industry. Collectively, government and industry invested the time and resources to fully understand the requirements, risks, and timelines to ensure the ILS-S Re-platform program was delivered as promised.

TSRI Technology & Past Performance

At the core of TSRI's technology is JANUS Studio®, an artificial intelligence based, model-driven transformation engine. Rather than simply transliterating "source" code to "target" code, the JANUS Studio engine executes a mature

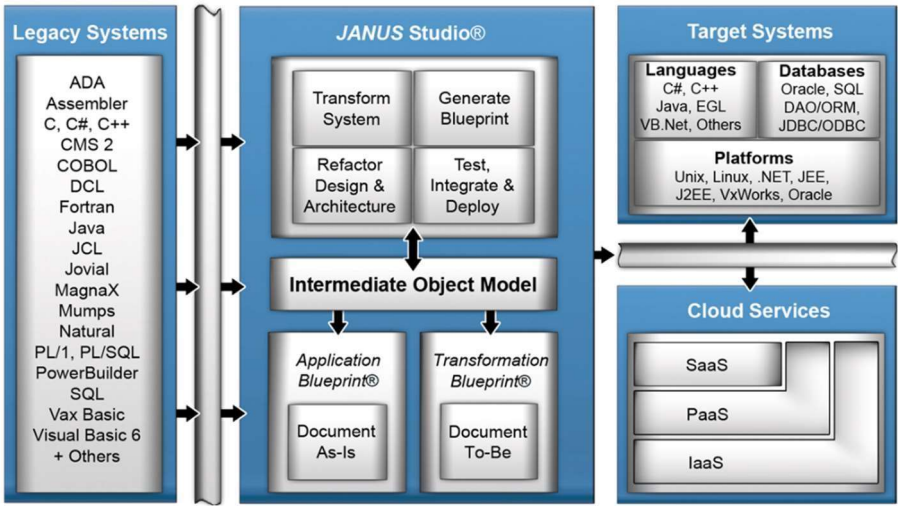


Figure 1: TSRI's JANUS Studio® Toolset

translation/transformation to modern architectures by first constructing a comprehensive model of the legacy system in an intermediate language.
This model-driven approach allows for fully automated refactoring between any practical combination of source and target languages, as well as production of code-level documentation, and automated refactoring of systems.

TSRI's fully automated modernization, documentation & refactoring technology has now been used on over 150 major modernization projects, with every project completed successfully and referenceable.

The final requirement was the production of code-level documentation. TSRI uses the JANUS Studio engine to quickly generate comprehensive UML-based code-level blueprints, the Application Blueprint® and the Transformation Blueprint®. These allow developers to understand both the as-is source code (COBOL, for ILS-S), as well as the transformed target code (in this case Java), in side-by-side hyperlinking format. The artifacts and graphs produced include Control Flow, Data Flow, Cause-Effect, Complexity Analysis, State Transition Tables, and other analysis of the structure and flow of the application.

This approach has now been used in over 150 major modernization projects, including:

- The extraction of business rules for an assessment project on the Air Force CAMS in 2002-2003.
- Modernization of the Air Force REMIS System twice—first in 2004 from COBOL to C++ and again in 2014 from COBOL and C++ to Java.
- Modernization of the Air Force MEMSIZE and NEWSCAN applications from Fortran to C++ in 2007.
- Modernization of the Air Force Weapons System Cost Retrieval System (WSCRS) from COBOL to C++ in 2003.
- Assessment and Transformation of Air Force F-16 Data Entry Cockpit Interface Set (DECIS) Up Front Display System from Jovial to C++.
- Modernization of Air Force Joint Mission Planning System (JMPS) from VB6 to C# in 2012-2013.
- Modernization of the Air Force Weather Data Architecture Capability (WDAC) from Fortran to Java in 2003.
- Modernization of the Air Force Electronic Systems Center Ballistic Missile Early Warning System (BMEWS), an Ada and Fortran to Java and C++ project completed in 2004.
- Modernization of portions of the Air Force Command and Control System-Consolidated (CCS-C) program in 2001.
- Other projects including avionics modernization for the F-16, P-3C Orion, and E-2C aircraft, air traffic control systems, radar and electronics systems, and many others.

SBSS Modernization

In the case of SBSS, the source application was a Unisys/COBOL system comprising 1,260,679 lines of COBOL code and 10,078 lines of C code. The SBSS has been in existence for over 52 years and the Air Force has tried several times to modernize away from Unisys and failed each time due to the seemingly overwhelming complexity of the task. In fact, SBSS modernization was regarded as such a difficult task that it was highlighted in the 2003 book *Modernizing Legacy Systems: Software Technologies, Engineering Processes, and Business Practices*, introduced in chapter 2 as "The Beast".

After SBSS was modeled within JANUS Studio engine (intermediate model construction), TSRI's modernization engineers began an iterative process of applying rules and tuning to the engine to output the transformed system in the target language from the intermediate language, according to the Air Force's specifications. The target, as defined by the customer, was a mid-tier Java environment, including Unix Shell Scripting, Red Hat® Enterprise Linux® operating system, Apache web servers, JBoss, and Oracle RDBMS.

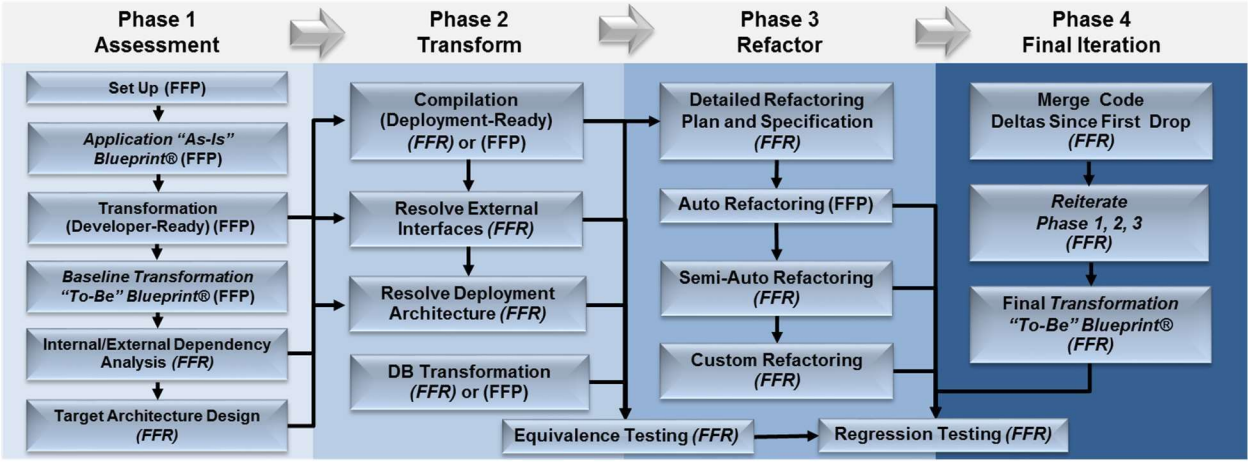


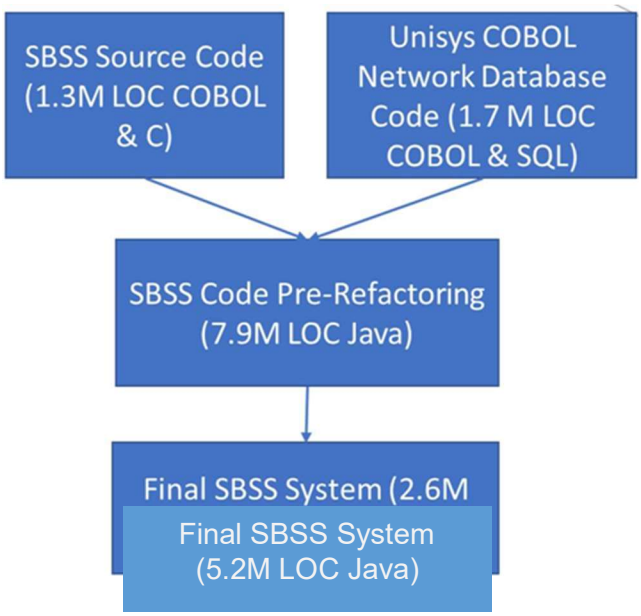
Figure 2: Four-Phase Process Followed on SBSS Modernization

TSRI considers a system like SBSS to be of medium size. As a result, although TSRI can undertake code transformation using a "spiral" development model in which case the code is parceled into units conducive to rapid, progressive delivery, integration and testing, the relatively small size allowed TSRI to regenerate the entire code base with each iteration. With each new improvement, the code was delivered to the ARRAY development team for integration and testing as compiling, integration-ready Java. TSRI's transformation left external calls stubbed out; the ARRAY team completed the work of reintegrating items like schedulers and other utilities, testing the initial spiral of code, and providing TSRI with defects via Bugzilla. In response to each spiral delivery evaluation, TSRI adapted its transformation rules and regenerated improved iterations of the spiral code according to internal evaluation and ARRAY's requests.

TSRI proceeded through spirals sequentially until all 1.3 million lines of source code were transformed.

During the project, the Unisys COBOL network database code was transformed to 1.7 million lines of COBOL with SQL. TSRI then transformed this COBOL/SQL code to a Java target along with the rest of the application code.

The transformation resulted in 7.9 million total lines of Java code—a large expansion from the original 3 million lines of COBOL (1.3 million source lines of code and 1.7 million lines of code from the database transformation). This is a common result for a first pass transformation due to the expansion of copybooks and other cut and paste portions of the application. TSRI used its



proprietary automated refactoring capability to reduce the size of output code through the following refactoring methods:

- Identical Record consolidation
- DAO method consolidation
- Unused definition removal

These techniques, along with the improved method synthesis algorithm mentioned above, brought the code size down to 5.2 million lines of Java.

Application Testing

TSRI and ARRAY also collaborated on application testing of the output product. The NTT DATA Services solution architect chaired daily meetings during the testing phase to coordinate synchronization of the involved teams.

Automated Test Telemetry Insertion allows for rapid isolation of defects and proof of functional equivalence. It is used by many TSRI customers including the U.S. Navy.

TSRI used its automated toolset to inject telemetry into the legacy COBOL system, which was then placed back in production. Telemetry allows the team to capture data put into the system, and data generated back out, so that data can be compared with the same inputs and outputs of the modernized system. When the data streams from the legacy system were compared with data streams from the modernized application, discrepancies were identified by ARRAY and fed back into TSRI's toolset. This automated approach allows for more rapid debugging and proof of functional equivalence. As of today, this automated telemetry approach has been used by TSRI for a number of customers, including the U.S. Navy, and results in relatively efficient and accurate testing as compared to less automated methods.

After testing, the following issues were identified. Often when undertaking transformation work, TSRI's process uncovers bugs latent in the original source code, e.g., the 43 original COBOL and 48 Unisys COBOL/SQL issues listed above. In the entire 1.3 million lines of code, only 62 transformation issues were identified, 1 defect per 21,000 lines of code transformed—this compares very favorably with validated industry standards of 15-50 defects per 1,000 lines of code written manually:

- 290 issues total
- 43 issues with original COBOL
- 48 issues with Unisys COBOL/SQL
- 31 necessitated enhancements
- 102 framework issues
- 4 telemetry related issues
- 62 transformation issues

In the entire 1.3 million lines of code, only 62 transformation issues were identified, 1 defect per 21,000 lines of code transformed.

Lessons Learned

ILS-S's original COBOL took advantage of many features of the COBOL language as yet undefined in JANUS Studio. TSRI's toolset was extended to systematically handle these features by conducting COBOL-language experiments on the original mainframe to arrive at a sophisticated characterization of the behaviors in question. TSRI's frameworks were then updated accordingly.

TSRI's artificial intelligence based toolset improves with each project. Drawing from over 150+ major prior projects, TSRI's toolset only needed minor extensions to support modernization of SBSS.

In the ILS-S codebase, control flow proved to be particularly tricky. The application contained numerous GOTOs jumping out of perform ranges, never to return. To resolve this, TSRI extended its method synthesis algorithms and GOTO elimination strategies. However, this resulted in some code expansion and duplication in the transformed code. Addressing these by-products was not within the project's scope, and didn't have a functional impact, but TSRI has since identified strategies to further improve handling of similarly convoluted control flow patterns, which will be pursued to reduce or completely eliminate the code expansion observed in the current solution.

Results

The SBSS system is now in deployment at select locations and will be fully deployed in the coming months. The project was considered highly successful, and the Air Force was provided an end-to-end solution for a mission-critical legacy application in a very low-risk and efficient development, testing and deployment environment. The TSRI/ARRAY/NTT DATA Services team not only modernized and implemented "The Beast", but also in parallel delivered three major FIAR releases, moved to the cloud, migrated into big data, and embarked on a DoD leading mobile implementation.

The TSRI, ARRAY & NTT DATA team not only modernized and implemented "The Beast", but also in parallel, delivered three major FIAR releases, moved to the cloud, migrated into big data, & embarked on a DoD leading mobile implementation.

Now that TSRI has a JANUS Studio instance tuned specifically to the SBSS target code and architecture, future modernization projects for the Air Force with similar source code and a similar target architecture will enjoy advantages of schedule and effort reduction. The artificial intelligence-based JANUS Studio® toolset learns with each project, and the code transformation rules created will be automatically reused in future projects. This has major implications for other large Air Force systems that are written in COBOL and may be targeting similar Java implementations.

Both ARRAY & NTT DATA Services have also learned how to rapidly take TSRI's outputs and evaluate, test, and provide back information for retransformation. Together, our team is prepared to tackle larger modernization projects, and deliver them successfully.

We are Here to Answer All Your Questions!

You can find more information on our website at www.tsri.com and read through hundreds of our case studies.

Please see our Questions & Answers page to find answers to a lot of common questions.

We are ready to hear from you. Send us a message through our website or directly to information@tsri.com.

Contact us directly at:

Greg Tadlock, Vice President	gtadlock@tsri.com / 425-284-2782
René Wagner, Director	rwagner@tsri.com / 425-284-2800
James Pippin, Director	jpippin@tsri.com / 425-284-2802

Follow us and stay up-to-date on modernization!