G00123793
D. Vecchio

Research Note
18 November 2004

U.S. Navy Benefits From a Composite Application Strategy

Using a model-driven service framework and a composite application strategy, the U.S. Navy significantly reduced the cost and time to manage its contract process.

Core Topic
Application Development: Application Architecture

Key Issues
What methods can be used to manage business rules, business flow and Web services to enable more-agile businesses?

What application development techniques will allow enterprises to implement flexible rules and business flows for efficiency and effectiveness?

The U.S. Navy Facilities Engineering Command (NAVFAC) manages the $6 billion facility contract process for the Navy. This includes contract bid, negotiation, document writing, cost accounting and award in support of facility projects and construction. In addition, NAVFAC manages another $2 billion of facility maintenance efforts. This spending is related to all aspects of real estate, design, construction, Navy housing, base operations, base development and environmental efforts. As with any organization that has used computing throughout the years, NAVFAC has a mix of systems running on different computing platforms.

Problem: Managing the end-to-end contract process for the Navy is significant. In addition to the sheer volume ($6 billion), there are about 2,000 people executing 64,000 contract actions, which total about 6.4 million mostly manual steps. The Navy's goal, like many commercial organizations, was to reduce process time and cost to accomplish its mission and rely more on speed and agility.

Objective: NAVFAC's objective was to provide a lower-cost, more-efficient end-to-end contract management solution across the entire service. The Navy's strategic initiative was to reduce costs through restructuring and transforming this branch of the armed services. Because contract management is such a significant piece of the Navy's budget, it was a prime candidate for cost reduction.

Approach: The Navy's first step was to consolidate a mix of systems to a core set of four key production systems: projects, contracts, construction and maintenance. This step occurred in stages between 1997 and 2004. The second step occurred in two phases. The first phase was to integrate within these core systems, and the second phase was to integrate between them. Until recent efforts to use model-driven service frameworks and

programmatic integration servers, NAVFAC's success in this area had been problematic, with limited success. Before 1997, mainframe integration did not connect to PC systems. During this same time frame, custom Microsoft Windows applications were contemplated, but abandoned due to high cost and risk.

Between 1999 and 2002, the Navy considered an enterprise resource planning solution but found the cost to be significant and the resultant facilities functionality to be inadequate. In 2002, the Navy also considered a broker-based solution. Again, the cost was high and its ability to use its mainframe-based Financial Information System (FIS) was limited, requiring the Navy to replicate significant, complex business logic. Also, a broker-based solution provided an asynchronous solution to a problem that required point-to-point synchronous interfaces. The broker-based solution did nothing to get at the significant local-based contracting data in a variety of spreadsheets.

NAVFAC settled on a composite application strategy that used a model-driven service framework (AppsCreate) from LogicalApps and a service-oriented architecture to extend its core systems. It then developed its integration layer by orchestrating the flows between these core systems. Jacada Fusion significantly helped the Navy's strategy. Although many vendors offer programmatic integration, Jacada elected to include computing platforms not normally addressed by these vendors. Jacada Fusion enabled the Navy to integrate the mainframe FIS and its WIndows-based PowerBuilder Standard Procurement System (SPS) used for contract writing.

The Navy's new implementation uses an Oracle 10G application platform suite to provide overall orchestration of the services. The process is implemented via a nightly batch process but will be enhanced to provide real-time updates. Driven by contract information in an Oracle database, the Java orchestrating application begins by interacting with the SPS application. It uses information extracted from this system to interact with the mainframe-based FIS application. Status information is returned from FIS and used to update the SPS running on a Windows platform.

Results: By orchestrating the independent contract steps in the FIS and SPS, NAVFAC dramatically reduced the manual effort and time in processing contracts. NAVFAC has one interface in production with seven more scheduled for the fourth quarter of 2004.

By automating the manual workflow necessary to manage new contract awards, the contract writing system and contract financial information, NAVFAC has improved the quality of data

and significantly reduced manual effort. Initial efforts have automated approximately half a million manual steps per year. Full implementation will automate 1.2 million manual procurement steps per year. When fully implemented, 83 percent of the manual efforts will be automated. More importantly, NAVFAC has proven the validity of its composite application strategy. This success, although early, has given it credibility within the federal sector. NAVFAC intends to extend this success to more of its core processes, including funds management and invoicing. Finally, this approach may become the template for solving these problems across the broader Department of Defense for other branches of the armed services.

Critical Success Factors/Lessons Learned: Critical Success Factors

- Migrate 2,000 users from more than 100 varying and costly systems to a transactional-capable composite application within budget and schedule. Achieve return on investment (ROI) within six months by turning off more-costly project and contract management systems.
- Leverage core production systems as well as application platform suites, model-driven service framework and programmatic integration servers to develop and prove a viable end-to-end enterprise integration solution to full executive and functional sponsorship.
- Deliver service-oriented architecture (SOA) end-to-end contract integration by November 2004 at Norfolk, Virginia.
- Support significant productivity improvement (dual-entry elimination).
- Achieve ROI within six months of deployment.

The Navy experienced some expensive integration failures during the past several years. Part of the success of this project can be seen in its on-time, on-budget delivery. By leveraging established systems and SOA-development technologies and techniques, NAVFAC delivered on integration where other more-costly initiatives had failed. Providing significant cost saving and achieving ROI within six months are strong value propositions for this approach.

Lessons Learned

- Consolidate, then integrate.
- Lead enterprise integration as a program.
- Build an SOA factory.
- Set expectations for incremental improvement.

By consolidating its application portfolio from about 300 production systems to about four, NAVFAC created a manageable integration project. Application portfolio rationalization is a growing initiative by many organizations to "clean up the IT garage" before implementing a portfolio modernization initiative. By providing an integration competency center for integration, NAVFAC was able to provide a unified vision, develop SOA-skills and use a less-complex set of integration software (programmatic integration servers) to provide quick ROI. Taking an SOA-factory approach provided the right organizational framework to establish a reuse mind-set. NAVFAC was able to create parallel development teams supporting data integration, end-to-end process enablement, legacy service enablement, collaboration and platform components.

NAVFAC's reuse mentality helped provide a near-codeless solution. The operational data store provided self-service reporting to end users. The model-driven service framework provided up to 80 percent code generation, dramatically minimizing coding time. The Jacada Integrator product provided about a 90 percent codeless transactional integration solution for its Windows-based PowerBuilder application and its mainframe-based financial information system. Portal technology provided a common point for integration and collaboration. The Oracle 10G application platform suite team provided a Java 2 Platform, Enterprise Edition framework for services.

Bottom Line: The U.S. Navy Facilities Engineering Command significantly reduced the cost of managing the contract process for more than $6 billion of goods and services for the U.S. Navy around the world. Using a composite application and a service-oriented architecture strategy based on practical programmatic integration, NAVFAC was able to provide dramatic results in six months for less than $1 million. Consistent with our findings at many organizations, smart reuse of systems as building blocks for new composite applications provides great value at less cost and less time than more-dramatic enterprise resource planning and redevelopment solutions.

Acronym Key

FIS	Financial Information System
NAVFAC	U.S. Navy Facilities Engineering Command
ROI	return on investment
SOA	service-oriented architecture
SPS	Standard Procurement System