Paper

# Enterprise integration lessons learned

Anh D. Ta and Scott Starsman

**Abstract—This document describes the lessons learned from a United States Navy enterprise integration initiative called Web Enabled Navy (WEN). WEN was initiated in April 2001 with the foci of integrating navy resources and providing a single-point-of-access, Web environment to all business and operational applications. The navy's applications operate within a complex network environment that spans commands afloat, ashore, and overseas. The challenges addressed are similar to those faced by large, multi-national corporations and include some unusual characteristics including islands of intermittently, bandwidth-limited, connected information consumers. Both technical and management lessons learned will be described and denoted as either prerequisite or success factors.**

***Keywords— web services, service oriented architecture, navy, military, web enablement, Task Force Web.***

## 1. Web Enabled Navy initiative

As the United States Department of Defense (DoD) is going through its *Force Transformation* in response to the Quadrennial Defense Review, the United States Navy is also going through a fundamental *transformation* into an enterprise with integrated information and knowledge resources – *one integrated navy*. In April 2001, an initiative called Web Enabled Navy (WEN) was started to integrate navy resources and provide a single-point-of-access, Web environment to all business and operational applications. The tasking was in two parts: the first issued to the Navy's System Command's to convert applications to web services; the second to Task Force Web (TFWeb) to plan, design, and implement a strategy that would allow the navy to realize the WEN vision [1]. Additionally, TFWeb was tasked to provide vision and guidance to both the ashore and afloat system integration government agencies and contractors.

In striving for the goal of one integrated navy, the WEN initiative must deliver solutions addressing the following issues:

- heterogeneous IT environments,
- multiple network architectures,
- diverse user communities,
- technological volatilities,
- distributed authority,
- multiple parallel IT efforts,
- limited financial resources.

These issues will be elaborated further on.

**Heterogeneous IT environments**. At the start of WEN initiative, it was estimated that there were one hundred thousand applications across the U.S. Navy implemented in a variety of programming languages and technology products. The U.S. Navy IT environment is analogous to a multi-national commercial enterprise. For example, a typical ship might have thousands of applications, hundreds of databases, and vendor unique solutions that didn't support innovation, plus users with diverse needs ranging from administrative activities (e.g., writing a performance review) to mission-critical activities (e.g., to tracking potential threats). Furthermore, most applications are poorly integrated across functional boundaries. While ad hoc solutions to cross-application integration issues have been implemented in some instances, they tend to be slow, inefficient, and required a complex series of agreements between application owners to maintain. Furthermore, as the navy modernizes its fleets, there is a growing dependency on information technologies to monitor and operate the machinery that sustains the ships in their vital missions. Finally, significant authoritative data originates from ships, which needs to be accommodated by the heterogeneous environment. Extending the integration problem at the ship level to the fleet level and further scaling it to the navy enterprise level involving both the afloat and ashore communities describes the scope of the technical challenges faced by the Web Enabled Navy initiative – o*ne navy with connected communities*.

**Multiple network architectures**. The WEN initiative is required to leverage the existing U.S. Navy infrastructures. Currently, there are four major information technology infrastructures. In no particular order, they are the Navy-Marine Corps Intranet (NMCI) infrastructure for the Continental US (CONUS) shore community, the Base Level Information Infrastructure (BLII) for the Outside Continental US (OCONUS) shore community, the Marine Corps Enterprise Network (MCEN), and the Integrated Shipboard Network System (ISNS; commonly referred to as "IT-21") for the afloat community. Each of these infrastructures has a unique constituency, operational characteristics, and programmatic along with business support. At present, combat systems must retain their separate network architectures. The challenge is to integrate these disparate physical infrastructures into a single logical infrastructure – *one enterprise services layer*.

**Diverse user communities**. The targeted U.S. user communities for the WEN initiative include active-duty personnel, reserve personnel, family members, retirees and U.S. Navy contractors. Each user community has different

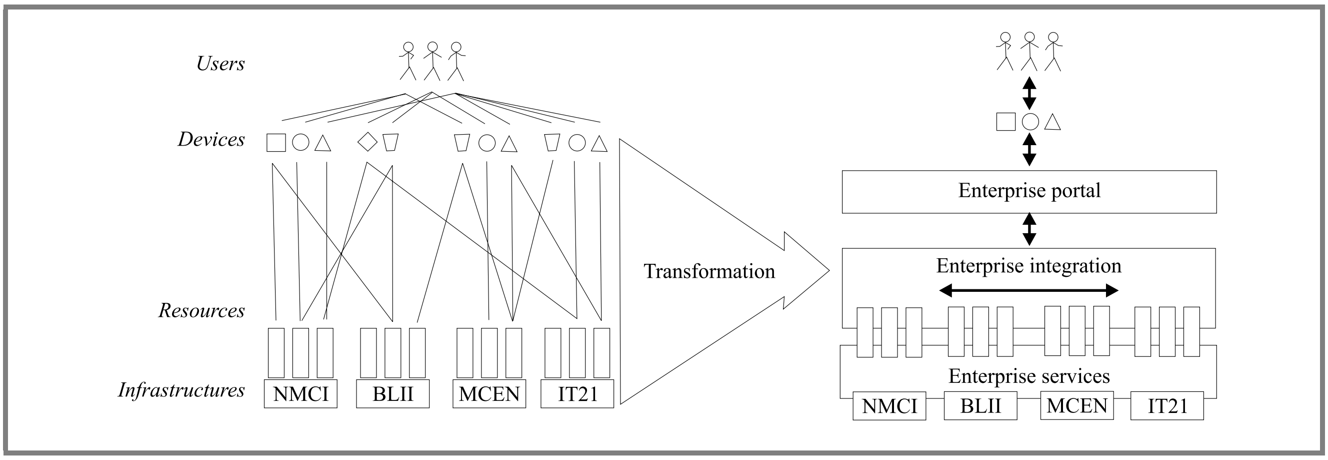


*Fig. 1.* WEN transformation vision.

technical and security requirements matrixed across the various network environments. Additionally, information is shared with key coalition partners during multi-national operations.

**Technology and market volatilities**. At the start of the WEN initiative, the major integration technologies available were middleware and enterprise portal products. While middleware technology has been available for many years, the industry sector lacks open interface standards resulting in numerous vendor-specific and proprietary solutions. As for the enterprise portal sector, the technology was just then emerging into its own identity. As characteristic of an emerging market place, the available portal engines had propriety interfaces and there were numerous mergers of enterprise portal vendors. For a large initiative such as the WEN, the risks associated with a single vendor solution was unacceptable from both technical and acquisition perspectives. Furthermore, a single vendor approach would limit the ability to share and innovate.

**Limited financial resources**. In addition to the technical integration challenges, a large organization such as the U.S. Navy has to seriously consider the financial effects of any initiative because of the vast number of applications in use across the U.S. Navy enterprise. Further complicating the issue, each application was at a different lifecycle stage, supported by varying logistics infrastructures, and with budgets already allocated to tasks spanning a multi-year timeframe. Because the allocated budgets did not contain resources explicitly designated to comply with the WEN requirements, program managers were required to identify existing application development and maintenance funding to support Web enablement.

## 2. Navy enterprise architecture

The Task Force Web team has developed an enterprise architecture and an integration strategy to realize the WEN vision (see Fig. 1). These products address the enterprise integration problem from two dimensions. The *application-to-user* delivery dimension focuses on methods for integrating content across the U.S. Navy into a single-point-of-access, Web environment. The *application-to-application* integration dimension focuses on the integration of U.S. Navy resources within an end-to-end business process perspective.

As an additional constraint on the TFW team, the resulting solution must comply with the architectural guiding principles:

- Maintain a flexible architecture that can accommodate both mission changes and technological changes.
- Implement an architecture consisting of *best-of-breed* components, where possible.
- Implement a standard-based, vendor-neutral architecture that minimizes ripple effects to back-end applications when architecture components are changed and allows innovation at the back end.
- Implement an architecture that minimizes information overload with automated information flow.
- Implement an architecture that enhances “shared awareness” through seamless integration of data sources.
- Implement a cost-effective architecture that maximizes acces to information and services.
- Implement an architecture that leverages XML-based industry standards to abstract from platform and programming language peculiarities.

The success of the WEN initiative depends on a balanced solution for both dimensions.

### 2.1. Prerequisites

A holistic approach was taken that addressed authoritative data sources and interoperable infrastructure issues.

**Authoritative data sources**. The U.S. Navy's approach to the establishment of authoritative data sources is through an organizational structure called the Functional Data Managers (FDMs). Each FDM is responsible for rationalizing the large number of databases into an orthogonal subset that more closely meets the navy's needs within a functional area. The expected benefits are:

1) the reduction of duplication among navy applications and databases,

2) the cost saving from items such as licensing costs, management of change overhead (such as revisions and updates), user training, etc.

A long-term goal is to align functional applications managed by the Functional Application Managers (FAM), an organization analogous to FDM, as a source for composite application frameworks (CAF) [2].

**Interoperable infrastructures**. TFWeb is leveraging the exiting infrastructures (i.e., NMCI, IT-21, BLII, and MCEN) to provide many basic support services required to support WEN functionality. However, some enterprise requirements are so critical that distributed development and/or deployment cannot meet target requirements. Two of these critical enterprise services are global directory services for identify management and enterprise data replication/synchronization providing ubiquitous data presentation.

The integration of the navy's directory services into a unified Navy Global Directory Service (NGDS) is a key enabler for many important capabilities such as personalization, enterprise single sign on, and enterprise rolebased access control. To date, there have been few standards that support cross-domain authentication. The Security Assertion Markup Language (SAML) standard promises to begin addressing this issue and is seen as a critical standard in building a true enterprise-wide single-sign on solution. These traditional infrastructure issues will have an unusually large effect on the user experience.

Currently, the data replication and synchronization strategy is predominantly being addressed through existing file-based replication solutions such as Collaboration at Sea (CAS). A more comprehensive solution providing true data synchronization between Relational Database Systems is planned but has yet to be implemented.

### *2.2. Vendor-neutral portal architecture*

Given the challenges described and the recognition of the rapid rates of technological changes, Task Force Web focuses on bringing to bear multiple parallel efforts by government and vendors to create the desired environment. When possible we select best of breed recognizing that speed and facilitating vendor neutrality are key elements to be considered. The Navy Enterprise Portal (NEP) architecture has the following key characteristics:

- Presentation layer:
  - a standard-based, component based, modular enterprise portal architecture to provide U.S. Navy users with a single-point-of-access, Web environment that supports and integrates both the afloat and the ashore communities;
  - XML-centric design to support multiple display formats (e.g., HTML, Wireless Markup Language – WML, VoiceXML).
- Portal abstraction layer:
  - a portal connector component to provide an abstraction between the portal and content providers that minimizes ripple effects to content providers from replacement of the portal;
  - an external portlet registry component to provide a registry that further enhances independent from the portal.
- Business logic layer and data store layer:
  - XML-centric design to support capturing metadata and flexible customization (e.g., CSS, XSD, XSL);
  - an XML Web services integration strategy to provide a flexible environment for *composite application* leveraging resources across the heterogeneous IT environments.

The portal architecture focuses on standards-based interfaces, which will enable the U.S. Navy to develop its enterprise architecture and Web-enabled systems incrementally along with maintaining vendor neutrality. For areas where an industry standard is not available, TFW has developed an abstraction layer that provides vendor neutrality. For example, in the absence of a portlet API standard (e.g., JSR-168 and WRSP), TFWeb developed a "portal connector" component that provides an abstraction between the portal and content producers' applications. The portal connector satisfies several key requirements:

- It provides URL rewrite to ensure the seamless access of resources across the navy, where many of the resources are behind firewall with strict security filtering rules (e.g., IP filtering at the proxy server).
- It invokes an XSL engine to provide a consistent, server-based XSL transformation.
- It provides access to the portal context information such as user name and selected style template. This information is similar to the information defined in emerging portlet standards such as Web Services for Remote Portals (WSRP) and Java Portlet Specification JSR-168.

As the result of having a portal connector component, the navy enterprise architecture can integrate any portal with minimal ripple effects to content producers' applications and services.

### *2.3. Pragmatic integration strategy*

Redesigning all U.S. Navy applications as Web applications is an imposing task in the near term given financial, operational, and organizational constraints. TFWeb encourages content producers to either develop their application from the beginning as Web application with Web services interface or to provide a Web services "wrapper" for key functionality of the existing application. This Web services-enablement of applications is referred to as a TFWeb *content integrated* application (see Fig. 2). Via the Web services interface, the returned data and metadata should be formatted in XML along with the associated eXtensible Stylesheet Language (XSL) stylesheet specifying the default display format (e.g., an HTML XSL stylesheet for desktop Web browsers or a WML XSL stylesheet for mobile Web browsers). The associated content provider can be referenced/accessed either through a URL over HTTP protocol (e.g., REST style Web services) or as Web services over SOAP/HTTP protocol. Display content is rendered by transforming the XML data using the provided XSL stylesheet.

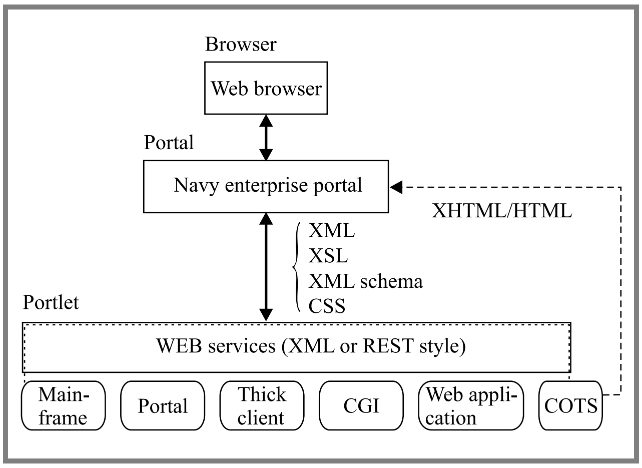


*Fig. 2.* Content integration requirements.

The expected benefit from TFWeb's content integration approach is a foundation for a Service-Oriented Architecture (SOA) with *composite applications* based on Web services. In addition, the approach focuses on an Enterprise Application Integration (EAI) strategy that allows leveraging of existing investments through existing Web standards.

For Commercial off the Shelf (COTS) products, the integration requirement is content integration. To support application-toapplication integration, the vendor shall provide supporting data showing how its Web services interface can be used to achieve the equivalent capabilities as the "out-of-the-box" HTML interface. The granularity of the Web services interface should correspond to the granularity of business logic codified in the product. If a COTS product does not have a Web services interface then an estimate of the level of effort required to develop such an interface (e.g., using a vendor toolkit) must be provided. "Out-of-the-box" HTML interfaces can be integrated into the NEP via URL references (i.e., *reference integration*). However, the HTML interface needs to be tested in the NEP because some HTML constructs that work in a stand alone Web browser mode (i.e., accessing the COTS' HTML interface directly) will not work within a portal environment (e.g., dynamically generated relative URL, frameset, etc.).

## 3. Lessons learned

### *3.1. Management*

This section presents organizational and process lessons learned from the management perspective.

**Identify and overcome social resistance**. In general, management commitment on all IT initiatives is important. However, management commitment is essential for *enterprise-wide* IT initiatives. Voluntary compliance with strategic business changes will generally not produce the speed or depth or change required. Program managers have previously defined assumptions, program requirements, schedule, and budget. Furthermore, the Web services architecture fundamentally changes the business model within which program managers work. The reason many of our systems are built as vertical stovepipes is that it is easier to design, deploy, justify, and support a system that is entirely controlled by a single program manager. The complexity of operations and synchronization of business data are transferred to the operational user or decision maker. Interoperability woes are suffered by virtually every user of these systems. Web services technology breaks this stovepipe approach by forcing a separation of data, business logic, and presentation and the accompanying decomposition of customary monolithic applications.

The TFWeb team has commitment and support at two levels. Senior navy leadership consisting of flag officers and Senior Executive Service (SES) civilians have all been given and acknowledged the mandate from the Chief of Naval Operations (CNO) that all navy applications be Web-enabled. The other layer demonstrating support and commitment for the enterprise Web enabling effort is the layer of technicians, engineers, and programmers who implicitly understand the technical drivers for Web enablement. Indeed, this layer did not need to be "sold" on the value of this initiative as they've been battling with the consequences of stovepiped systems for years. The most challenging layer to convince has been the middle layer of program managers who have specific program and budget goals and have incentives to continue the existing system of funding and development.

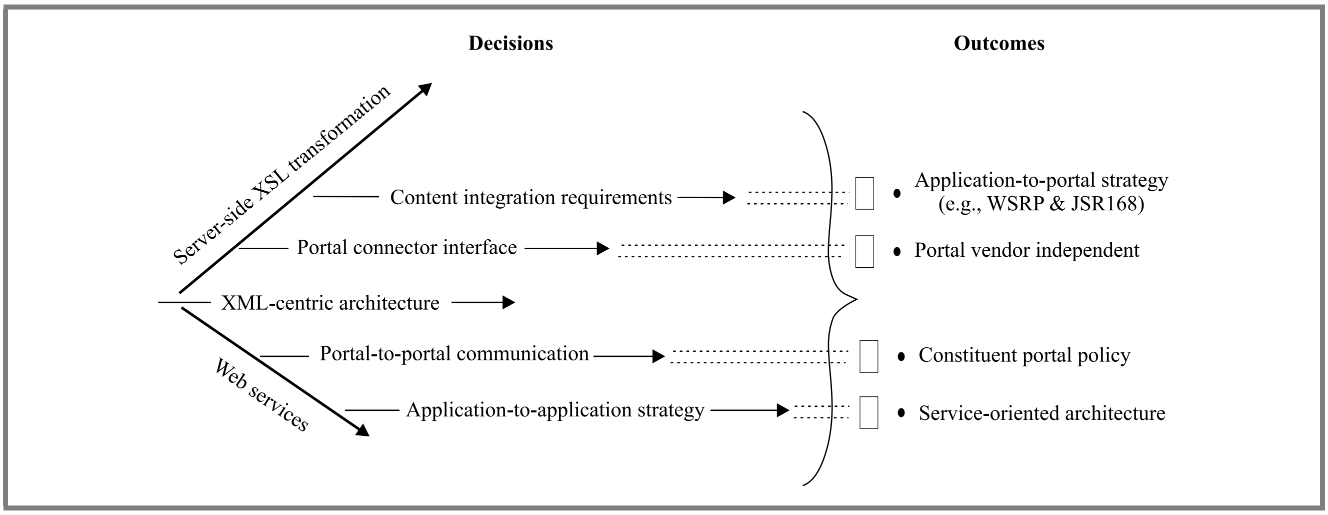

Decisions
Outcomes
Server-side XSL transformation
Content integration requirements
Portal connector interface
XML-centric architecture
Portal-to-portal communication
Application-to-application strategy
Web services
Application-to-portal strategy (e.g., WSRP & JSR168)
Portal vendor independent
Constituent portal policy
Service-oriented architecture

*Fig. 3.* Architectural decisions.

**Simplify integration**. A majority of the resistance to any enterprise-wide IT initiative is motivated by the perceived complexity and level of effort required to comply. This resistance is compounded if the developers and/or program managers perceive that the standards to which they must develop is in flux or is incomplete.

TFWeb began with a very aggressive schedule and released a pilot portal and development documentation within 180 days of its establishment. Unfortunately, this resulted in the external developers bearing the brunt of the learning experience as the Web-enabled environment took shape. By the time the NEP architecture had stabilized, many developers had become wary of the changing development standards and continued development at a much more cautious pace.

In order to overcome this problem, the architecture should be developed and deployed to a very limited audience consisting largely of developers eager to make the system work. Their feedback should be incorporated into design revisions. External developers should not be included until the design has stabilized to the point where backward compatibility is reasonably assured as the architecture moves forward. Future architecture releases should be publicly scheduled far in advance and adequate backward compatibility must be ensured over a reasonable period of time.

Once the design has stabilized, a set of small, completed examples should be developed and distributed. “Learning through examples” is a popular learning approach among software developers. In addition to distributing examples, the process of compliance should be simplified.

**Identify and integrate key enterprise content/services**. The first production version of the NEP was deployed as a piece of the infrastructure where content was to be provided by the existing application owners. Initial acceptance of the NEP instance would have been accelerated had it initially included a set of tools in common use across the enterprise. In this case, basic collaborative tools such as chat, message boards, content management, and white boarding are excellent examples of enterprise content that proved compelling to our user base.

The initial non-availability of these services arises from the fact that the navy enterprise does not acquire and deploy applications in an enterprise fashion. Instead, each navy program supporting a specific user community will purchase implementations of these functions. This has yielded a costly, duplicative, and noninteroperable suite of tools that were difficult to individually integrate into the NEP. Ultimately, TFWeb elected to use some of the broadest existing licenses to deploy an initial capability and address the need for enterprise procurement to fulfill longer-term requirements.

### 3.2. Technical

This section presents architecture and/or integration and content lessons learned from the technical perspective.

In reviewing the progress of TFWeb, there were several architectural decisions that were fundamental to the quality of the current WEN architectural. Figure 3 illustrates the important architectural decisions and their outcomes. First, the recognition of the importance of XML as an interface technology among the U.S. Navy’s heterogeneous IT environments laid the foundation for adopting enterprise focused technology such as Web services. Second, the decision to adopt server-side XSL transformation led to the development of the portal connector component that provides an abstraction interface between the portal engine and the back-end applications. In addition, the server-side XSL transformation decision led to the emphasis on integrating content as XML formatted data. This emphasis is the motivation for the TFWeb’s content integration requirements. The TFWeb’s content integration requirements are the foundation of the application-to-application integration strategy allowing data to be accessed without requiring parsing through HTML or propriety formats. Overall, the emphasis of Web services in the TFWeb's content integration requirements provide a simple yet powerful solution to issues such as portal-to-portal communication, where portlet content can be syndicated to other portals (e.g., syndicating portlets content from constituent portals to the enterprise portal). (*NOTE. It is important to note that while emerging portlet standards such as WSRP and JSR-168 provide a standard interface for integrating content into portal addressing TFWeb's application-toportal integration dimension, they do not solve the TFWeb's application-to-application integration dimension because their interface can only return data embedded in HTML fragments for display. Thus, if the content consumers want the raw data, an HTML parser is required. As for TFWeb's content integration approach, the data is always available in XML.*)

**Develop architectural guiding principles**. Architectural guiding principles convey the criteria, aligned with program vision or requirements, for selecting candidate solutions or technology products. This aspect is essential in an IT initiative that involves many organizations with often-conflicting objectives.

**Develop a technology roadmap**. In general, a technology roadmap document promotes an *active* technology management approach that helps organizations identify and track candidate technologies for adoption. Specifically, a technology roadmap is the best place to provide the rationale for selecting a candidate technology/product because it contains comparisons at both the technology and technology product levels (e.g., identifying dead-end technology). With a technology roadmap and its taxonomy of technology, organizations can quickly evaluate emerging technologies or available technology products. In addition, proof-of-concept prototypes should be performed to gain insights into potential implementation risks. A technology roadmap document should be developed as a companion document to an architecture document. While a technology roadmap captures the rationale for adopting technologies, an architecture document mainly focuses on describing the architecture with emphasis on describing the interactions and functionalities of architectural components.

**Emphasize server-side implementation**. In any enterprise-wide deployment, the mandate of one client device is not realistic. Hence, server-side implementations of candidate technology should be considered to facilitate the rapid technology insertion while avoiding software distribution problem such as Web browser upgrade. An example is the adoption of XSL technology. Through prototypes, the TFWeb team discovered that XSL client-side processing is only available starting with Microsoft Internet Explorer 5.5. In addition, there were several server-side XSL engines with different interfaces. To ensure consistent XSL engine interface and behavior, TFWeb team adopted a specific server-side XSL engine.

**Emphasize portal abstraction**. The enterprise portal market is still evolving. In fact, many of the risks that TFWeb team faced during the beginning of WEN initiative are still valid (i.e., market volatility). TFWeb developed two architectural components to provide an abstraction around the portal engine to minimize ripple effects when the portal engine is replaced. The first component is the portal connector. Its function (e.g., portlet context info) is similar to the set of interfaces being defined in emerging portlet standards such as WSRP and JSR-168. The other component is an external portlet registry. The original intention for the external portlet registry was primarily management of portlet information, where the portlet many not be Web services- based. However, as the NEP architecture evolved to adopt Web services the external portlet registry has evolved to become a Web services registry that complies with the Universal Description, Discovery and Integration (UDDI) standard.

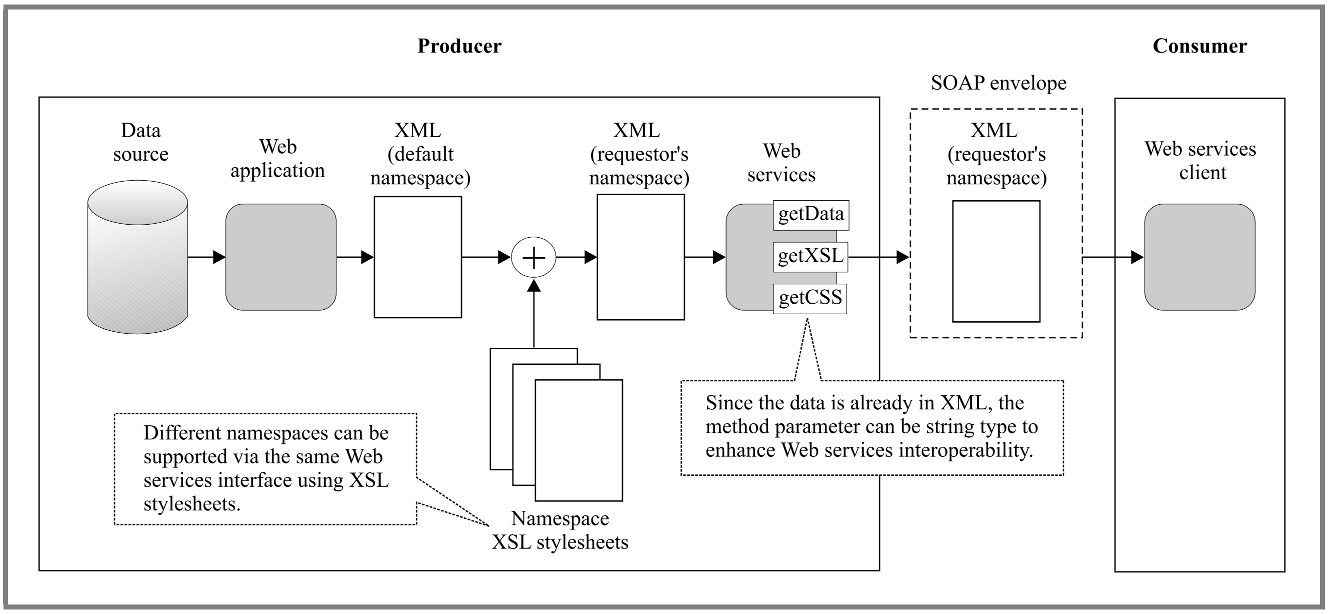
In the striving for maintaining an abstraction layer between the portal and the content providers, NEP use of the portal has been limited to just presenting the portlets' content. A desirable goal is to access more of the portal's functionalities (e.g., virtual community, federated search) via a standard interface.

**Develop a comprehensive content integration strategy**. The major challenge in any integration effort is implementing a common protocol and data format for disparate systems to communicate. Figure 4 illustrates the design of a Web services complying with the TFWeb's content integration strategy. TFWeb's approach of using XML to format data along with metadata and document-centric Web services as the common protocol provide a flexible foundation for integration with some unexpected benefits:

- XML-enabled database. Most available database products provide support for formatting result set data in XML, thus, providing content producers an efficient method to produce XML-formatted data.

- Web services interoperability. Since TFWeb adopted an XML-centric design before adopting Web services, most of the content providers were already producing XML-formatted data. As a result, their migration to a Web services interface was easy. Specifically, the Web services method parameter for accessing the data can be string type because the data is already in XML. The use of string type eliminates a common Web services interoperability problem between different Web services implementation. In fact, it is consistent with the recommendations in WS-I basic profile [3].

- Multiple XML namespace data. One of the profound benefits of the TFWeb's content integration strategy is the ability to produce the same data for different XML namespaces. Although most databases can format the result set into XML, the XML tag names are often mapped to the database column names. One can build custom Web services serializer to format the XML tag but this effort is not trivial. A recom-



Different namespaces can be supported via the same Web services interface using XSL stylesheets.

Since the data is already in XML, the method parameter can be string type to enhance Web services interoperability.

*Fig. 4.* Benefits of content integration requirements.

mended approach is to apply an XSL transformation to the generated data in order to obtain XML-formatted data complying with the requester's XML namespace.

- Support both integration dimensions. The approach of using Web services offers a solution that simultaneously addresses both the *application-to-portal* (i.e., vertical integration – integrated presentation) and *application-to-application* (i.e., horizontal integration – automated business process) integration dimensions. For example, portlet content in a constituent portal can be syndicated to an enterprise portal using Web services. As the result, the portlet content is available at both the enterprise and local levels. In addition, the presentation of the portlet content can be easily transformed to comply with the enterprise "look and feel" because the data is already in XML.

**Emphasize XML-based formats for content**. Given that the main objectives of the WEN initiative are to provide access to all resources and to support application-to-application integration, content formatted in using an XML standard (e.g., Rich Site Summary – RSS) is more efficient to process and repurpose for different presentation devices. In addition, XML-formatted data when use with compression produced bandwidth efficient content.

**Emphasize bandwidth efficient content**. Communication bandwidth is always a resource in demand and increasing bandwidth for the whole enterprise is often financially infeasible. Hence, content producers should be encouraged to review their implementations with an emphasis on reducing file and storage size through the use alternative formats. For example, the width and height attributes of an HTML < IMG/ > tag should not be used to present a small image of a picture. Instead, the HTML < IMG/ > tag should reference a file of an image that was reduced using an image editor. Beyond the scope of HTML, graphical content can be represented more efficient in terms of file size and display quality using new vector-based graphics standards such as Scalable Vector Graphics (SVG) and Extensible 3D (X3D) Graphics.

TFW has developed several proof-of-concepts demonstrating that using vector-based graphic formats with compression can dramatically reduce file size of graphical content while not compromising the content quality. Furthermore, it was discovered that the combination of JavaScript, XML-based formats (e.g., SVG, X3D), and Synchronized Multimedia Integration Language (SMIL) support offline content manipulation via the XML Document Object Model (DOM) interface and only require burst communication with the server.

## 4. Summary

Enterprise-wide IT initiative requires a flexible, innovative, holistic solution that addresses both the technical and management perspectives. Furthermore, the candidate solutions have to be evaluated in the larger enterprise context to avoid selecting local-optimized solutions. While the TFWeb team has learned many lessons and solved many problems, there are still some remaining challenges:

– enhance identify management with emphasis on single sign on capability;

– enhance portal personalization with an enterprise-wide replication and synchronization strategy;

– increase *core enterprise services* to reduce duplication in implementations among content providers;

- ensure common “portlet behavior” in addition to common “look and feel” at the enterprise presentation level;
- increase adoption and usage of metadata across the enterprise through the use of ontologies.

One remaining challenge for WEN is a typical struggle for IT initiatives; it is the struggle between developing a long-term, enterprise-wide solution and developing a solution that produces results immediately whether or not the solution is consistent with the long-term, enterprise-wide direction. From the beginning, TFWeb has chosen to focus on developing a long-term vision emphasizing flexibility and robustness of the architecture over portal content. While a vendor-specific stovepipe approach would have been much more rapidly fielded, TFWeb opted to deal with interoperability issues proactively rather than reactively. I hope you will remember Aesop’s fable of the hare and tortoise when you consider the results of this effort. We may be viewed as the tortoise in some minds but our version of a tortoise is scalable and reliable while moving at the Internet speed.

## References

[1] Task Force Web, Web Enabled Navy Architecture (Forward), Version 2.0, 31 Jan. 2001.

[2] Aberdeen Group, “Using Composite Applications to Lower Integration Costs”, Executive White Paper, Apr. 2003.

[3] K. Ballinger *et al.*, “Basic Profile”, Version 1.0a, Web Services Interoperability Organization (WS-I), 8 August 2003, http://www.ws-i.org/Profiles/Basic/2003-08/BasicProfile-1.0a.htm (accessed 29 August 2003).

**Anh D. Ta** is a technical manager with consulting experiences with executive management, software system development, and enterprise integration initiatives. In addition, he has taught software engineering and computer science courses at several universities. Dr. Ta has a doctorate in Information Technology from George Mason University.
e-mail: ata@mitre.org
MITRE Corporation
McLean, Virginia, USA

**Scott Starsman** has served in a variety of operational and command and control positions in the U.S. Navy for 17 years. He has been responsible for the development, deployment, maintenance, and operation of satellite communication systems, advanced networking systems, and complex C2 systems. He has a doctorate in electrical engineering from Old Dominion University.
e-mail: scott.starsman@navy.mil
United States Navy
Norfolk, Virginia, USA