Successful Product Line Development and Sustainment: A DoD Case Study

Sholom Cohen, Software Engineering Institute
Ed Dunn, Naval Undersea Warfare Center
Albert Soule, Software Engineering Institute

September 2002

Product Line Systems

Technical Note
CMU/SEI-2002-TN-018

The Software Engineering Institute is a federally funded research and development center sponsored by the U.S. Department of Defense.

For information about purchasing paper copies of SEI reports, please visit the publications portion of our Web site (http://www.sei.cmu.edu/publications/pubweb.html).

Contents

List of Figures

List of Tables

Executive Summary

The Department of Defense (DoD) supports several dozen specialized range facilities, including those to test and evaluate systems acquired for our military forces (Test and Evaluation [T&E] ranges) and facilities to maximize force readiness (Training ranges). A range is composed of the set of resources and physical assets required to conduct a specific test or training exercise. The instrumentation systems at these facilities share several common characteristics, including

- sensors and sensor systems to acquire data
- processors to manipulate and log data
- filters to sort and sift desired data
- display, analysis, and reporting systems to make information available to range personnel and range customers

Traditionally, these instrumentation systems have been built for specific categories of weapons systems and missions. Some previous attempts to capitalize on the commonality across range systems have achieved minimum levels of success. None of the efforts to build common software have achieved widespread adoption in major range instrumentation systems.

The Engineering, Test, and Evaluation Department of the Naval Undersea Warfare Center, Division – Newport (NUWC) develops and supports a dozen or more systems at range facilities around the world. Over the past decade, these ranges as well as similar DoD facilities have undergone significant hardware upgrades, from proprietary displays and processors to commercial off-the shelf (COTS) hardware components, yet for the most part, the software remains unique to each site. A second-generation of hardware upgrades is either underway or being planned at many of these facilities.

NUWC has pursued a very steady path toward a product line to address these upgrades for major range software systems. A set of assets called RangeWare provides the architecture and components to cover capabilities in sensor, host, and display processing. The systems generally address the T&E and training missions at these major ranges. Currently, three systems from the product line are in operation at the Atlantic Undersea Test and Evaluation Center (AUTEC). Systems in various stages of planning will definitely use the RangeWare architecture and components. There are another 5-8 programs that are potential product line candidates.

NUWC performs essentially all of its range software development in house, so it is not the typical DoD acquisition organization. In fact, the NUWC development group more closely resembles commercial organizations having business units for product development and sustainment. DoD ranges with acquisition programs needing new facilities or upgrades to existing facilities acquire systems from NUWC. A small asset group at NUWC assures the fidelity of the product line. Development groups build new systems contributing new or improved assets to the RangeWare asset base as a by-product of their development. This asset base addresses the domain-specific needs of NUWC's DoD customers, including

- customer ranges for Test and Evaluation (T&E) and Training
- customer requirements for range modernization

RangeWare is a growing asset base of platform-independent common software to replace the plethora of software currently running on range hardware. After the pilot applications of RangeWare at AUTEC, NUWC is taking RangeWare into a sustainment phase, expanding the coverage of the asset base in terms of new object and information distribution services as well as using the assets in new systems. NUWC is also looking for better ways to support its customers.

Since the assets are intended exclusively for the T&E and training range mission area, they support a product line of T&E/Training mission applications. In this context the term "range" is interpreted broadly. Although the primary target population for RangeWare is open air ranges (OARs), "range" can also include Hardware-in-the-Loop (HITLs), Installed System Test Facilities (ISTFs), Measurement Facilities (MFs), and selected Simulations.

NUWC has recognized both tangible and intangible benefits from use of RangeWare. Cost of building software for ranges is at least 50% lower using RangeWare. Development time has also been cut from years to months for several applications. Total personnel may be cut by 75%. As new programs become RangeWare users, the asset base grows and increases in capabilities to satisfy the requirements of the new programs. RangeWare can then deliver range systems to an even greater potential audience. Initial funding for RangeWare came from the Central Test and Evaluation Investment Program (CTEIP) Foundation Initiative and from funds of the original RangeWare user. This funding has paid for itself many times over in terms of savings from systems already fielded and those currently underway.

NUWC also derives less tangible benefits from greater customer and developer satisfaction. Customers are beginning to recognize the value of RangeWare in satisfying their requirements reliably and predictably. Many NUWC customers prefer the ownership rights they possess with RangeWare as government developed software. As the number of users increases, RangeWare can virtually sell itself to potential customers. NUWC's staff has the time to invest in the challenges that come from new acquisition programs – they are no longer engaged in rebuilding the same capabilities as those on previous programs. This has the salutary effect of having engineers constantly working on new and challenging elements in a program and on new ways to apply and enhance RangeWare. Engineers can easily move between programs because they are immediately knowledgeable of the design process and do not require retraining.

This report provides a case study of NUWC's product line development and sustainment effort, supporting the ongoing transition from initial asset base development and use to a fully supported product line. The report describes the RangeWare asset base and the organizational drivers that led to its development. It also documents the use of RangeWare in building the current operational systems in the product line as well as plans for future products. The report discusses product line sustainment by focusing on several product line practice areas [Clements 02] including

- Structuring the Organization
- Requirements Engineering
- Software System Integration
- Testing
- Configuration Management
- Operations and Tool Support
- Data Collection, Metrics, and Tracking
- Building a Business Case
- Funding
- Customer Interface Management

NUWC's success with RangeWare can provide lessons to other DoD organizations considering adoption of a product line approach:

- Plan the effort to deliver immediate benefits. This will secure and maintain management buy-in.
- Build on existing relationships. Work with real programs and let them contribute to the asset base.
- Establish clear business goals and architectural drivers. Address these drivers in the implementation.
- Define the goals for routine operations. Define and support the supplier-customer relationship.
- Even the earliest product line applications should show mature user interfaces, not merely showcase the underlying technology.

Abstract

The Engineering, Test, and Evaluation Department of the Naval Undersea Warfare Center – Division Newport (NUWC) has developed a software product line asset base, named RangeWare, to support test range operations. NUWC has also fielded a product line of range systems using the asset base. RangeWare provides an object services architecture to support integration of sensor and other range data for analysis and display by range equipment. After several pilot applications of RangeWare, NUWC is now taking RangeWare into a sustainment phase, expanding the coverage of the asset base in terms of object and distribution services as well as applying the assets to new systems.

This case study describes RangeWare and NUWC’s product line practices to sustain and support the evolution of RangeWare. These practices include “Operations,” “Data Collection,” “Metrics and Tracking,” “Software System Integration,” “Configuration Management,” “Tool Support,” “Structuring the Organization,” “Building a Business Case,” and others. The case study also examines NUWC’s lessons learned and its plans for improved process definition for RangeWare product production.

1 NUWC's Product Line Approach

1.1 Origins of the Product Line for Test & Evaluation / Training Ranges

The Department of Defense (DoD) supports several dozen specialized range facilities, including those to test and evaluate systems acquired for our military forces (Test and Evaluation [T&E] ranges) and facilities to maximize force readiness (Training ranges). The Naval Undersea Warfare Center – Division Newport (NUWC) is the Navy's center for developing and fielding systems for the Navy's T&E and training ranges. These ranges encompass a number of mission areas including live open air ranges, hardware-in-the-loop simulations, installed system test facilities, measurement facilities, and simulations.

Over the past five years, NUWC has pursued a very steady path toward a product line for major range software systems. The RangeWare asset base developed at NUWC represents the underpinning of a software product line for Navy range systems. These systems include resources such as sensors, simulators/stimulators, radar devices, analyzers and other equipment. A DoD test or training exercise will integrate a mix of these devices with actual weapon systems and the personnel who operate them. For live test ranges that incorporate operational weapon systems, actual real estate of the range (terrain, ocean surface or subsurface) plays a role in the exercise. A typical exercise may include firing a test torpedo at a drone undersea vehicle and tracking the torpedo through use of a hydrophone array.

The evolution to a product line approach emerged over a decade long period. During this period, the T&E and training communities recognized and began to realize the advantages of sharing software across ranges. As early as 1990, the Office of the Secretary of Defense (OSD) established the Central Test and Evaluation Investment Program (CTEIP) to provide a coordinated process for funding T&E investments that leverage Service investments and encourage joint development and use of new test capabilities. Specifically, the CTEIP charter is to

- solve T&E range capability shortfalls and use test assets of all Services efficiently
- achieve consistency, commonality, and interoperability in test instrumentation, targets, and threat simulators
- develop and exploit modeling and simulation as accredited test resources to support the acquisition process
- develop mobile test instrumentation as an alternative to fixed facilities

- expand and maintain the T&E technology base through prudent investments in emerging technologies

CTEIP investments most often support consistency, commonality, and interoperability across ranges, primarily in terms of sensor and other hardware devices. Under CTEIP sponsorship, ranges participating in these technology programs saw the creation of such systems such as the following:

Advanced Range Telemetry
Hardened Subminiature Telemetry and Sensor System
Translated GPS Range System
Next Generation Instrumentation Bus
Smart Munitions Test Suite
Common Airborne Instrumentation System
Weapons Modification and Simulation Capability
Aerial Cable Test Capability

The past five years have seen the growth of interest in systematic software reuse. Until then, each range perceived its own software needs as unique—needs for special displays, telemetry, processing, and analysis. Such innovations as the High Level Architecture (HLA) from the modeling and simulation community spurred the interest in common software. The range community recognized that many domains within their mission were not unique and could be addressed with common software. The community began to focus on architecture-based solutions, especially architecture-based software capabilities. By architecture-based, we mean the software structures that encompass the totality of software for a range—that is, the software architecture of the range system, rather than isolated components.

The first comprehensive effort to create an architecture for range systems fell under an effort called the Test and Training Enabling Architecture (TENA). The TENA project, led by NUWC, was created as a direct response to the efforts of the Common Test and Training Range Architecture (CTTRA) group—an ad hoc group of test and training range professionals. CTTRA highlighted the need and high-level requirements for a common range architecture. The TENA effort envisioned a common architecture, supporting components and interconnection mechanisms to support local range activities as well as inter-range exercises. TENA was also designed for use at other test community resources including installed system test facilities, hardware-in-the-loop facilities, measurement facilities, and simulations. TENA enjoyed the benefit of lessons learned from earlier efforts that tried unsuccessfully to mandate a common range interface specification and/or network connections. Allowing individual range flexibility within a common architecture was a key design goal.

The TENA project produced a specification for an enabling architecture in 1997. The products of this specification included the following:

- Requirements
- Technical Reference Architecture Definition
- Object Model and Application Programmer’s Interface
- Process for Planning and Executing Tests
- Transition Plan

The architecture was designed from the beginning to support (if not require) a product line approach for developing and deploying software systems for the mission area.

In 1998, CTEIP combined several independent projects into one effort, the Foundation Initiative (FI), to refine and implement portions of TENA. NUWC built a concept demonstration of a proposed application program interface (API) (originally named IKE 1) by leveraging the efforts of a near-term Navy range upgrade and FI funding. This effort became a key component in NUWC's effort to support its immediate customer needs. As an improvement and extension to IKE 1, NUWC developed the first release of RangeWare—a product line asset base for range systems. RangeWare includes applications used to build range systems, frameworks for building other applications, an API to create and manipulate RangeWare objects, and data interfaces.

Acquisition programs at DoD ranges acquire systems generally to address T&E and training missions at these major ranges. RangeWare assets include the architecture and components to cover capabilities in sensor, host, and display processing for range systems. NUWC delivers systems to ranges in a shrink-wrapped form using Web technology. As delivered, the package includes all the assets needed by the range and is customized for integration with the range's hardware and legacy software needs.

The FI project continues to invest in the refinement of TENA and prototyping solutions [Noseworthy 02]. NUWC's intent is to incorporate these and other technologies accepted by the range community into the RangeWare product line as the technologies become available. The remainder of this report deals with the development, sustainment, and application of RangeWare in support of NUWC's T&E and training range customers.

1.2 Early RangeWare Systems

In a set of pilot efforts, NUWC developers used RangeWare to construct and install operational capabilities for three range systems. These three systems are in operation at the Atlantic Undersea Test and Evaluation Center (AUTEC). There are other systems in various stages of planning that will definitely use the RangeWare architecture and components. There are another five to eight programs that are potential product line candidates.

The three pilot systems included the following:

1. AUTEC Hydrophone Replacement Program (AHRP) – a project that expanded the undersea tracking area coverage and added additional test capabilities at AUTEC by installing new undersea sensor systems and associated control software
2. TriService Measurement and Display System (TSMADS) - the Navy element of TSMADS, an experimental range system that relied on passive acoustic sensors and associated specialized processing and operator displays

3. East Coast Shallow Water Training Range (ECSWTR) - a shallow-water training range system scheduled for deployment off the North Carolina coast

The success and lessons learned from these efforts coupled with a growing need for range modernization has led to continued investment in the assets. As more candidate systems for use of RangeWare emerged, NUWC's support of range systems became a true product line effort. The SEI provides the following definition:

> *A software product line is a set of software-intensive systems sharing a common, managed set of features that satisfy the specific needs of a particular market segment or mission and that are developed from a common set of core assets in a prescribed way* [Clements 02].

Table 1 provides NUWC's application of this definition for its support of range systems.

Product Line Definition	**For NUWC**
Set of software intensive systems	Data acquisition, display and control systems for ranges
Sharing a common, managed set of features	Features for tracking, archiving, playback, debriefing, analysis, other applications to support range operations
(The systems must)	
satisfy the specific needs of a selected market segment or mission	T&E / training range community
(and be) developed from a common set of core assets	Rangeware assets including common architecture, pre-built applications, scripts for production builds
In a prescribed way	NUWC processes for • Java development and maintenance for RangeWare • production plan for range systems • configuration management plan

Table 1: NUWC Definition of Product Line for Range Systems

As with most product line efforts, the development and use of RangeWare in fielding systems moved through phases [Cohen 99]. The creation of RangeWare included activities characteristic of the Asset Development Phase: creating the initial asset base and using it for development of a small number of products. After the pilot applications of RangeWare at AUTEC, NUWC moved RangeWare into an Asset Sustainment and Product Development phase. Under this phase, NUWC has continued using the architecture and components for developing new product line systems, improved the assets, and refined processes for asset and product development. In the future, NUWC will expand the coverage of the asset base in terms of new object and information distribution services, as well as applying the assets to new systems. NUWC is also looking for better ways to support its customers as it acquires systems from its development group. With the continued success of RangeWare, NUWC is preparing for the routine operations of asset and product sustainment, while institutionalizing product line practices for all range development.

Since the initial pilots, NUWC has used or is using RangeWare on four more acquisition programs. All of these programs were, from their inception, part of the product line using RangeWare assets and contributing funding for the development of RangeWare. Additional programs have plans to adopt RangeWare over the next few years. The size and scope of these programs vary widely, from a significant instrumentation upgrade to a small research and development effort, yet all of them fall clearly within the product line scope. However, NUWC has not yet used RangeWare as the "end-to-end" software solution for a primary range instrumentation system. The replacement of the primary software system at AUTEC has recently been approved and will likely be the first "complete" RangeWare installation.

As NUWC uses RangeWare to develop systems, the asset base increases in terms of numbers of components. Within the asset base, NUWC has written 22 component applications and 4 more are currently in process. Table 2 lists components currently in the RangeWare asset base; Table 3 lists some of the assets under development. Many of the current applications are data interfaces, supporting communication with range instruments. A few of the applications (XYView, TextView, StripView) are representative of displays that might be seen at any major range facility.

Rangeware Exercise Manager	XYView	TextView
StripView	Exercise SetUp	Communications Data Interface
LATR Data Interface	SPC Data Interface	ARGOS Data Interface
GPS Data Interface	Archiver	Playback
Participant Manager	Message Tool	Command Processor Tool
Static Plot View	Field vs. Field view	Ambiguity Surface View
Range vs. Time View	Bearing vs. Time View	Voice Commander View
Telemetry Commander View	File Parser Tool (Matlab Only)	

Table 2: Current Application Components in RangeWare

2D/3D View	Image Data Interface	T502 Data Interface
SSE Data Interface		

Table 3: Planned Application Components

The scope, variety, and success of initial use of RangeWare has convinced the software staff that RangeWare is meeting its design goals: flexibility, adaptability, and reuse. RangeWare also successfully addresses the performance needs of user systems in terms of handling system throughput and meeting constraints for real-time recording and reporting. RangeWare must however, be regarded more as a "talented teenager" than a "mature adult"—as the existing component application suite of RangeWare, known as AppWare, covers only about one-third of the functions required in a major range instrumentation system.

Many of the current applications in AppWare have only those features that were requirements of the sponsoring program. User interface applications, especially, often lack the “look and feel” of more mature products. Technically, enhancing the “look and feel” is low risk. There is ample motivation within the software staff to make these enhancements, as the user interface can create a lasting first-impression on potential customers who are less likely to be concerned about the elegance of the underlying software solution. (One analogy made by the software staff is that they have successfully developed the software equivalent of a modern automotive design and manufacturing plant, with minimal time and cost from drawing board to factory floor and finished product, but the first few models produced only come with AM radios. The customer can upgrade at any time, but no one has ordered the surround sound. Maybe they *assumed* it was standard.)

Although NUWC can easily enhance the RangeWare assets, enhancement in many cases will have to wait for the right opportunity. Unlike the commercial sector, the Government can seldom include features speculatively. Additionally, DoD regulations place legal restrictions on use of sponsor funds. These restrictions often inhibit adding features the software staff knows will be needed later and/or would be useful or desirable to other programs. Finally, there are some areas of range software that currently have no support from RangeWare, as legacy systems are providing the information via a data interface in current deployments.

A small asset group assures the fidelity of the product line, accepting new or improved assets as a by-product of system development for range programs. As NUWC releases additional RangeWare-based products to ranges, NUWC expects the asset group will be extended to a “virtual development team.” This team will consist of an asset group at NUWC plus developers of specialized components and/or local range maintenance groups that interact with NUWC as an extension of its local team. Common tools and processes will support coordination between the NUWC-based asset group and distributed user groups. Under this concept, user group products can be easily incorporated into the asset base for future users.

1.3 Motivation for the Case Study

Case studies produced by the SEI Product Line Practice Initiative highlight current efforts to achieve a product line approach for fielding related software systems. Previous reports in this series include those listed in Table 4. These case studies cover product lines for commercial organizations, a product line developed by defense contractors for use within defense organizations, as well as product line assets developed by a defense contractor for a government organization.

Title	Mission or Market Area	Authors	Reference[1]
CelsiusTech	Shipboard C2	Brownsword and Clements	Brownsword 96
Control Channel Toolkit	Ground-based spacecraft C2	Clements, Cohen, Donohoe, & Northrop	Clements 00, Clements 02
Cummins, Inc.	Engine control	Clements & Northrop	Clements 02
Market Maker	Stock trading	Gacek, Knauber, Schmid, Clements & Northrop	Clements 02
Salion	Manufacture supplying	Clements & Northrop	

Table 4: Product Line Case Studies Written by the SEI

The RangeWare case study is particularly relevant for two reasons:

1. RangeWare represents a successful product line effort within DoD.
2. RangeWare is currently in a product line sustainment phase. The Navy has reached this phase after successful use of RangeWare in a limited set of pilots and commitment to expand the asset base and its scope of application. Under sustainment, NUWC will continue to refine the asset base but will also emphasize the adoption of organizational practices that will lead to the institutionalization of the software product line as the accepted means for delivering range systems to NUWC customers.

RangeWare does represent a departure from most product line experience within the DoD. NUWC performs essentially all of its development in house with government and/or contract employees. ("In house" does not imply that all asset software is written from scratch: other commercial or government off-the-shelf software (COTS/GOTS) are integrated into RangeWare.) Nevertheless, NUWC is not the typical DoD acquisition organization. For the product line development, NUWC more closely resembles a business unit for product development and sustainment within a commercial organization. The CTEIP Foundation Initiative funded part of the asset development for one year. It is sustained through additional funding as specific programs identify new areas for component development.

The remainder of the report covers the following topics:

- Section 2 – a discussion of the content and structure of RangeWare Assets and the early use of RangeWare on pilot applications. This section deals with technical capabilities of the RangeWare asset base and it architecture. Readers interested in only the product line approach and practices may skip this section.
- Section 3 – a discussion of the practices emphasized by NUWC, including "Structuring the Organization," "Requirements Engineering," "Software System Integration," "Data Collection," "Metrics, and Tracking," "Configuration Management," "Building a Business Case," and others.

[1] See the References section on page 39 for full bibliographic citations.

- Section 4 – summarizes organizational status, future plans for system development using RangeWare, and lessons for other DoD organizations.

2 RangeWare: The Asset Base

Several business drivers influenced the development of RangeWare as an asset base to support future range system products. These drivers included the following:

- reducing the costs of software development and sustainment
- utilizing common instrumentation at multiple facilities
- preparing for the potential demand for multiple-site exercises and/or exercises which cross T&E/training or live/virtual/constructive boundaries

These needs reflect a business context that helped define the RangeWare architecture. Although they are not functional requirements that range systems must meet, they are, in fact, architectural drivers. Another set of architectural drivers is more technical in nature:

- ability to incorporate in real time new or modified range facilities without interruption to ongoing operations
- ability to integrate components into systems and swap components in and out without recompiling
- ease in accommodating changes in format of sensor data or other system data
- integration of legacy systems with the new RangeWare products. Most deployments will involve a migration of older systems to newer ones as upgrades are funded vs. a wholesale replacement of a range system
- coexistence of the architecture with other initiatives that support or interface with range systems
- continuous product improvement as a normal part of architecture sustainment and recognition of the need for local range variations/preferences while maintaining product line configuration management/integrity
- a target of 70%+ as the percentage of software in a range system covered by RangeWare assets

These drivers represent the nonfunctional characteristics of RangeWare. These quality factors cannot be satisfied through algorithms and coding alone; their satisfaction requires the proper structuring and division of functionality through systems built using the RangeWare assets. The response to these drivers and to the RangeWare functional requirements is the RangeWare reference architecture. The rest of this section provides views of the reference architecture and of an instance architecture built using RangeWare assets.

2.1 RangeWare Context

In order to set appropriate boundaries for users of RangeWare, it is necessary to review the context model (Figure 1) as well as the product line architectural drivers outlined above. Within this context, the architecture addresses the drivers as follows:

1. Support the integration of distinct hardware subsystems (sensors, radars, simulated targets, etc.) that may be geographically remote to allow them to interoperate and share data in real time. This is the concept of assembling a logical range for an exercise.
2. Accommodate an evolutionary path to the product line approach for developing future T&E/training ranges. This path leads to a production plan and routine operations. The product plan uses a composition approach to create the range system infrastructure and the sharing of range components.

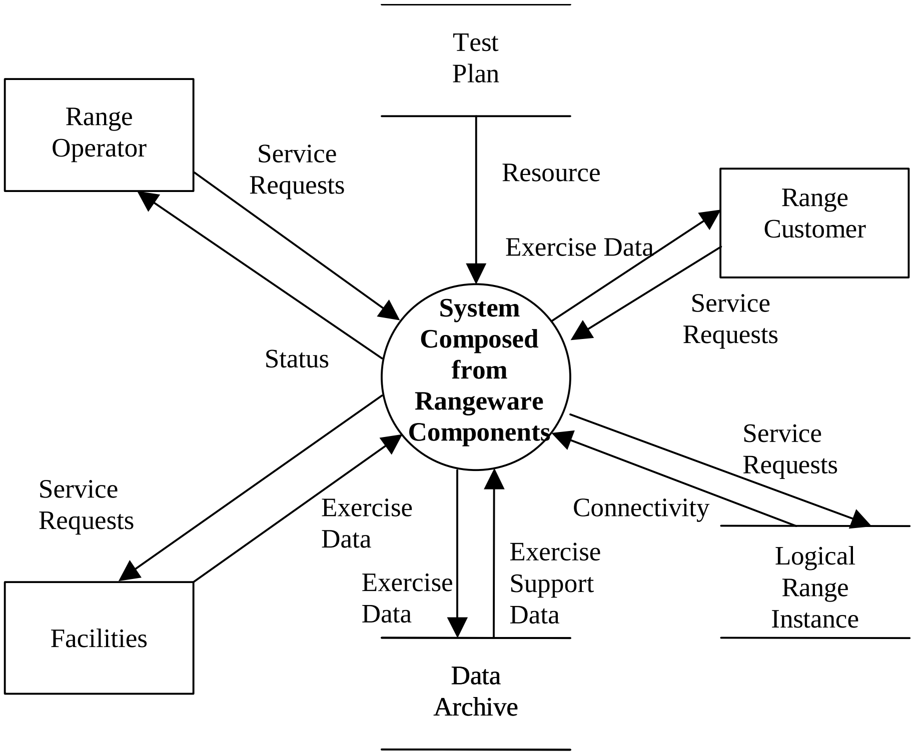

Figure 1: RangeWare Context Diagram

This context represents a range capability—that is, the product line supported by RangeWare supports the ability to assemble assets for the purpose of running specific range exercises. The product line is the collection of systems built using RangeWare to support specific range exercises. The external entities or interfaces that participate in this capability are as follows:

- **range operator** – defines range system configuration, including range assets and connections
- **range customer** – informs range system operator of the types of data required of the range and receives back exercise data

- **facilities** – actual range hardware assets (sensors, radar, targets, etc.) plus network connections with which range system must communicate
- **test plan** – requirements driver for range system for a specific exercise
- **logical range instance** – range hardware assets needed to achieve exercise goals assembled through RangeWare capability (including interfaces to non-RangeWare systems)
- **data archive** – stores information about range configuration and current exercise

The arrows in the context diagram represent data and/or control flow between participants.

A torpedo launch exercise that must integrate undersea sensor data to track the torpedo's path illustrates this context. The range operator defines the system configuration for the range. The configuration may include local range capabilities such as a hydrophone array to track undersea objects. The system configuration may also incorporate simulations from a remote location. For example, before the launch of an actual test torpedo, the crew may run through several simulated launches using a hardware-in-the-loop simulation of the torpedo. The operator sets up the range facilities to integrate sensor data with displays that track the exercise participants within the undersea test range. The integration of these physical range assets includes network connections from range data, to data processors, and then to the display systems. The customer for the exercise is a submarine crew on a training exercise. They must submit a test plan to indicate how they will deploy, run simulated launches, and finally launch the test torpedo. A range system built using RangeWare allows the operator to set up the exercise, process sensor data, display that data, and save it for further analysis.

2.2 RangeWare Content

RangeWare assets include the following:

- applications (such as viewers or archivers) that use and/or implement RangeWare objects
- frameworks that make it easier to build common types of applications
- an API that allows easy creation and manipulation of RangeWare objects
- an API implementation
- interfaces to non-RangeWare applications

These assets define a relatively high-level architecture—that is, a *reference architecture,* capable of supporting the need to integrate independent subsystems, allowing them to interoperate and share data. The reference architecture forms the basis of the architecture for systems built using RangeWare and for implementation of a range system.[2] Object, layer, and

[2] The RangeWare reference architecture satisfies the conditions for a product line architecture [Clements 02]. It specifies the structure of range systems (components and interrelationships), provides guidelines for component use, and establishes the means for handling variability among ranges.

component views provide an understanding of the content offered by the RangeWare reference architecture.

2.2.1 Object View

The RangeWare object model in Figure 2 depicts one view of the software architecture—the relationships between the "players" in an exercise built by RangeWare. Objects in the "Exercise Requirements, Planning, and Execution" category of Figure 2 capture the customer's requirements and plans. Once established, these plans can be invoked by a large number of potential scenarios. The Mission Space, Provider Space, and Support Space objects work together to deliver range capabilities for an exercise. The Mission Space definition includes the Environment (physical range assets), Participants (weapon platform or other systems or combatant participants), and Events. Provider Space options determine how the mission space will be populated. For example, sensor input to the exercise (Environment within the Mission Space) may come from a live sensor tracking a live target or may come from a simulation. Support Space includes the range resources necessary to manage the range and range exercises.

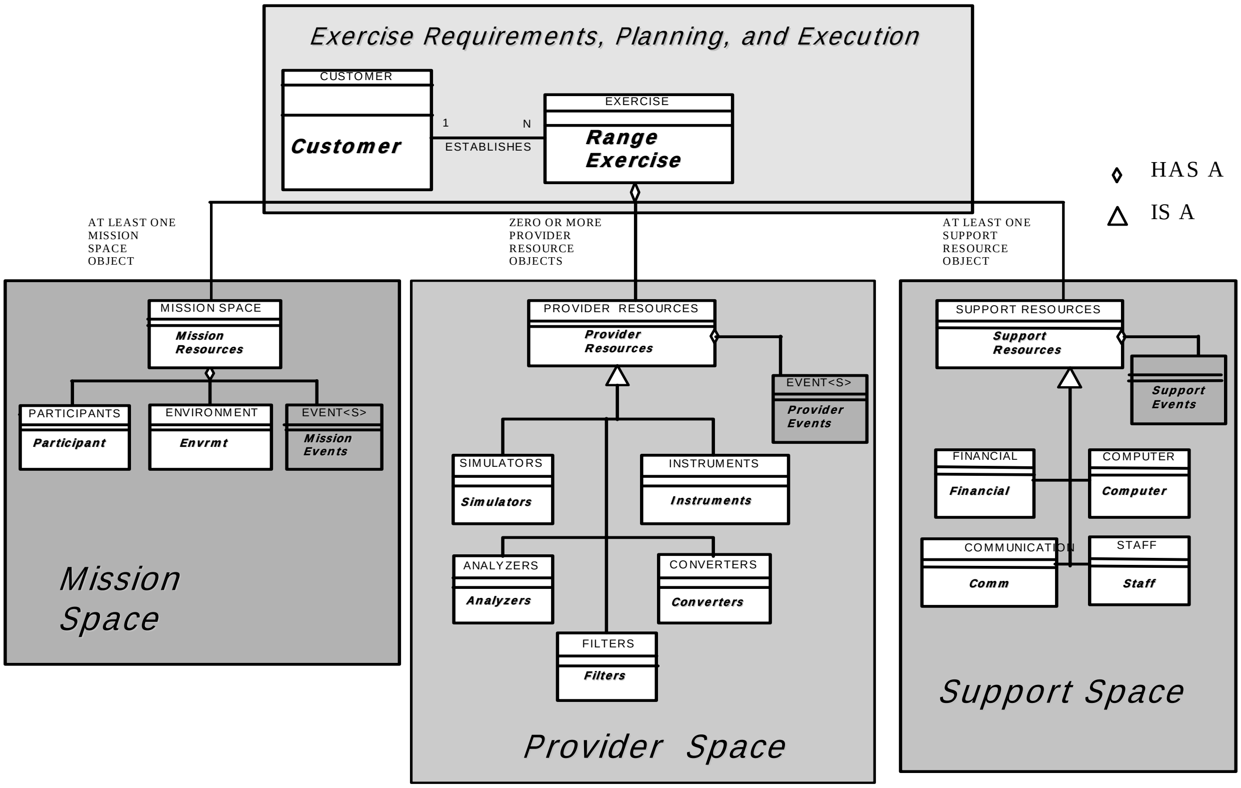

Figure 2: RangeWare Object Model

The scope of coverage of RangeWare assets allows it to support a full set of range operations. The range system developer using RangeWare can rely on assets to develop all aspects of an exercise. Only the Provider Space is covered in most existing range architectures. RangeWare assets include specific components for the Provider Space, but also include an integration capability to build these assets into a range system. Support Space assets give RangeWare the ability to manage an exercise and provide direct support to customers. Expanding the scope of the traditional range problem gives this architecture enormous flexibility and integrative power. If the assets that are available do not meet the developer's requirements, the RangeWare architecture provide two alternative approaches:

1. an API for building new capabilities on existing services
2. a framework for defining an interface and capabilities that must be met by an alternative component

Many of the architectural requirements derive from the need to very closely integrate components into systems.

2.2.2 Reference Architecture Layer View

This high-level structure, or layer view of the RangeWare reference architecture, offers another view of the architecture. Figure 3 illustrates a common API as the core of RangeWare.[3] This API eliminates the need for application developers to rewrite their applications to accommodate various platforms and/or platform modifications required by frequent changes in vendor preference or obsolescence. A combination of language and platform independence was the design criterion for the API design. (See Section 4.2 for more information on this decision.)

DataWare is the name of the implementation of the API. Over time, there may be several implementations of the API, tuned to the unique needs of different systems and/or created by simple mappings to commercially available products. Although the design of the API required language independence, all of the current applications and DataWare itself are written in Java.

AppWare is the term given to the collection of RangeWare applications that rely on the API. These application assets include viewers (XYView, StripView and TextView in Figure 3), archivers, filters, exercise managers and others. Translation applications called Data Interfaces (DIs) support interfaces to systems not native to RangeWare. There are several DIs that use C++ to Java interfaces. These interfaces are a necessary translation tool—in the range community, the opportunity to upgrade all instrumentation systems to a common baseline is rare. Easy integration of legacy systems was a critical design driver, as a multitude of systems, at different stages in their life cycle, often need to be integrated.

[3] Internal working documents describing RangeWare are available from NUWC.

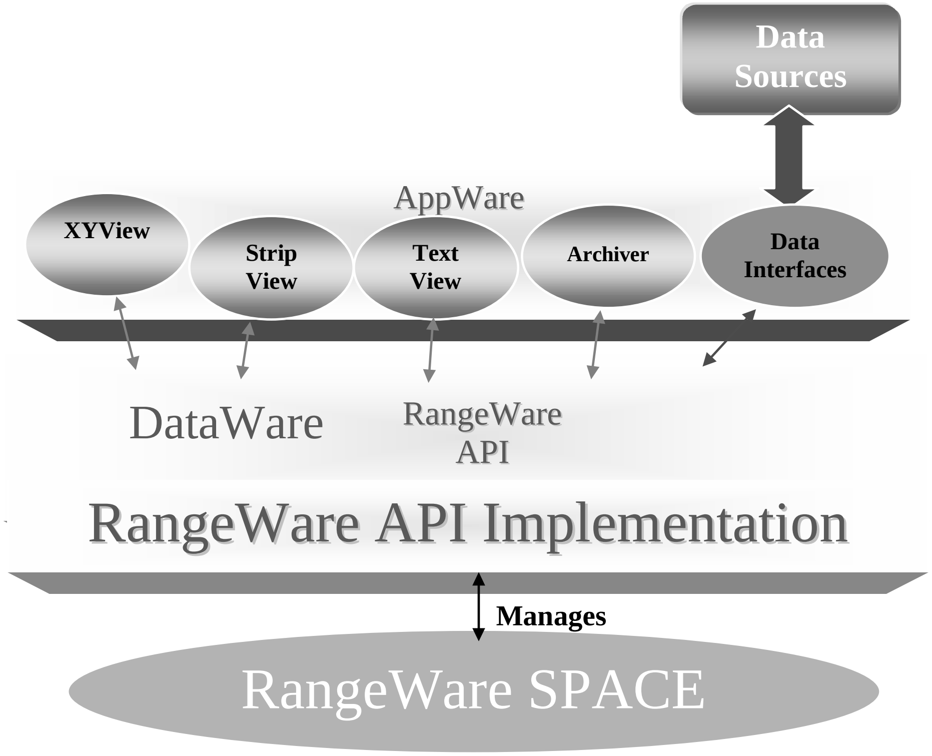

Figure 3: Layers in RangeWare Reference Architecture

The API's main purpose is to allow object-to-object interaction through RangeWare Space—a virtual shared memory resource. RangeWare Space is a dynamic area for range objects such as sensors. RangeWare Space defines the abstract concept of sensor with the mix of typical sensor operations (initialize, start, stop, pause, and resume). The API then supports instantiation of sensor objects as required. Other abstractions are also available in RangeWare Space including viewer, archiver, and other application objects. A range system uses the API interface to notify an object whenever another object tries to activate one of its methods. The API interface traps similar events to an update to an object's attributes. The event may be a create, delete, change, notify, or update – specific conditions are reported to interested objects.

The RangeWare Space is, in effect, the repository for all public interactions among applications. All data does not have to "pass through" RangeWare. The API can be used to invoke methods that set up/negotiate private communication channels for specialized devices. The objects in RangeWare space are logical proxies, indicating only that the capability is part of the exercise, not specifying which application is implementing the capability. A RangeWare object definition can have multiple valid implementations. For example, a simulated device can replace a live device by a runtime selection of which implementation to choose, while the RangeWare object model (and exercise definition) need not change.

RangeWare Space permits integration of different combinations of more well-coordinated software applications. The integrated composition can supply different aspects of the object implementation. For an “aircraft” object, for example, an engine model could supply engine data on one test, an external sensor could support the data on a second test, and the on-board test suite on a third—all without changing the abstract “aircraft” object. This added level of indirection also supports a uniform method invocation to shield callers from the called object. The method need not know how it is invoked, directly or as part of a composite. (The method may be able to determine the caller for security reasons.) For example, a sensor can be modified by a sensitivity command from a range operator or by a monitoring system to inform the sensor that it no longer needs a previous level of sensitivity. Both invocations fire the same method from different origins and use the same API features.

RangeWare supports easy creation of general-purpose command processor and display tools. Introspective services in the API allow a tool at run time to inspect the available methods, parameters, and valid range of values. They also allow retrieval of all public attributes and associated metadata. This information is often sufficient to allow real-time display of data and/or the creation of general-purpose command/control interface without having to reprogram the system for each method and/or attribute change. Each developer provides a startup method for object implementations (i.e. applications). Startup puts application proxies and other objects into the RangeWare Space. A browser tool shows everything needed to the range system operator.

The developer assembles applications and objects through use of the Ant[4] script. This script builds a distribution (or software release) for installation at a range. However, the script does not actually build a running range system. The developer at NUWC must also customize a RangeWare manager to actually begin the startup boot operation that instantiates objects in the RangeWare Space. A RangeWare manager application offers a selection of data interfaces that are available, including displays and viewers. On startup, the manager activates interfaces one at a time as needed and builds the display as requested by a customer. Selection of displays and data interfaces varies from system to system as a result of parameterization within the manager. While all managers perform essentially the same services, the developer must (currently) customize what information is available for each system deployment. Although not currently implemented, the manager also supports dynamic adaptation for joint exercises at different sites that require access to remote data and/or control of remote objects. This feature of the manager allows shared access without the need to modify the local manager.

[4] From the Apache Web site (http://jakarta.apache.org/ant/faq.html#what-is-ant): “Ant is a Java-based build tool. In theory, it is kind of like Make, without Make’s wrinkles and with the full portability of pure Java code. According to Ant’s original author, James Duncan Davidson, the name is an acronym for ‘Another Neat Tool.’ Later explanations go along the lines of ‘ants do an extremely good job at building things,’ or ‘ants are very small and can carry a weight dozens of times their own’—describing what Ant is intended to be.”

2.2.3 Component View

Functional areas within the RangeWare API include the following:

- object creation and management
- attribute operations
- relationship operations
- method operations
- introspection
- notification
- data integrity
- data groups

RangeWare also includes several prebuilt applications for use by ranges. These are built on the API including the following:

- RangeWare Exercise Manager
- RangeWare Exercise setup
- RangeWare Participant Manager
- viewers (XY, text, strip)
- data interfaces (communication, AUTEC real-time graphics online support system [ARGOS], Large Area Tracking Range [LATR], Signal Processing Controller [SPC], Global Positioning System [GPS])
- archiver (currently has limited number of formats)
- playback (matches available archive formats)
- positional viewers (static plot, field vs. field, ambiguity service, image vs. time, bearing vs. time, undersea voice controller, and undersea telemetry controller)
- file parser tool (for use with Matlab)

RangeWare will soon add applications for general purpose 2D/3D viewing plus additional data interfaces. All of these applications come about either as a result of recognition of a general need among range systems for this capability or as a result of a specific request from a range for a new capability. The decision process is covered in Section 3.2.

Table 5 documents some basic capabilities implemented by RangeWare:

Move data
Discover modification dynamically
Display participant data
Display live data describing participant on a range
Display participant data received from a remote range
Display RangeWare objects from multiple data sources simultaneously on multiple views
Dynamically change range environment
Receive and display raw data from a sensor
Monitor sensor/system status
Control sensors and other objects

Table 5: Tested RangeWare Capability

RangeWare object services make extensive use of “listeners.” Almost every object operation (all state changes) can be trapped by the API implementation, and if another application is registered to “listen” for that operation, it will be informed. Since all operations between objects use object services and all operations that change the state of an object can be trapped, virtually anything of interest that happens within this architecture at run time can be monitored. This feature will be used extensively in real range applications. Some obvious uses include

- notification to a display application that the value of an attribute has changed (voids expensive polling—or recomputing raw data that hasn’t changed)
- notification to a range safety officer that an alarm has been triggered
- dynamic reconfiguration of real time exercises—that is, changing relationships between objects, having the effected applications automatically notified of changes (and adapting) without having to interrupt real-time processing (changing data sources, changing I/O channels in response to congestion, bringing additional processing resources online, etc.)
- passive system debuggers and/or security monitors

2.2.4 Process View

From a process view of the architecture, RangeWare supports the host system at a range, as illustrated in Figure 4. Via data interfaces, the host requests information from provider resources such as sensors or simulations. Host processing uses this information as raw input for archiving, filtering, and other data operations. RangeWare includes the prebuilt applications mentioned above to support host processing. The software applications communicate with one another strictly by use of object services. There is a rich set of services defined, including support to add and remove objects, set and get attribute values, lock and unlock resources, invoke methods, set and get relationship information, and several variations of these basic functions. The host utilizes RangeWare display capabilities to manage information distribution to customers.

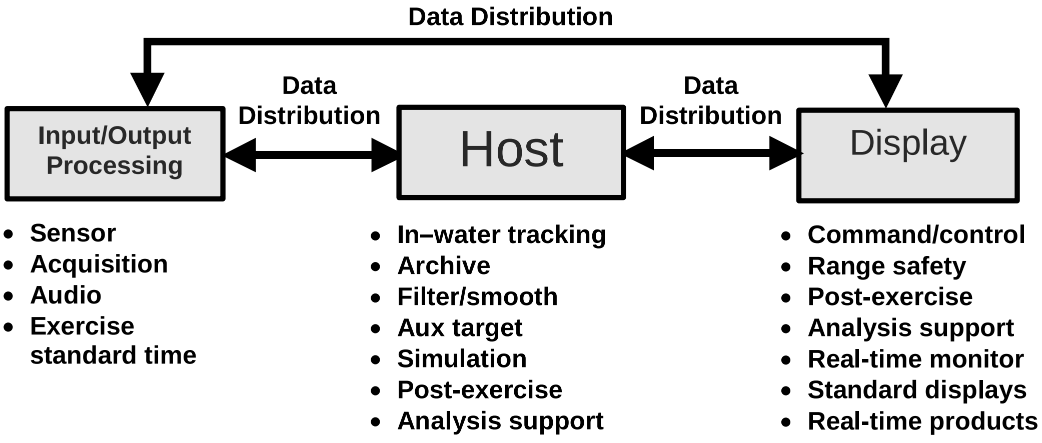

Figure 4:   Processing View of Range System Built Using RangeWare

2.3 Instance Architecture for Systems Built Using RangeWare Assets

RangeWare is not a set of isolated components from which systems are built. It includes and is structured by a reference architecture intended to cover the complete set of range operations. NUWC develops range systems using the RangeWare reference architecture, tailors assets for range-unique capabilities, and delivers the complete system to ranges packaged in a shrink-wrapped form. As delivered, the package includes the assets listed above (see Table 2) needed by the range and customized for integration with the range’s hardware and legacy software needs. The architecture accommodates certain basic design principles:

- discovery – RangeWare components are built to support dynamic discovery. At runtime, discovery determines the capabilities of an object and applies those capabilities without requiring source code changes. The self-describing features often allow integration of the object into the exercise without the need to modify, reintegrate, and re-certify the large software systems used at these facilities. Dynamic discovery is especially applicable to command/control, display, archive and playback applications—which historically comprise about 50% of range software by volume and a high percentage of the maintenance updates.
- dynamic configuration – Dynamic test conditions are integral to most test and training ranges. Dynamic relationships associate site specific and/or test-specific data with a standard object definition. (This is conceptually similar to having multiple simultaneous

"properties" associated with an object.) Examples of site and/or test-specific information which may change independently of the "stable" portion of an object include: command syntax, quality of service requirements, archiving formats, security/access restrictions, display preferences, data format descriptions, cost/usage information, and Configuration Management data. These data would simply be stored in additional objects, each of which was the destination of a relationship originating at the "stable" source object. Exercise setup and other tools access this information to create and modify the test configuration.

- knowledge management and enhanced analysis – Current analysis techniques rely on capturing data in real time and analyzing that data with offline programs. When data is collected at different ranges or from different systems at a single range, formats may differ, or the ranges may use different filter techniques. The analyst must construct special software for merging and analyzing such data. Most often, that task occurs uniquely for each analysis task. RangeWare provides the mechanism for storing meta-information about objects and, thus, about data collected. Analysis programs may use the metadata – for example, an XML specification—to recognize position information, for example, coming from two different systems and make appropriate comparisons and analysis.

A range system built using RangeWare assets (Figure 5) will include data interfaces to capture data from multiple sources. It must also support viewing of that information by specific viewers—XYViews, text viewers, or third party viewers. That data will be archived and available for playback. RangeWare supports the construction of such systems, based on exercise customer requirements for exercises and planning. During execution, sources, viewing parameters, platforms or other elements in the exercise may change. The construction of RangeWare is intended to handle these types of changes. In addition, the knowledge management and advanced analysis characteristics of the architecture support real-time views of data merged from a variety of sources including previous exercises or other archived data.

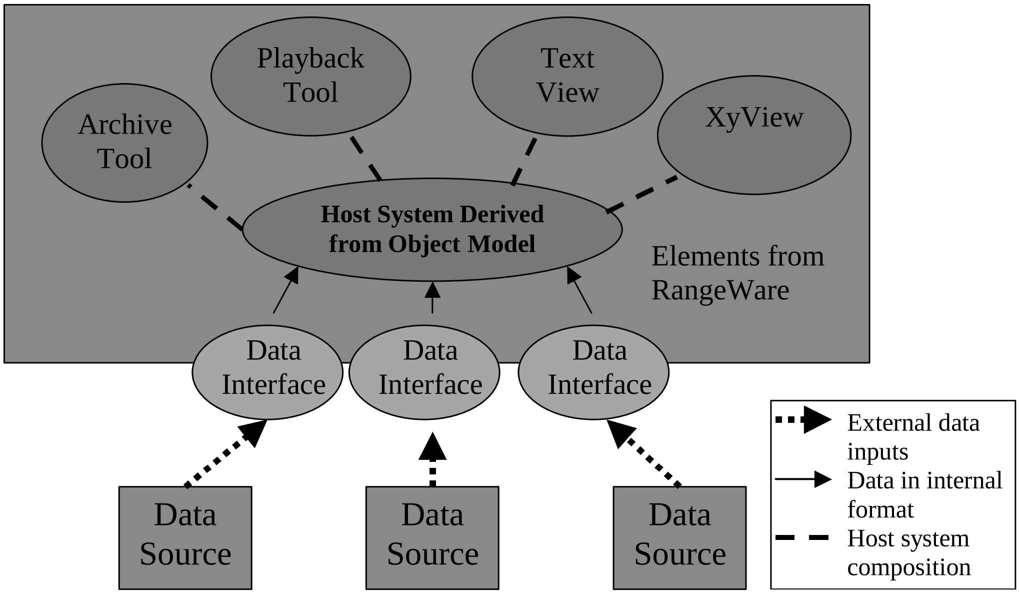

Figure 5: Design for System Built Using RangeWare Assets

The object services API is designed to hide the implementation details of the object operations. This supports the concept of writing source code once and having it used by multiple test situations. Application source code that uses a RangeWare object definition need not change in order to accommodate different implementations of the object (live, high-fidelity simulation, low-fidelity simulation) and/or different transport mechanisms (shared memory on one test, common object request broker architecture [CORBA] on another, and high level architecture [HLA] on a third, for example).

As new systems migrate to use of RangeWare assets, the assets themselves will evolve. The architectural principles that have guided the development of RangeWare and the systems already using the assets will not see significant evolution, however. Having demonstrated the application and effectiveness of the architecture, NUWC feels confident that this architectural approach will continue to provide support to the next generation of systems, even as additional lower-level components are added to the asset base.

3 Product Line Practices

Clements and Northrop define a practice area as follows:

> *… a body of work or a collection of activities that an organization must master to successfully carry out the essential work of a product line. Practice areas help to make the essential activities more achievable by defining activities that are smaller and more tractable than a broad imperative such as "Develop core assets." Practice areas provide starting points from which organizations can make (and measure) progress in adopting a product line approach for software* [Clements 02].

In developing RangeWare, NUWC displayed mastery of many of the software engineering practice areas defined by the Framework for Software Product Line Practice [Clements 02] for asset creation including "Understanding Relevant Domains," "Requirements Engineering," "Mining Existing Assets," "Architecture Definition," "Component Development," "COTS Utilization," and "Testing." NUWC also made inroads in several of the technical and management practice areas, such as Configuration Management, Tool Support, Launching, and Funding.

As part of the transition to a Sustainment Phase[5] for RangeWare, NUWC is improving its application of software engineering as well as technical and organizational management practices. The improvement activity includes reviewing the existing product line and architecture to establish a reasonable maintenance/upgrade/enhancement plan for RangeWare based on the results obtained from system development. Architecture maintenance and enhancement may include architecture assessments to determine the needs for enhancement as new programs adopt RangeWare. Component development may include actual enhancements to product line components and ensuring that new versions of COTS products are integrated into the product lines. Product line support should include working with vendors to coordinate maintenance of their products. Product line operations also provide for updating products for the various customers/users according to maintenance/upgrade agreements established at the initiation of a system development or acquisition. The maintenance and support of the product line architecture and components are a natural consequence of the product line development strategy.

[5] The Sustainment Phase includes ongoing product line operations—the institutionalization part of the "Launching and Institutionalizing" practice area.

This section describes and evaluates the progress in several of the practice areas on which NUWC is concentrating at this time.

3.1 Structuring the Organization

NUWC assigns a project manager (a “team leader”) to each range system development. The actual development staff for that system is matrixed from a staff of software engineers. System development has a designated budget to support the software staff and/or negotiates a budget with the software team. It is up to the project manager to track the tasks of individual software engineers. Only recently has it become commonplace for team leaders to share cost information with the software management—unless, of course, software costs were exceeding budget estimates.

NUWC has charged a single RangeWare group of 6-10 developers with the tasks of asset development, asset maintenance, and product development for range systems. At any one time, the group may have six systems under development for acquisition programs at the ranges. These systems may be in various stages of planning or development. Not all systems are part of the product line supported by RangeWare. Currently, NUWC has some range systems that are not using RangeWare but is working to move all major development within a product line approach. NUWC does not have a separate asset support group, because all asset sustainment must occur in connection with the effort for a specific acquisition program. Furthermore, there are currently no funds to support RangeWare improvements not tied to a specific program.

This organizational structure works well for delivering systems. The RangeWare group determines what is available from RangeWare as it currently exists, what’s needed to be added for new system capability, and, if added, how the capability might be of advantage to others. NUWC has created a production plan that includes enactment of specific practice areas to deliver systems. The enactment of these practice areas is characteristic of the Build Product pattern documented by Clements and Northrop [Clements 02]. The next sections discuss NUWC’s application of most of the following practice areas from the Build Product pattern:

- “Requirements Engineering” for range systems
- decisions about “Component Development” where current assets do not meet those requirements
- “Architecture Definition” for creation of the instance architecture (see Section 2.3)
- “Software System Integration” of RangeWare assets into systems delivered to ranges
- “Testing” of systems and of changes to assets used in developing those systems
- “Configuration Management” of asset use within the product line

The current organizational structure at NUWC does not, however, encourage the software organization or the RangeWare group to consistently obtain the metrics needed to track process improvements. As NUWC makes improvements in the “Data Collection, Metrics, and Tracking” practice area (see Section 3.6), these data will support planning for improvements in the organization.

3.2 Requirements Engineering for Range Systems

When a request for a range system arrives at NUWC, the RangeWare group must determine if the product development capabilities of RangeWare are capable of supporting the new or upgraded system’s requirements. This determination is a manual process of assessing requirements against RangeWare capabilities. Where the requirements are a match, the RangeWare group can immediately begin product development of a range system. If current assets are not sufficient, the group will take one of several paths:

- Negotiate with the customer over changes that existing assets can accommodate and upgrade existing assets to support the unmet requirements.
- Perform component development to add new reusable assets to support the unmet requirements if it seems likely that other users in the future will need the new assets or the same users will need future adaptations of the same asset. (The sponsor’s cost to develop a reusable asset cannot normally exceed the cost of developing a similar component that is not designed for systematic reuse. In practice, this yields some applications designed for reuse, but populated only with those features the current customer requires.)
- Identify other customers that have similar and/or compatible requirements and may wish to capture the opportunity to share development cost of an asset designed for systematic reuse.
- Recognize the requirements are so unique that designing a reusable asset is not appropriate. A range or system-unique RangeWare application can be built. (A system-unique RangeWare asset still offers opportunistic reuse potential.)
- Determine that the range requirements are beyond the scope of RangeWare. In approximately eight systems where RangeWare was chosen for at least part of the customer solution, and in most instances the majority of the solution, each system has at least some components or subsystems for which RangeWare was not determined to be the best solution at the time.

3.3 Software System Integration

During software system integration, the product development group combines individual software components into an integrated whole. To carry out system integration, NUWC has established a production plan that combines a simple scripting approach with management of customer requirements.

All RangeWare assets are stored in a repository. This repository is closely controlled both in terms of configuration management, to assure only authorized users may change or add assets, and in terms of applying assets to a given system. The RangeWare group uses standard Ant scripts to build a range system for turnkey installation and use at the range. From the repository, the scripts perform the equivalent of a "make" operation to assemble the proper components into a build. In creating a build for a given range, the RangeWare group relies on prior experience and knowledge of RangeWare. The execution of the Ant script produces software for an installation disk of the range system.

NUWC offers several modes for distribution of system builds derived from RangeWare:

1. NUWC distributes updates by mailing a CD.
2. The development staff installs a CD at the remote location.
3. Ranges perform a remote download from a protected Web page maintained by NUWC.

Ranges select the appropriate mode based on their system and personnel specifics. A key factor is the nature of the local range support system, including hardware, software, and people. The actual target environment and its complexity are also factors.

The RangeWare configuration management (CM) system offers a direct access capability for range software maintenance staff. This direct access to NUWC's CM system allows ranges to check out, build and test systems in their operational environment after lab testing at NUWC. Ranges may make a "test build" before "promoting" to a deployed build, as certain elements of the system might be completely testable only with specialized equipment on site. (Although the range may perform tests with simulation of this equipment, such tests are not identical with a full, operational test.) Ranges have not yet used this "remote CM" capability for RangeWare, but they have used the capability for other NUWC software distributions. NUWC expects ranges will use this distribution and testing approach as more applications are deployed for future RangeWare users.

3.4 Testing

As the RangeWare group makes changes to assets, testing occurs at two levels:

1. A programmer checks in an update to an existing module that other systems use. The programmer is *not* responsible for assuring that updates didn't "break" every other (or any other) system. Programmers are responsible for local testing of the module, but that might not cover everything about the module that everyone else cares about.
2. CM and quality assurance (QA) staff note the check-in of the updated module. CM informs software staff of the update, and project managers receive updates of CM status on a monthly basis.

No regression or integration testing of the updated component actually occurs until a project tries to rebuild or re-deploy a system using that module. Testing of the upgraded asset may occur in two ways:

1. Programmers are working on modifications to a system using a component that was updated within the asset base. With knowledge of the area of change affected by the updates, they perform testing to see if everything "still looks OK" within the system using the new version of the component. Programmers together with QA complete a full test procedure when the completed system is ready for deployment. In effect, this test procedure performs regression testing on the component, at least within the context of a single system.
2. Other times, no one is actively working on systems potentially affected by an update. No one does complete regression testing—running through factory and/or operational test procedures—on projects that are "inactive"(that is, those not having money to pay for testing at the time). The CM system has tagged the current deliverable, but when or if a new deliverable, requiring a rebuild, is needed, it will be built with all of the upgraded software, unless the program directs the use of previous versions. It is that project's responsibility to test/accept/update a new release, or revert to the current release under CM if necessary.

Fortunately, projects generally benefit far more often from upgrades/fixes to modules done by other projects than they are inconvenienced by the need to upgrade/test to maintain currency with upgrades. Although the NUWC software staff would prefer to perform more rigorous testing more often, current processes recognize fiscal constraints while still maintaining the integrity of deployed systems.

3.5 Configuration Management

The RangeWare CM program establishes and maintains the integrity of RangeWare and range systems products throughout the software life cycle. The CM program identifies developmental items for the software project that must be controlled. These items may include a framework, code components, a requirements description, Ant build scripts, or a delivered range system. The CM program controls these configuration items and controls changes to them. CM tools record and report status and change activity for these configuration items. Because individual assets may be used on more than one system, CM must report which assets a specific range system uses in order to support upgrades.

A complete Software Configuration Management Plan (SCPM)[6] defines CM for RangeWare. The plan defines the CM organization consisting of a software configuration management group, project management, and configuration review and control boards. The plan extends beyond the CM function and defines the tools used for the development life cycle: design, development, implementation, and testing. These are primarily off-the-shelf, general purpose development tools, tailored to the specifics of the RangeWare configuration and requirements.

[6] The SCMP is an internal NUWC document available from the RangeWare group.

The plan also covers coding and documentation standards. It prescribes the procedure for creating, updating, and viewing documentation for a specific project. The CM plan defines procedures for creating a RangeWare software mainline and software baselines. The mainline is the collection of assets from which all developers create a build or a branch. The mainline is upgraded whenever there is a change to an existing asset or a new asset is added to RangeWare. CM creates a baseline for each system consisting of a suite of configuration items that the RangeWare group will eventually use to create a distribution for a range system. The plan defines procedures for submitting change proposals for RangeWare and the specific activities for checking items into and out of the repositories for either the mainline or baseline software. Finally, the plan describes the process of building a configuration for release to a range.

NUWC strongly supports its CM program. To date, it has established

- a Software Configuration Control Board
- authorization and tracking of software change proposals for RangeWare
- authorization and tracking of source code changes
- requirements for updating the SCMP

As the SCMP is updated, ranges will have a consistent approach for accepting and integrating newly distributed systems.

3.6 Operations and Tool Support

The “Operations” practice area defines how the product line organization develops and maintains the asset base and how it uses the asset base to create products [Clements 02]. For range system products, the operation begins a new cycle when a customer representing a range system comes to NUWC with a request to build or upgrade a system. NUWC determines if RangeWare is appropriate. Other systems are also available to support the needs of the range, for example a PC-based display and debrief tool called Tsunami and an earlier range system called NTADS. Where long-term maintenance is a primary concern, RangeWare is usually the recommended approach. The expected longevity of the range system is a factor in determining this approach. For a prototyping effort, a range capability built using MATLAB may be sufficient. Where Tsunami or NTADS is installed and only minor changes are required, upgrades to those systems are preferred, especially if the requirement is only for a short-term project or exercise and/or is easily met by the other products.

NUWC is applying a Software Process Improvement Initiative (SPII) to improve operations. The initiative will develop process definitions and long-term plans. NUWC only supports this effort on a part-time, voluntary basis. As RangeWare moves through the sustainment stage, SPII must be built into operations as a part of continuous improvement process.

NUWC uses off-the-shelf tools to support software development. For building and maintaining elements of RangeWare, most developers use Visual Cafe for Java. Win-CVS is the configuration management tool to support versions of RangeWare and range systems built using RangeWare assets. All developers have available the same set of tools. Allowance is made for developers to use other integrated development environments as long as they utilize the common CM tools. To configure an exercise, a range operator must link together resources available at the range. The RangeWare Exercise Manager application (See Section 2.2) supports the configuration of sensors and other range resources into the exercise system.

Figure 6 illustrates this process in operation. Without RangeWare, much of this configuration must be hard-coded into the range system.

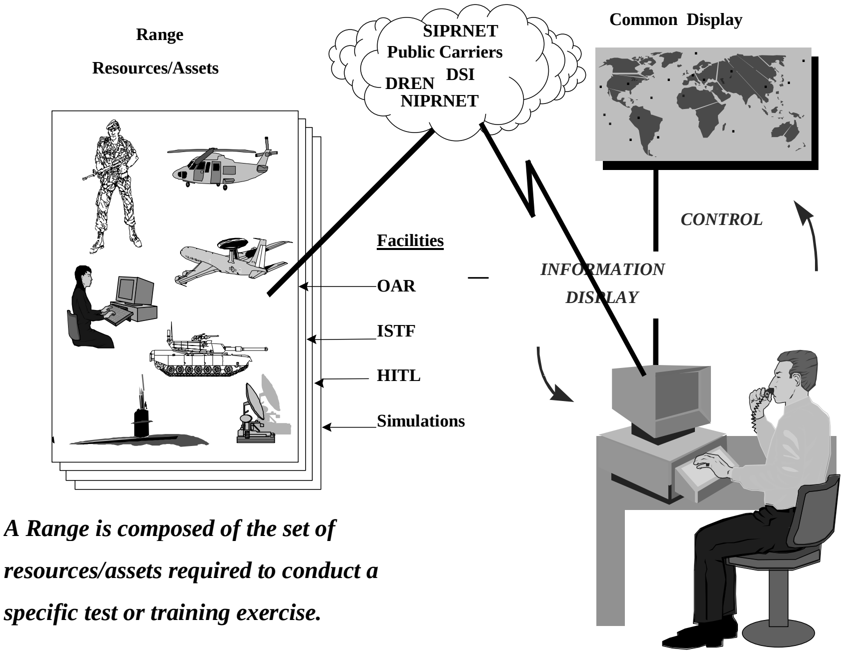

Figure 6: Range Operator Using RangeWare Exercise Manager to Build an Exercise

3.7 Data Collection, Metrics and Tracking

To track the effectiveness of a product line effort, management must first set performance goals for an organization. The goals may include reduced costs, improved quality, reduced time to field, or less tangible results such as customer or developer satisfaction. In order to determine whether the organization has achieved its goals, the organization must collect data related to that effort. The organization must determine which measurable attributes of the effort's process and product are relevant, and collect data that reflect those attributes. Looking at the data, a manager should be able to determine if the organization is meeting its goals or if the effort has diverged from expectations. This vital organizational data can then support management decisions to maintain the course, if the effort is successful, or to replan if goals are not met [Clements 02].

NUWC has an intuitive sense (and some large-grained metrics to support this belief) that RangeWare is achieving the primary goals: reducing the costs of software development and sustainment, improving time to field, and achieving common instrumentation across multiple facilities. Although there are some metrics available, NUWC has not yet established an adequate software metrics program. The only data that NUWC formally tracks are lines of code in RangeWare itself and lines of code from RangeWare that are used in the deliveries of individual systems to the ranges. The large-grained measure of effort on specific programs is often limited to total software cost. It is often not possible, with surety, to determine the cost or time assigned to a specific program for a specific software task. NUWC also does not have an archive of historical data in order to make comparisons of current and legacy developments.

NUWC's current strategy is to improve data collection in order to measure the effectiveness of use of RangeWare. On a system-by-system basis where RangeWare is a development factor, certain managers will now be tracking specific task hours to meet both ISO process standards and assist in metrics collection. Implementing a complete measurement program is a strategy that will permit NUWC to justify support budgets and also demonstrate measurable results to potential customers at DoD ranges.

3.8 Building a Business Case for Product Line Sustainment

NUWC plans to use metrics data and product development plans to predict benefits of widespread product line adoption for the organization. The business case that NUWC has developed uses three elements that may be termed A-B-C as a mnemonic:

- **A**pplications – number of range systems NUWC plans to deliver using RangeWare over the time period of the business case. This may be three systems per year, based on prior years' experience.
- **B**enefits – the projected cost savings or other return on the use of RangeWare should provide. NUWC hopes to reduce costs for delivering a system to range by 70% over current approaches. Again, without consistent historical data and complete tracking of current data, these results are only estimates.
- **C**osts – the actual costs NUWC incurs in using assets. Because RangeWare uses scripts to perform a large portion of the system integration effort, there is little cost for using a pre-existing RangeWare asset in a delivered product. There will be costs for maintaining the assets, such as correcting errors or adding assets, but those costs are calculated separately and can be amortized over the number of systems supported or delivered with the asset.

A typical business case based on return on investment in RangeWare development includes projected costs for a set of systems built using traditional development approaches and comparing that to the estimated cost of using RangeWare to produce the same systems. These projections may then be compared to currently available empirical data and refined.

NUWC builds many "classes" of range systems, often simultaneously. Table 6 shows the approximate costs to develop and maintain these different classes of systems. During any given year, they may be working on a major range system (i.e., complete end-to-end replacement from sensor software to operator stations, a small range system with limited sensors or operator positions, various types of range subsystems, or engineering prototypes). It is rare that all of this effort would be happening in a given calendar year. Usually every development year includes work on a mix of systems for multiple projects. Table 6 estimates costs based on a $10 per line of code[7] and 10% maintenance cost that NUWC bases on historical data.

Program Name	Lines of Code (in thousands)	Development Cost (in millions)	Maintenance Cost per year (in millions)	Number of Years in Maintenance	Cost with Maintenance (in thousands)
MAJOR RANGE A	700	$7,000	$700	20	$21000
Small Range B	400	$4,000	$400	20	$12000
Subsystem A	250	$2,500	$250	10	$5000
Subsystem B	150	$1,500	$150	10	$3000
Subsystem C	50	$500	$50	10	$1000
Eng Prot A	150	$1,500	$150	4	$2100
Eng Prot B	50	$500	$50	1	$550
Total		$17,500			$44,650

Table 6: Estimated Costs of Building Range Systems without RangeWare

NUWC's business case development next considers the estimated costs of developing the range systems using a product line approach (PLA). For each program in Table 6, Table 7 provides the costs of developing the systems where RangeWare provides 70% of the total code for each program.

[7] The low cost per line of code results from a high level of ad hoc reuse already in place within NUWC.

Program Name	Lines of Code (in thousands)	Development Cost (in thousands)	Maintenance Cost per year (in thousands)	Number of Years in Maintenance	Cost with Maintenance (in thousands)
MAJOR RANGE A	210	$4375	437.50	20	$13125
Small Range B	120	$2500	250.00	20	$7500
Subsystem A	75	$1562	156.25	10	$3125
Subsystem B	45	$937	93.75	10	$1875
Subsystem C	15	$312	31.25	10	$625
Eng Prot A	45	$937	93.75	4	$1312
Eng Prot B	15	$312	31.25	1	$343
Asset Base	150	$2250	225	20	$6750
Total		$13,185			$34,556

Table 7: Estimated Costs of Building Range Systems with RangeWare

The estimate provides a 25% cost of reuse to obtain that 70% figure, meaning that there is a substantial cost for using RangeWare. In reality, this value should be much lower, but the business case makes a very conservative estimate of savings. The business case also assumes a higher development cost per line of code. NUWC will invest resources in developing any system-unique software for possible future reuse, 150% of the non-PLA case, to account for additional code complexity of the reusable software. Table 7 also shows the required investment in “baseline” assets—in this case the RangeWare API and other supporting software, required for other projects to use the product line.

A comparison of Table 6 and Table 7 shows that using these conservative assumptions, the PLA will save over $4 million dollars in development cost and $10M in maintenance costs versus the non-PLA approach. Realizing these savings is a function of the mix of projects (both how many and what type) are actually funded and in progress over a given time frame. A mix of projects is required to recoup the $7M investment in the asset base.

Table 8 shows actual information from recent software efforts that used RangeWare. During this time frame, there have been no major or small range projects, but there have been multiple range subsystems—one in class A, four in class B, and one engineering prototype. The actual code count and development costs are available, and show an average cost per line of code of $12.56. As expected, this is higher than the $10 estimate for the historical development approaches. It is not, however, nearly as high as the 150% penalty estimate, and already includes the “cost of reuse” factor.

Program Name	Lines of Code Estimated (in thousands)	Actual Lines Developed	Estimated Cost (in thousands)	Actual Cost (in thousands)	Years of development
Subsystem A	250	(Est.) 75	$1562		
ECSWTR	**245**	**80**		**$1300**	
Subsystem B	150	(Est.) 45	$937	$10	
AHRP	**165**	**25**		**$200**	**99-01**
LSVTC	**150**	**20**		**$210**	**01-02**
OASYS	**150**	**28**		**$219**	**01-02**
SCORE	**150**	**20**		**$115**	**01-02**
Eng Prot A	150	(Est.) 45	$937	$4	
TSMADS	**152**	**22**		**$540**	**99-02**
Asset Base	150		$2250		
DataWare	**130**			**$1500**	**98-00**

Table 8: Actual Results from Use of RangeWare

These results show that the ABC method projections understate the actual savings from delivered product line systems. Some mix of lowering reuse costs could result in a more accurate predictive model. As NUWC's data collection process matures, the business case will improve from a rough estimation of cost savings to a predictive model for product line investment.

3.9 Funding

RangeWare depends on new projects for continued funding under the purview of DoD regulations. NUWC must address the challenge of identifying other sources of funding to permit sustainment of RangeWare that are not dependent on a specific project. The RangeWare group operates as a customer-reimbursable (not-for-profit) government lab. Within the DoD regulations of this business environment, NUWC can only develop the software specified by the customer for an individual range. NUWC may build software for one acquisition program and share that software. However, NUWC will not charge a program for an anticipated future need of another program, although the overall savings to the DoD may be greater than the original charge. NUWC may add provisions for variations, as long as costs are not increased, but they may not add the variations. NUWC has developed and evolved RangeWare within this regulated environment by identifying how it can field capabilities, using the RangeWare architecture that can satisfy customer requirements but also extend RangeWare. Since AHRP, five more programs have provided funding. This evolutionary approach will work as long as there is a small number of customers. As the customer base expands—and expansion is anticipated—NUWC must adopt a long term strategy to anticipate a broader set of evolutionary improvements.

Up-front investment for RangeWare came from the CTEIP Foundation Initiative and AHRP program funds. CTEIP and AHRP recognized the value of building software to address common needs across multiple ranges and building applications to accommodate future range needs, respectively. The AHRP program immediately benefited from RangeWare, and successive range developments have extended the coverage of RangeWare. To satisfy DoD regulations, NUWC must identify similar funding sources such as the Office of Naval Research, Defense Advanced Research Projects Agency, or other DoD organizations. NUWC may go back to the CTEIP office for funding. NUWC would use this funding so that recognized improvements in RangeWare or desired extensions need not wait for a sponsoring program that needs those specific improvements or extensions. It may also be possible to build RangeWare enhancement into funding requisitions for specific sensor, weapons, or range programs to allocate dollars for new capabilities. Under this approach, part of the program infrastructure budget will put RangeWare into applications.

Similarly, NUWC may seek funding from overhead accounts to work on software process improvement. NUWC recognized it did not have sufficient funding for personnel to satisfy mission requirements without taking radical approaches. Because development and use of RangeWare departs from standard approaches, the RangeWare group must develop its own process descriptions and development guides. However, these go beyond the normal level of overhead activities. Based on savings to NUWC customers, the RangeWare group will identify new ways to fund its own process and product improvements.

3.10 Customer Interface Management

The range systems built using RangeWare are delivered to RangeWare users—the ranges. These ranges are NUWC’s actual customers. The ranges, in turn, have their own set of customers—the DoD operational organizations dependent on ranges to support their missions in training, testing and system evaluation. When a T&E or training need arises, the operational organization needing to run an exercise goes to the range to use its services. That operational organization becomes a direct customer of the range. Indirectly, the operational organization is also a customer of NUWC and may raise issues with the range systems NUWC supplies to the range. NUWC must justify the solutions offered through use of RangeWare over the previous range-unique solutions.

NUWC knows that many of its range customers prefer the ownership rights they possess with RangeWare as government-developed software. With real-time software, customers usually need their own people in their own organization capable of maintaining range software. While much of the infrastructure is COTS, use of government display and other applications software avoids proprietary systems, outside maintenance, and license fees. The range software applications are sufficiently specialized that there is not a wide range of off-the-shelf solutions available. When available, commercial software may offer additional features, but at the cost of loss of control and, potentially, long-term dependence on a contractor or a specific platform. More than a few ranges have experienced a support crisis when commercial suppliers went out of business or simply discontinued support for unprofitable software products still used at the ranges.

With NUWC as a source, the ranges recognize shared responsibilities. NUWC and the range customer negotiate alternative solutions within the scope of RangeWare or other product lines. NUWC has experienced problems in managing customer expectations. Some customers, or potential customers, have asked why RangeWare doesn't produce displays as elegant as those of some more mature deployed systems—systems from commercial developers, or even their home computers' video games. Because the product line is "new," customers are often expecting a "wow."

Much of the "wow" in RangeWare is "under the covers" in its ability to deliver systems to field more rapidly, with higher quality, at lower costs. Unfortunately, often what the customer sees at the user interface is the "same old thing" they've always seen. In some cases, they actually do see less capability—that is, only that which meets the requirements, not the requirements "creep." They may have seen a pre-deployment prototype that does not have all the user interface features required in an operational deployment, but does have a glitzier look and feel. Some of the things they would like to see are absolutely trivial to add, they simply require a sponsor willing to pay for the enhancements. RangeWare is still, for many, in the "prove it to me" stage. For a smaller, but growing population, it is a superior product that is just beginning to realize its promise. NUWC must persist in selling the advantages of product line development and assure potential customers that their participation will accelerate RangeWare development and permit it to catch up in the packaging without compromising quality.

RangeWare does not yet have a user's group. However, a user's group could address several key issues of interest to ranges. So a RangeWare user's group could satisfy two interests of the ranges:

1. working with internal project teams at NUWC to address RangeWare in general and the specific use of RangeWare for individual ranges. Users could recommend the kinds of improvements or enhancements they would like, and as new systems come into the product line, NUWC could target those areas within a new system. For example, there is little interest in reengineering the lower level infrastructure components of RangeWare because users do not see these components. Users may agree on a set of new viewers or improvements to existing viewers that could be candidates should a new development require them, as well.
2. a users group such as the Range Commanders Council could help manage the interface between ranges using systems from the product line and the range customers from throughout the DoD. Users would recognize common range issues surrounding their use of RangeWare applications to support exercises. This approach may be appropriate after completing RangeWare Improvement (RWI), the next system in the product line.

Government and government organizations must do what's in their best interest. NUWC's success with RangeWare will continue only as long as it can compete by satisfying customer needs in cost, capability, and schedule.

3.11 Launching and Institutionalizing

Successful software product line practice is not simply a matter of the correct architecture and software. Customer relationships, organizational structure, and management practices are also significant ingredients in product line adoption. The successful creation, use, and evolution of a product line require insight into customer needs and careful planning. While there are substantial benefits from product line adoption, they come only over a course of successive product developments using the product line assets. RangeWare has demonstrated the staying power with a succession of customers due to the insight and planning within NUWC. The success of RangeWare demonstrates that product line development in the DoD can succeed where technical expertise, management foresight, and long-range goals are present.

The CelsiusTech case study [Brownsword 96] defined four stages in product line adoption:

- pre-product line
- product line creation
- product line routine use
- product line evolution

Table 9 summarizes NUWC's experience in moving through these stages of product line adoption.

Stages in Product Line Adoption	**NUWC Experience**
Pre-product line	1997: NUWC published the specification for TENA
Product line creation	1999: NUWC used TENA concepts in the implementation of assets to support range operations creating RangeWare
Product line routine use	2000-02: NUWC validated use of RangeWare on three systems and subsequently used RangeWare as the basis for development of four additional range systems
Product line evolution	2002: NUWC has adopted a Software Process Improvement Initiative. This initiative will examine practices for use of RangeWare to support its long-term viability.

Table 9: NUWC Experience in Stages of Product Line Adoption

NUWC continues to add RangeWare users and range systems to the product line. The evolution that will take place during the Sustainment Phase will expand the product line in two ways: 1) RangeWare will cover more capabilities required by ranges, and 2) the capabilities offered will give users more alternatives to select from in meeting their needs.

4 Summary

NUWC's experience with RangeWare demonstrates that DoD organizations can succeed in product line development. This experience includes changes in technical, organizational, and economic approaches needed for similar organizations to embark on a similar path. This story is similar to those of other organizations described in the product line case study series. This section summarizes the main points of the NUWC experience.

Although the RangeWare experience at NUWC is ongoing and lacks process maturity in some areas, the RangeWare group has made significant progress. From a comparatively small investment of $3.5 million, they have realized cost savings of approximately $15 million. Approximately $1 million of the initial investment occurred before the first product was delivered. The development of the initial asset base and first product occurred within one year, so there was no significant lead time or large up-front investment.

Use of RangeWare is considered by most a recent innovation. Although it has some strong supporters, especially among team leaders with strong software backgrounds, for many managers, the product line approach is not yet seen as the standard way to build a new system. Team leaders are asking for answers as to when RangeWare will start saving them money. They point out that saving their customer life-cycle costs is nice, but not at the expense of overrunning development cost estimates. They expect RangeWare products to have (or soon have) the look and feel of more mature range systems. There is often attribution to "RangeWare" for any and all problems with software, such as requirements creep or misunderstandings. Senior people from NUWC often must convince the project team manager of RangeWare's effectiveness and/or value. Of course, part of the reason is that the business case has not been fully articulated, as metrics are sparse. Recently, however, NUWC has noticed greater buy-in to the operations approach served by RangeWare. Two relatively small projects have recently committed to RangeWare. They saw most of their requirements met by already existing RangeWare, and they would be hard-pressed to duplicate these with their budget.

As RangeWare became the basis for product development, NUWC has experienced a positive transformation. Prior to RangeWare, NUWC was a system house, with individual project groups developing and maintaining individual systems for its range customers. Project group developments duplicated one another, and even minor coordination among groups required significant effort. Today, NUWC sees itself as the supplier of off-the-shelf software, tailored for installation at a range. The development groups prefer their current role, the increased productivity they achieve, and the collaboration among projects. NUWC can now engage in the technical and organizational planning to adopt practices that will allow it to support a growing number of customers without expanding the development team. The planning includes new ways to secure funding, manage the customer interface, and define standard processes.

4.1 Product Line Payoff

NUWC and its customers have recognized both tangible and intangible benefits from use of RangeWare. Cost of building software for ranges is at least 50% lower using RangeWare. Development time has also been cut from years to months for several applications. Total personnel for projects may be cut by up to 75%, allowing NUWC to take on additional assignments. As new programs add to the list of RangeWare users, the asset base grows and increases in capabilities to satisfy the requirements of the new programs. NUWC can then deliver range systems to an even greater potential audience with a compelling set of competitive benefits.

NUWC also derives less tangible benefits from greater customer and developer satisfaction. New programs recognize the value of RangeWare in satisfying their requirements reliably and predictably. As the number of users increases, RangeWare can virtually sell itself to new programs. At the same time new programs represent new challenges to the staff – they are no longer engaged in rebuilding the same capabilities as those on previous programs. This has the salutary effect of having engineers constantly working on new and challenging elements in a program and on new ways to apply and enhance RangeWare. Engineers can easily move between programs, since they are immediately knowledgeable of the design process and do not require retraining.

4.2 Lessons Learned

As might be expected, after several years of working with RangeWare, the software staff has collected a list of things that they would do differently if and/or when the opportunity arises. Several of these are in the category of things that work, but could be done better—perfective maintenance. Fortunately the architecture is partitioned such that this is possible. There are at least a few things that don’t work the way one would expect, but that have well-documented workarounds. In most cases, these kinds of improvements will have to wait for a project that absolutely needs them, as there is currently no general purpose “sustainment” budget for the product line.

The design of the RangeWare API illustrates the changes that have occurred leading to possible reengineering. NUWC set several desirable quality attributes for RangeWare. Of these, several are derived from expectations that after deployment, the underlying OS, language, or vendor might change. As is the case with most legacy systems, RangeWare may need to be re-hosted on obsolete hardware, not supported by a vendor. This is the case with many of the range legacy systems. Tremendous effort is expended effecting a port, in order to change all dependencies. Sometimes the changes are so extensive that it is more cost effective to build an entirely new system (NUWC has experienced this, as well as other range developers, on a cycle of between about 8 and 20 years.) Even compiler version changes can often require changes to the systems.

To avoid these difficulties, the original RangeWare development called for language and platform independence. NUWC developed the RangeWare API with features running on a "virtual machine." The intent was to create individual language bindings and hardware interfaces, avoiding any assumed underlying implementation. At the time, a Java implementation appeared to satisfy platform independence, but NUWC was unsure how long this would be the case (Java was at the time relatively new and unproven in this problem domain). They opted for a language independent set of interfaces to perform many operations that are directly supported in Java. The API, for example, provides a means to get to object attributes and methods. This feature is built into Java, so why create an API that builds an additional layer above the implementation?

Since those decisions, Java is now pervasive and is adequate in meeting performance requirements. So far, no user has required C++ or other bindings. Platform independence remains a goal, and Java largely supports this. Java will probably outlive the lifetime of any RangeWare user system, so Java is no longer a high-risk path to platform independence.   The utility of "hiding" some of Java's built-in features behind a "generic" API might be questioned if the decision were revisited today, although a separation of concerns has validity in its own right. Designers today would assume languages could handle the layering overhead with little/no performance penalty. Nonetheless, although some of the architects and programmers would love to rebuild and simplify the interface, there is no driving need for this. Customers would prefer effort going into the building of user-visible applications. Customers never see the API and such changes require a sponsor for the reengineering task.

4.3 Lessons for the DoD

NUWC's success with RangeWare can provide lessons to other DoD organizations considering adoption of a product line approach:

- Plan the effort to deliver immediate benefits. The product line effort must have a short startup phase, low upfront costs, and deliver tangible products to customers. It can expand scope of coverage of the product line through planned evolutionary steps. This will allow the product line effort to sell itself to program managers who must achieve results and reduce costs.
- Build on existing relationships. Work within ongoing programs and build products that will have immediate application. Let the programs contribute to the asset base. Don't take a "build it and they will come" approach.
- Establish clear business goals and architectural drivers. Address these drivers in the implementation. For example, a driver for RangeWare is the ability to change sources of input during a range exercise. NUWC developed the architecture for RangeWare to address this driver so that customers do not need direct access to modify the architecture.

- Define the goals for routine operations. For RangeWare, NUWC is the supplier and ranges are the customers. This model is similar to a contractor supplying products built from assets. This model may pose a risk for the DoD—whether the DoD can select a contractor to be the sole provider.
- Early product line applications should show robust user interfaces and functionality, not merely showcase the underlying technology within the product line. An effective product line requires a strong underlying architecture, but architecture alone will not sell a system, and is, in fact, of little interest to a large segment of end-users.

References

Brownsword 96 Brownsword, Lisa & Clements, Paul. *A Case Study in Successful Product Line Development* (CMU/SEI-96-TR-016, ADA315802). Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, 1996. <http://www.sei.cmu.edu/publications/documents /96.reports/96.tr.016.html>.

Clements 02 Clements, Paul & Northrop, Linda M. *Software Product Lines: Practices and Patterns*. Addison-Wesley, 2002.

Clements 00 Clements, Paul & Northrop, Linda. *A Framework for Software Product Line Practice - Version 3.0* [online]. Pittsburgh, PA: Software Engineering Institute, 2000. <http://www.sei.cmu.edu/plp/frame_report/productLPAs.htm>.

Cohen 99 Cohen, Sholom. *Guidelines for Developing a Product Line Concept of Operations* (CMU/SEI-99-TR-008, ADA367714). Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, 1999. <http://www.sei.cmu.edu/publications/documents /99.reports/99tr008/99tr008abstract.html>.

Dunn 00 Dunn, Edward P. & Rumford, George. "TENA: A Domain-Specific Architecture For Live Participants" [online]. Test Instrumentation Workshop, March 27-30, 2000. Lancaster, CA. <http://www.edwards.af.mil/itea/papers00.htm>.

Noseworthy 02 Noseworthy, J. Russell. "IKE 2—Implementing the Stateful Distributed Object Paradigm,"44–53. *Proceedings of ISORC 2002, 5th International Symposium on Object Oriented Real-Time Distributed Computing.* Washington, DC, April 29 - May 1, 2002. Los Alamitos, CA: IEEE Computer Society, 2002.

SEI 00 Software Engineering Institute. *Capability Maturity Model - Integrated for Systems Engineering/Software Engineering, Staged Representation, V1.0* (CMU/SEI-00-TR-018, ADA388775). Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, 2000. <http://www.sei.cmu.edu/publications/documents/00.reports/00tr018.html>.

SISO 00 Simulation Interoperability Standards Organization. “Simulation Interoperability Workshop Reading List” [online]. <http://www.sisostds.org/siw/00spring/readlist.htm>.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY
(Leave Blank)

2. REPORT DATE
September 2002

3. REPORT TYPE AND DATES COVERED
Final

4. TITLE AND SUBTITLE
Successful Product Line Development and Sustainment: a DoD Case Study

5. FUNDING NUMBERS
F19628-00-C-0003

6. AUTHOR(S)
Sholom Cohen, Ed Dunn, Albert Soule

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Software Engineering Institute
Carnegie Mellon University
Pittsburgh, PA 15213

8. PERFORMING ORGANIZATION REPORT NUMBER
CMU/SEI-2002-TN-018

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
HQ ESC/XPK
5 Eglin Street
Hanscom AFB, MA 01731-2116

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

12A DISTRIBUTION/AVAILABILITY STATEMENT
Unclassified/Unlimited, DTIC, NTIS

12B DISTRIBUTION CODE

13. ABSTRACT (MAXIMUM 200 WORDS)

The Engineering, Test, and Evaluation Department of the Naval Undersea Warfare Center – Division Newport (NUWC) has developed a software product line asset base, named RangeWare, to support test range operations. NUWC has also fielded a product line of range systems using the asset base. RangeWare provides an object services architecture to support integration of sensor and other range data for analysis and display by range equipment. After several pilot applications of RangeWare, NUWC is now taking RangeWare into a sustainment phase, expanding the coverage of the asset base in terms of object and distribution services as well as applying the assets to new systems.

This case study describes RangeWare and NUWC's product line practices to sustain and support the evolution of RangeWare. These practices include "Operations," "Data Collection," "Metrics and Tracking," "Software System Integration," "Configuration Management," "Tool Support," "Structuring the Organization," "Building a Business Case," and others. The case study also examines NUWC's lessons learned and its plans for improved process definition for RangeWare product production.

14. SUBJECT TERMS
asset base, software product line, case study

15. NUMBER OF PAGES
54

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT
Unclassified

18. SECURITY CLASSIFICATION OF THIS PAGE
Unclassified

19. SECURITY CLASSIFICATION OF ABSTRACT
Unclassified

20. LIMITATION OF ABSTRACT
UL