NPS-AM-09-015

EXCERPT FROM THE PROCEEDINGS

OF THE

SIXTH ANNUAL ACQUISITION RESEARCH SYMPOSIUM

ONTOLOGY-BASED SOLUTIONS FOR SOFTWARE REUSE

Published: 22 April 2009

by

Jean Johnson and Curtis Blais

6th Annual Acquisition Research Symposium of the Naval Postgraduate School:

Volume I: Defense Acquisition in Transition

May 13-14, 2009

Prepared for: Naval Postgraduate School, Monterey, California 93943

ACQUISITION RESEARCH PROGRAM
GRADUATE SCHOOL OF BUSINESS & PUBLIC POLICY
NAVAL POSTGRADUATE SCHOOL

The research presented at the symposium was supported by the Acquisition Chair of the Graduate School of Business & Public Policy at the Naval Postgraduate School.

To request Defense Acquisition Research or to become a research sponsor, please contact:

NPS Acquisition Research Program
Attn: James B. Greene, RADM, USN, (Ret)
Acquisition Chair
Graduate School of Business and Public Policy
Naval Postgraduate School
555 Dyer Road, Room 332
Monterey, CA 93943-5103
Tel: (831) 656-2092
Fax: (831) 656-2253
E-mail: jbgreene@nps.edu

Copies of the Acquisition Sponsored Research Reports may be printed from our website www.acquisitionresearch.org

Conference Website:
www.researchsymposium.org

Proceedings of the Annual Acquisition Research Program

The following article is taken as an excerpt from the proceedings of the annual Acquisition Research Program. This annual event showcases the research projects funded through the Acquisition Research Program at the Graduate School of Business and Public Policy at the Naval Postgraduate School. Featuring keynote speakers, plenary panels, multiple panel sessions, a student research poster show and social events, the Annual Acquisition Research Symposium offers a candid environment where high-ranking Department of Defense (DoD) officials, industry officials, accomplished faculty and military students are encouraged to collaborate on finding applicable solutions to the challenges facing acquisition policies and processes within the DoD today. By jointly and publicly questioning the norms of industry and academia, the resulting research benefits from myriad perspectives and collaborations which can identify better solutions and practices in acquisition, contract, financial, logistics and program management.

For further information regarding the Acquisition Research Program, electronic copies of additional research, or to learn more about becoming a sponsor, please visit our program website at:

www.acquistionresearch.org

For further information on or to register for the next Acquisition Research Symposium during the third week of May, please visit our conference website at:

www.researchsymposium.org

THIS PAGE INTENTIONALLY LEFT BLANK

Ontology-based Solutions for Software Reuse

Presenter: Ms. Jean M. Johnson is a Lecturer in the Systems Engineering Department at the Naval Postgraduate School, Monterey, California. After serving on active duty in the US Navy, she supported the NAVSEA Warfare Systems Engineering Directorate (NAVSEA06) before coming to Naval Postgraduate School. Her current research focus areas are software repositories to enable reuse and the use of modeling and simulation in DoD acquisition. She holds an ME in Operations Research and Systems Analysis and a BS in Applied Mathematics from Old Dominion University and is currently a PhD candidate in Software Engineering.

Jean M. Johnson
Systems Engineering Department
Naval Postgraduate School
2669 Alamance Circle
Virginia Beach, VA 23456
Tel: 757-689-4492
E-mail: jmjohnso@nps.edu

Author: Curtis Blais is a Research Associate Professor with the Naval Postgraduate School Modeling, Virtual Environments, and Simulation (MOVES) Institute. His primary areas of research and development include application of semantic web technologies to improve interoperability and for identifying and delivering valued information in network-centric environments such as the Global Information Grid. Blais earned Bachelor of Science and Master of Science degrees in Mathematics from the University of Notre Dame.

Curtis Blais
Modeling, Virtual Environments and Simulation Institute
Naval Postgraduate School
Monterey, CA 93943-5000
Tel: 831-656-3215
Fax: 831-656-7599
E-mail: clblais@nps.edu

Abstract

The commonly recognized benefits of software reuse are increased productivity, higher quality, shorter time-to-market, and reduced development and maintenance costs. Software reuse is a key thrust of DoD acquisition improvement initiatives including the Naval Open Architecture program. Successful reuse depends on many aspects of a reuse program, ranging from organizational climate to technical solutions. As technical solutions, current software repositories do not provide robust search and discovery capabilities due to limitations of current information organization practices.

This research explores potential solutions that are enabled when ontologies are used as the framework for information contained in the software repository. In this paper, we will briefly summarize previous work on an ontology-based repository framework. We will then present current efforts to specify a software repository tool that exploits the framework to enable more sophisticated search and discovery.

The suggested tool will emphasize human interaction and allow users to bring their context to the search process. New navigation techniques will be employed that guide human

users, offering suggestions based on projected needs. The improved search capability will encourage developers to consider reuse and aid in its success.

Introduction

In August 2006, Program Executive Officer of Integrated Warfare Systems (PEO-IWS) established the Software Hardware Asset Reuse Enterprise (SHARE) repository to enable the reuse of combat system software and related assets. In July 2007, the Naval Postgraduate School (NPS) was tasked to develop a component specification and ontology for the SHARE repository.

A description of SHARE and the requirements for a component specification and ontology supporting this repository are available in Johnson (2008). A vision of the component specification and ontology for the repository framework, a brief survey of initiatives and technologies relevant to desired repository capabilities, a development approach, and initial design are described in Johnson and Blais (2008, March). In Johnson and Blais (2008, September) we provided the initial component specification and ontology for the repository framework, as well as initial information models supporting future implementation of stronger semantic representations of assets and artifacts in the repository.

This paper and presentation summarize the previous work and discuss the current research being conducted, which will result in a requirements specification for improved software repository tools.

Repository Framework

In Johnson and Blais (2008, March), we proposed a repository framework for SHARE, consisting of two major aspects: a component specification and ontology. The component specification is a description or model of the items in the repository and consists of two parts: metadata and software behavior representation. The ontology describes concepts and relationships to create various perspectives or contexts for examining the contents of the repository. These aspects of the framework are discussed below.

1. Component Specification: Metadata

The metadata for each artifact should incorporate all necessary data for discovery and implementation. The metadata will aid repository users in determining if the item is suited for their use and will provide information about how to use the asset when it is retrieved. We refer to this as “standard” or “typical” metadata since there are many existing examples of metadata we can use to develop the metadata for SHARE.

We developed a metadata schema for the SHARE repository and presented its details in Johnson and Blais (2008, September). An initial list of required asset information developed by the SHARE Program Office at Naval Surface Warfare Center, Dahlgren, VA, was used as a starting point. We began by creating an Extensible Markup Language (XML) Schema for this metadata set and then enhanced the schema based on a more current “wizard” that leads a user through the SHARE asset information entry process.

After careful analysis of this initial schema, as well as known metadata examples found in existing software repositories, we began to modify the schema by reorganizing the data and complementing the fields with information that should be included. We also incorporated the necessary information to place each artifact in the appropriate context based on the ontology development. Finally, we evaluated the schema against the minimum requirements of the DoD Discovery Metadata Specification (Deputy Assistant Secretary of Defense, 2007) to promote future exposure of SHARE contents across the DoD Enterprise.

The most significant recommended change to the current SHARE approach to handling metadata is the level of application. It is our assertion that to enable the satisfaction of repository user needs, metadata must be applied at the artifact level rather than at the asset level, which is the current methodology for SHARE.

To be clear, we must provide our definition of these two concepts. The Navy Open Architecture (OA) program has adopted similar definitions for asset and artifact as those used in the Object Management Group (OMG) Reusable Asset Specification (RAS). In the RAS, artifacts are defined as “any work products from the software development lifecycle,” and assets are a grouping of artifacts that “provide a solution to a problem for a given context” (Object Management Group, 2005, p. 7). Accordingly, the RAS describes an approach for packaging artifacts into an asset.

This is consistent with the current SHARE approach and remains consistent in the proposed metadata schema. However, the current SHARE approach is to package artifacts into assets at the convenience of the submitter and to enable the current retrieval process. We believe it is more useful to enable packaging of artifacts into assets based on users’ needs. This means that the grouping of artifacts into an asset should have the capability of being user-defined. In order to enable this approach, the users must be able to discover the artifacts of potential value to their particular context in order to solve a particular problem and then package those artifacts into an asset for retrieval.

Therefore, the proposed metadata schema includes separate definitions of structures for artifacts and assets. This does not preclude the pre-packaging of artifacts into assets for submission to the repository or for extraction to solve common problems. We envision the capability for users to discover a problem solution by either locating a prepackaged (reusable) asset or by building an asset from artifacts they believe will help solve their particular problem.

Splitting the metadata into two schemas, one for assets and another for artifacts, also enables a clearer distinction about the data that needs to be collected for each. For example, the current SHARE metadata collects data on the type of artifacts included in the asset, such as whether they are documents or code. Then, it separately asks for thousands of lines of code (KSLOC) for the asset. This would more likely be tied to particular artifacts that are of the type “code” in the asset. By separating the asset and artifact schemas, we can better distinguish the necessary data for an asset from the necessary data for an artifact, and we will be able to manipulate the data more appropriately with tools that implement the search.

Collecting metadata information for each artifact may seem like a daunting task when compared to the current method. However, it is highly likely that a good portion of the metadata that applies to one artifact also applies to the remaining artifacts in a group of submitted artifacts. The submission tool can be constructed to minimize duplicative entries of data by prompting users to verify that the information being entered applies to all the artifacts in a group. This construction would minimize the individual entries required in the submission and metadata collection process. It is also possible to create tools that automate much of the metadata collection from the artifacts themselves. Other organizations are conducting research and development to auto-generate metadata from the source products. This is a critical capability in making legacy content available for search and discovery. Adoption of structured metadata makes autogeneration feasible, although certainly nontrivial. This is a recommended area for future research and development in the SHARE program.

Artifact Schema

The artifacts schema is designed to be flexible in its implementation. All the elements, types and attributes in the schema are defined globally so they can be reused in other schemas that developers may create for working with artifact information. The root element, *Artifacts*, is simply a container for any number of artifacts contained in a single instance of the schema, as shown in Figure 1. Repository managers and tool designers can decide if they wish to keep a separate XML file describing each artifact or if they prefer to group multiple artifact descriptions into a single XML file.

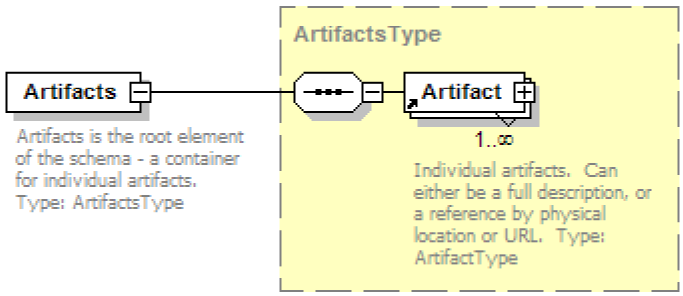

Figure 1. Artifacts Element

The individual descriptions of each artifact are also designed to be flexible. A specific artifact can be incorporated into the file in one of three ways. The first is by providing the full artifact description. This full description represents the heart of the metadata development effort and should be considered the preferred method for representing an artifact. However, if the full description is not available, or if the information required is provided in some other location, the schema allows the inclusion of the artifact representation by reference—either to a physical location or by URL. This is shown in Figure 2.

The full description of each artifact, contained in the element *ArtifactFullDescription*, is composed of eight sub-elements as depicted in Figure 3. Each sub-element is discussed in detail in Johnson and Blais (2008, September).

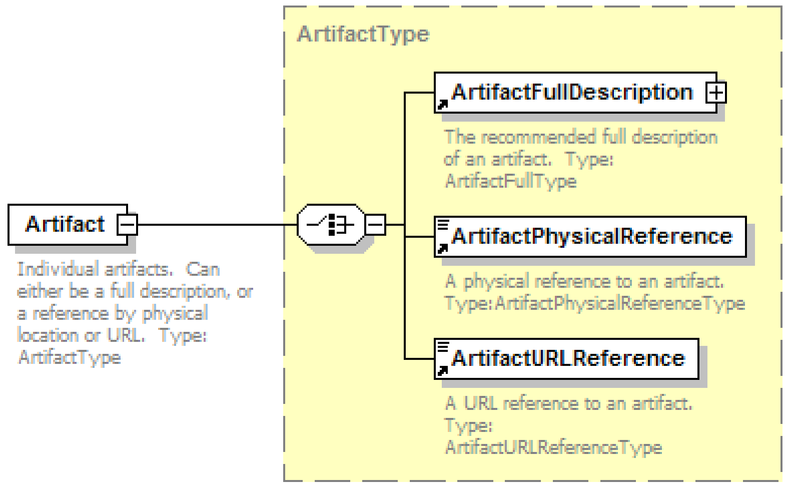

Figure 2. Artifact Element

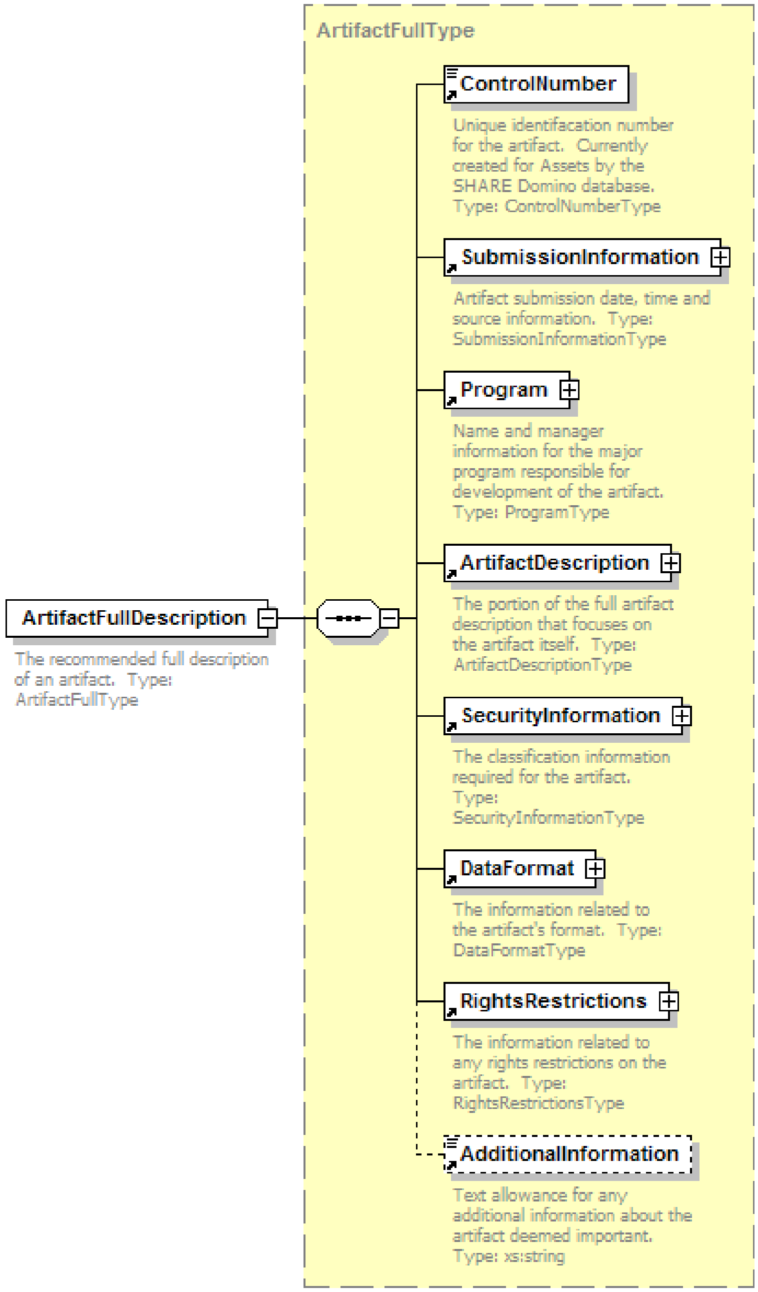

Figure 3. Artifact Full Description Element

Asset Schema

In the preceding description of artifacts, we see that much of the detail about a submission has been moved to the artifact level. The information needed to describe an asset is thus simplified to be primarily an identification of the artifacts contained in the asset. The root element of the assets XML structure is a container for one or more asset records, as shown in Figure 4. The proposed top-level XML structure for an asset is shown in Figure 5.

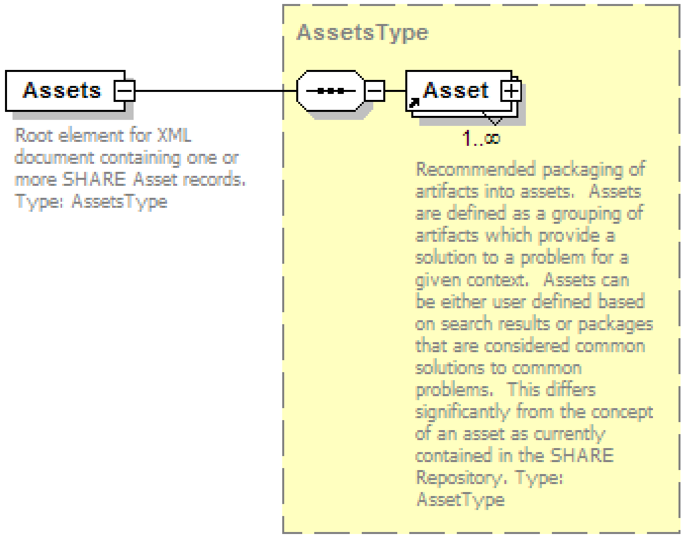

Figure 4. Assets Root Element

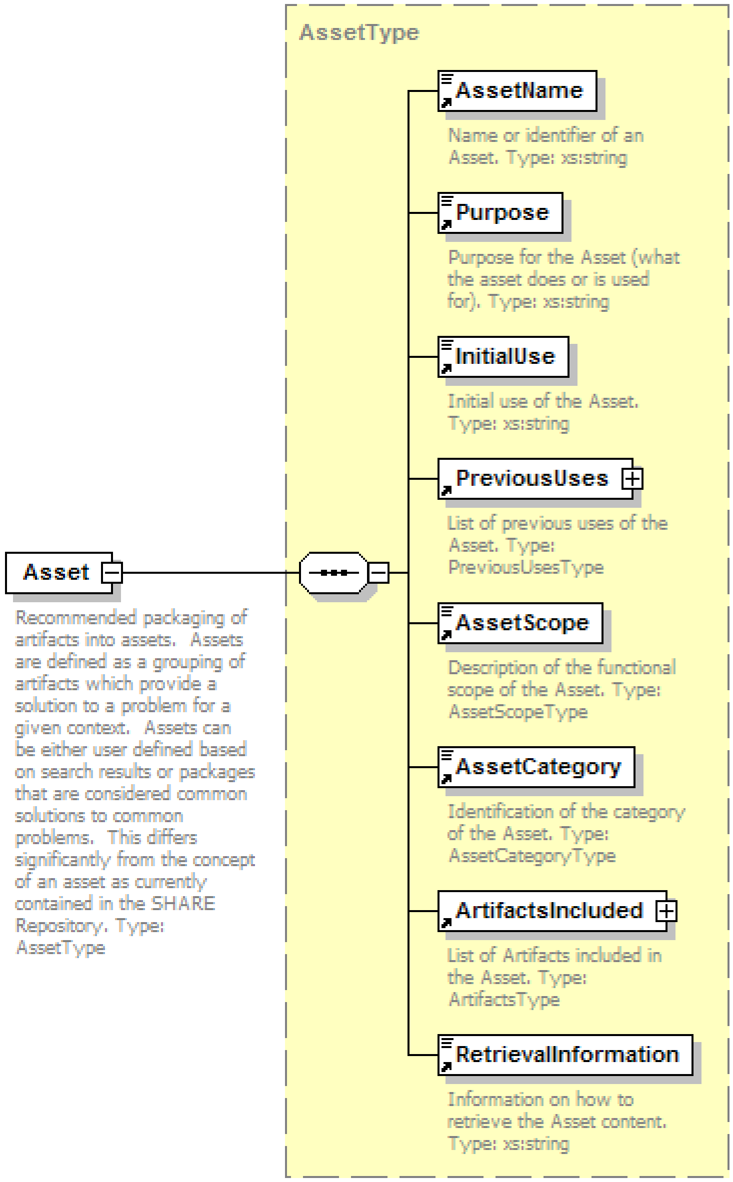

Figure 5. Asset Element

The remaining sub-elements of the asset schema are described in Johnson and Blais (2008, September).

2. Component Specification: Software Behavior

The metadata for many current repositories fail to capture a searchable representation of the behavior of the items outside general categories of functionality (e.g., Archiving Compression Conversion, Control Flow Utilities, Graphics, and Security) and text-based search of code descriptions. Unlike current practice, the SHARE component specification will consist of both typical metadata and a behavioral model of the component. Since this piece of the component specification is not commonly incorporated into repositories in a standardized manner, we feel it is a specific focus area to identify the appropriate representation mechanisms for software behavior in the repository context.

One of the loftier goals of a software repository is to support automatic composition of systems from reusable components. This is a difficult problem, which many have tried to solve.[1] It is especially difficult if the components were not originally designed for reuse. As a necessary first step towards more sophisticated uses of a repository, behavioral descriptions must be machine-readable in order to support automated search and discovery. Furthermore, the behavior descriptions must be formalized and consistently applied to each item in the repository if the intent is to automatically compose them into a larger functioning system.

In our efforts towards standardized specification of software behavior for the SHARE repository, we have sought a balance between method robustness and ease of implementation. Each type of presented representation offers advantages for certain purposes. However, it is recognized that the array of contributors to SHARE requires caution in dictating standards that will impact the development processes of the asset developers.

We explored characterization of software interfaces based on current and emerging Web Services (e.g., WSDL) and Semantic Web Services (e.g., WS-BPEL, OWL-S) approaches. However, the work is preliminary, since the current approach to describing code artifacts making up an asset is extremely limited. It will be necessary to adopt a more precise description of code artifacts to introduce these techniques. As a start, we included the option of inserting a WSDL description of software services in the *SoftwareBehaviorDescription* element.

We also proposed a near-term solution that uses domain information to standardize descriptions of software functionality; namely, the well-established Common System Function List (CSFL).[2] We developed a taxonomy based on the CSFL and incorporated fields into the metadata (XML schema) that will assign functions to repository items. If we require asset submitters to state the functionality of the components in these terms, we can then build the tools to guide users in selecting desired behavior in the same terms.

The CSFL was captured in an OWL structure to use as an initial characterization of software behavior. The process by which the taxonomy was generated is a good example of methods for creating a practical set of structured data from initial raw formats. The taxonomy was constructed from a Microsoft Excel spreadsheet (CSFL version 3.0). The spreadsheet provided definitions of the domains and functions, identified what the domain or function is derived from and identified sources of the definitions. Microsoft Excel provides the capability to export the content of the spreadsheet to XML format. A simple Extensible Stylesheet Language for Transformations (XSLT) was written to transform the source XML format (spreadsheet data) to a target XML format (OWL). The transformation created a simple class/subclass hierarchy expressed in OWL. A portion of the resulting OWL structure is shown in the Protégé ontology editing tool in Figure 6.

---

[1] The proceedings from the International Symposium on Software Composition, an annual event, provide examples of research into the breadth of research topics currently being pursued in the area of software composition. The website for the 2008 conference is located at http://www.2008.software-composition.org/

[2] DoD Warfighter Service Components in the DoD Enterprise Architecture Service Component Reference Model are derived from the DoN CSFL.

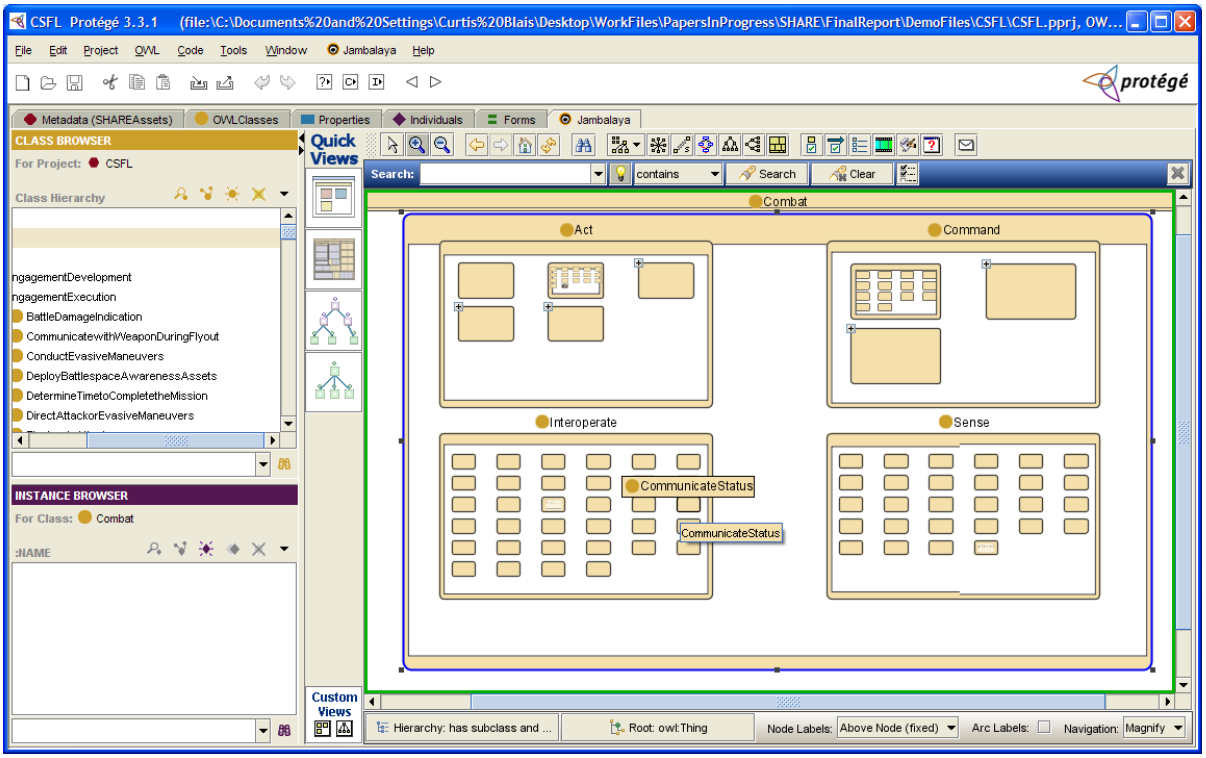

Figure 6. Portion of the CSFL Taxonomy Displayed in Protégé under the Jambalaya Graphics Tab

Other similar lists have been developed for operational activities (i.e., the Common Operational Activities List (COAL)) and for information elements (Common Information Element List (CIEL)). It may be valuable to also capture these in OWL classes and then create interrelationships across the classes (e.g., what information elements are generally employed in performing certain system functions and what information elements are generally produced by performing certain system functions, etc.). Further exploration with subject-matter experts is needed to determine potential benefit from such approaches.

Although we cannot solve the software composition problem in the near-term, initial descriptions of software behavior through identification of functionality and specification of interfaces are necessary steps toward that capability. These intermediate steps toward formalized behavior descriptions will prove useful in the near-term and helpful in advancing long-term goals.

3. Ontology of Framework Relationships

The framework ontology includes descriptions of the component relationships to form a contextual model of the repository items.[3] These relationships may include the component's use/role in existing systems, its mapping to reference or domain architectures, and its utility in various software development lifecycle phases. Contextual information about the artifact can be

---

[3] Throughout the document, *ontology* is used as a general term for describing concepts and relationships among concepts, with *taxonomy* as a special case in which the classes in the ontology are related by a single property, such as "is-a" or "has-a."

exploited to enable sophisticated search and discovery methods that more closely match recommended retrieval items to a user's problem context.

Assets and artifacts in the SHARE repository can be examined from a number of perspectives, reflecting a variety of associations. We chose to create initial classification schemes that can provide benefit in the near-term. The resulting taxonomies and ontologies are meant to be illustrative, not exhaustive. The taxonomies/ontologies we developed for SHARE are based on several types of relationships between the items in the repository, as well as with relevant domain architectural descriptions and other information. They capture an artifact's place in the software engineering lifecycle (see Figure 7), its architectural fit in its original system (see Figure 8), its architectural fit in any system in which it was subsequently used, identification of the component's fit in the Surface Navy Objective Architecture (see Figure 9), and the semantic relationships of various documents in the repository. Each of these ontologies is discussed in detail in Johnson and Blais (2008, September).

This enriched semantic specification of the assets in the SHARE repository will enable users to more readily find resources that meet their needs in their context. Extensive work in the Web community is providing tools and techniques that can be applied to the framework when it is based on these ontologies. We have created an initial semantic foundation on which enhanced capabilities can be implemented.

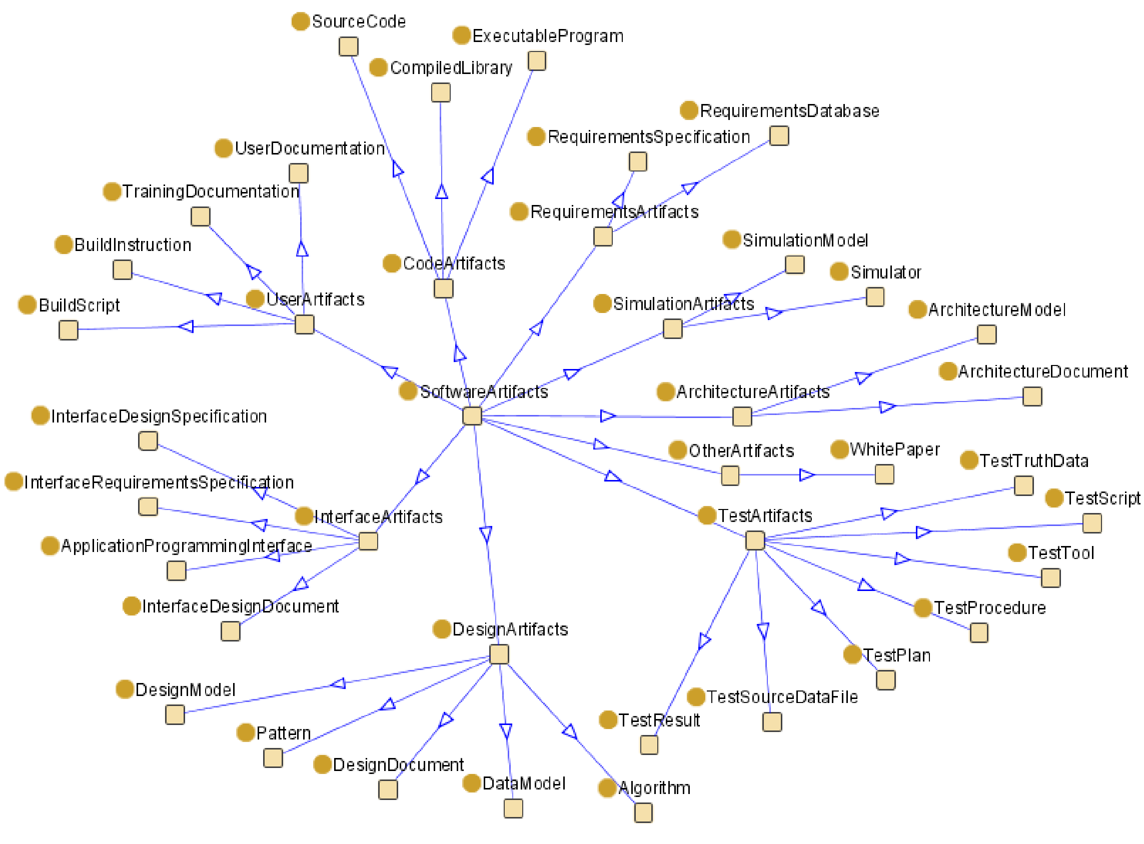

Figure 7. Software Artifact Taxonomy

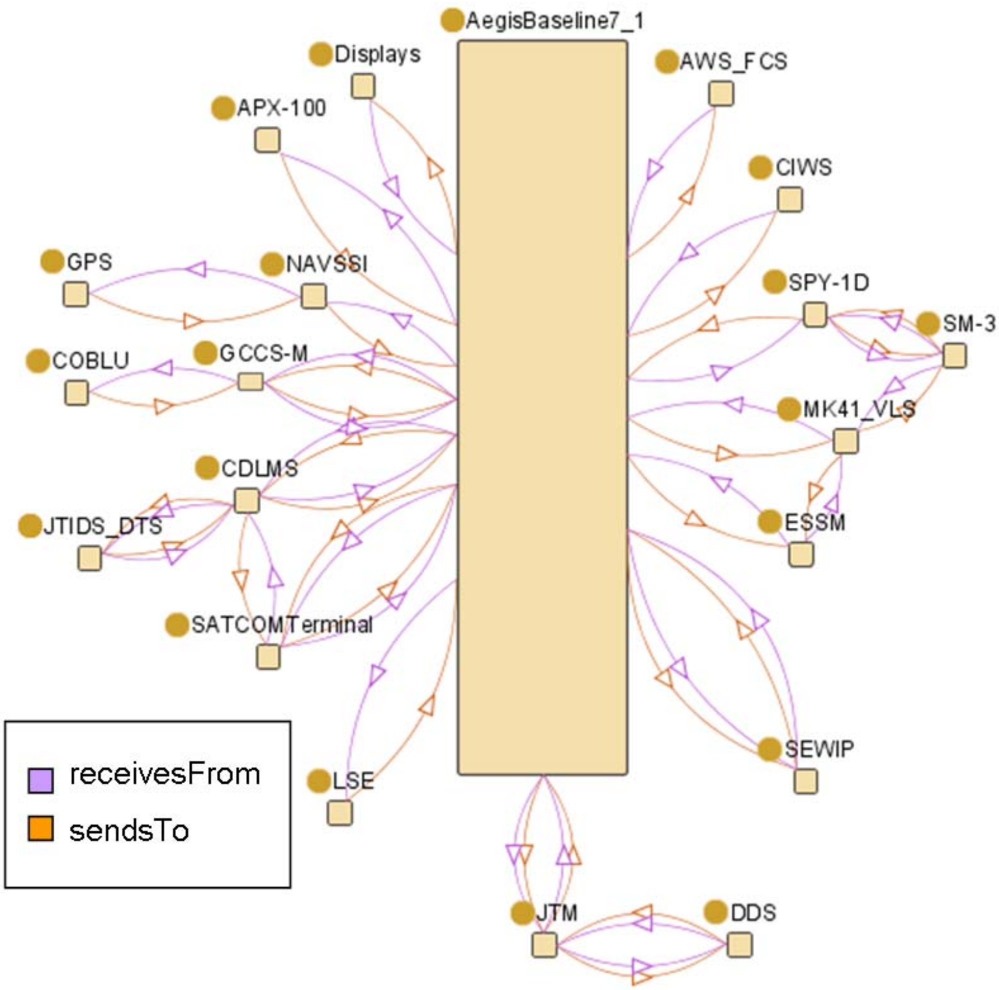

Figure 8. System Ontology Example (AEGIS)

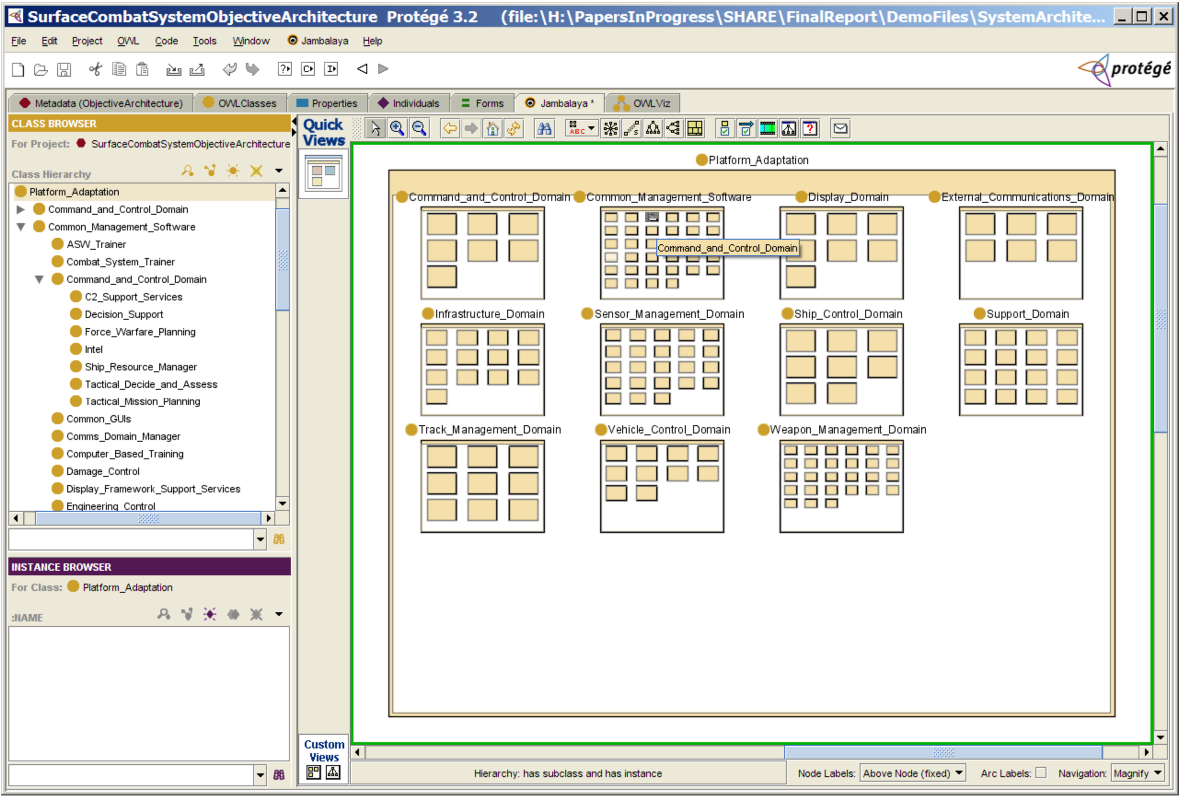

Figure 9. Surface Combat System Top-level Objective Architecture Described as a Taxonomy in OWL (Jambalaya Graphic Tab in Protégé)

Current Research

Current research efforts focus on designing repository tools that allow for guided navigation of artifacts in software repositories. These tools will take advantage of the improved repository framework developed during the previous effort. The value of the repository tools will be demonstrated through use case demonstrations, sponsor evaluations, and a focus group study.

The results will be detailed requirements specifications for user tools associated with the new repository framework, including specifications for both the repository user interface tool as well as the asset-submission tool. The repository user interface tool will enable multiple views of repository contents for improved search efficiency. The tool will be open-ended to allow extension based on the domain knowledge of the repository manager and users. The asset-submission tool will aid software developers in properly describing and characterizing items as they are submitted into a repository. When implemented as a repository system, these products will enable sophisticated search and discovery of reusable artifacts and maintenance of the repository, which will improve the current state-of-the-art.

Summary

Each piece of the repository framework enhances the search capabilities in different ways. The basic metadata in the XML schemas provides search criteria for finding components of interest in the repository as well as specific information about the artifacts in order to determine if they are appropriate for retrieval. OWL taxonomies and ontologies enable identification of functionality and associated resources that may be beneficial to users. In short:

- The metadata is evaluated to enable retrieval decisions.
- The software behavior representations enable searches based on functionality.
- The ontologies point users to helpful artifacts they may not have initially considered.

The current efforts will result in designs for repository tools that will take full advantage of the repository framework to enable guided search and discover as well as asset submission.

List of References

Johnson, J. (2008, March). *SHARE repository component specification: Needs assessment* (Technical Report NPS-AM-08-015). Monterey, CA: Naval Postgraduate School.

Johnson, J., & Blais, C. (2008, March). *Software hardware asset reuse enterprise (SHARE) framework: Related work and development plan* (Technical Report NPS-AM-08-021). Monterey, CA: Naval Postgraduate School.

Johnson, J., & Blais, C. (2008, September). *Software hardware asset reuse enterprise (SHARE) repository framework final report: Component specification and ontology* (Technical Report NPS-AM-08-134). Monterey, CA: Naval Postgraduate School.

Deputy Assistant Secretary of Defense. (2007). *Department of Defense discovery metadata specification (DDMS)* (Version 1.4.1). Washington, DC: Deputy Chief Information Office.

Object Management Group. (2005). *Reusable asset specification, Version 2.2.* Retrieved January 29, 2008, from http://www.omg.org/technology/documents/formal/ras.htm

SourceForge. (2007). *SourceForge.net: Welcome to SourceForge.net.* Retrieved October 8, 2007, from www.sourceforge.net

2003 - 2009 Sponsored Research Topics

Acquisition Management

- Acquiring Combat Capability via Public-Private Partnerships (PPPs)
- BCA: Contractor vs. Organic Growth
- Defense Industry Consolidation
- EU-US Defense Industrial Relationships
- Knowledge Value Added (KVA) + Real Options (RO) Applied to Shipyard Planning Processes
- Managing Services Supply Chain
- MOSA Contracting Implications
- Portfolio Optimization via KVA + RO
- Private Military Sector
- Software Requirements for OA
- Spiral Development
- Strategy for Defense Acquisition Research
- The Software, Hardware Asset Reuse Enterprise (SHARE) repository

Contract Management

- Commodity Sourcing Strategies
- Contracting Government Procurement Functions
- Contractors in 21st Century Combat Zone
- Joint Contingency Contracting
- Model for Optimizing Contingency Contracting Planning and Execution
- Navy Contract Writing Guide
- Past Performance in Source Selection
- Strategic Contingency Contracting
- Transforming DoD Contract Closeout
- USAF Energy Savings Performance Contracts
- USAF IT Commodity Council
- USMC Contingency Contracting

Financial Management

- Acquisitions via leasing: MPS case
- Budget Scoring
- Budgeting for Capabilities-based Planning
- Capital Budgeting for DoD

- Energy Saving Contracts/DoD Mobile Assets
- Financing DoD Budget via PPPs
- Lessons from Private Sector Capital Budgeting for DoD Acquisition Budgeting Reform
- PPPs and Government Financing
- ROI of Information Warfare Systems
- Special Termination Liability in MDAPs
- Strategic Sourcing
- Transaction Cost Economics (TCE) to Improve Cost Estimates

Human Resources

- Indefinite Reenlistment
- Individual Augmentation
- Learning Management Systems
- Moral Conduct Waivers and First-tem Attrition
- Retention
- The Navy's Selective Reenlistment Bonus (SRB) Management System
- Tuition Assistance

Logistics Management

- Analysis of LAV Depot Maintenance
- Army LOG MOD
- ASDS Product Support Analysis
- Cold-chain Logistics
- Contractors Supporting Military Operations
- Diffusion/Variability on Vendor Performance Evaluation
- Evolutionary Acquisition
- Lean Six Sigma to Reduce Costs and Improve Readiness
- Naval Aviation Maintenance and Process Improvement (2)
- Optimizing CIWS Lifecycle Support (LCS)
- Outsourcing the Pearl Harbor MK-48 Intermediate Maintenance Activity
- Pallet Management System
- PBL (4)
- Privatization-NOSL/NAWCI
- RFID (6)
- Risk Analysis for Performance-based Logistics
- R-TOC Aegis Microwave Power Tubes

- Sense-and-Respond Logistics Network
- Strategic Sourcing

Program Management

- Building Collaborative Capacity
- Business Process Reengineering (BPR) for LCS Mission Module Acquisition
- Collaborative IT Tools Leveraging Competence
- Contractor vs. Organic Support
- Knowledge, Responsibilities and Decision Rights in MDAPs
- KVA Applied to Aegis and SSDS
- Managing the Service Supply Chain
- Measuring Uncertainty in Earned Value
- Organizational Modeling and Simulation
- Public-Private Partnership
- Terminating Your Own Program
- Utilizing Collaborative and Three-dimensional Imaging Technology

A complete listing and electronic copies of published research are available on our website: www.acquisitionresearch.org

THIS PAGE INTENTIONALLY LEFT BLANK

ACQUISITION RESEARCH PROGRAM
GRADUATE SCHOOL OF BUSINESS & PUBLIC POLICY
NAVAL POSTGRADUATE SCHOOL
555 DYER ROAD, INGERSOLL HALL
MONTEREY, CALIFORNIA 93943

www.acquisitionresearch.org