An Open Systems Perspective on Horizontal Reuse

Marilyn T. Gaska

IBM Federal Systems Company
MD 0124, Owego, NY 13827
Tel: (607) 751-4156, fax: (607) 751-3727
Email: mtgaska@vnet.ibm.com

Abstract

From an application programmer's perspective, there are many similarities between the problems of working with previously developed code in a reuse library and Commercial Off The Shelf (COTS) software. In particular, open-systems-compliant COTS platform infrastructure services can be considered to be horizontal reuse of the "blackest" box category since knowledge of the contents of the box or component may not even be available at any level of the organization. This position paper discusses similarities between the goals and needs of open systems in the federal market place and horizontal reuse concepts. A stronger synergy between the two areas of research is proposed.

Keywords: Horizontal Reuse, Open Systems, OSE Profile.

Workshop Goals: Broaden perspective on horizontal reuse by leveraging experience in open systems environment (OSE) services to explore synergism between reuse and OSE efforts to solve the "software crisis". Work with recognized leaders in current reuse issues, methods, and metrics to learn approaches for potential application to open systems infrastructure design. Share experiences with others on efforts to institutionalize reuse.

Working Groups: Horizontal Reuse, Domain Analysis, Institutionalizing Reuse.

1 Background

As the current IBM Federal Systems Company Owego Software Reuse Council Chair, the author is directly involved in assisting the institutionalization of reuse in a corporate environment. While she has had previous software development experience on a large Computer Aided Design (CAD) development project, her most recent position is in the area of open systems technology and Commercial Off the Shelf (COTS) software integration. She is currently the OSE Profile task lead for the Army Sustaining Base Information Services (SBIS) contract. This task involves interfaces with the National Institute for Standards and Technology (NIST) and other federal agencies. The author has also applied OSE perspectives to concurrent engineering infrastructure concepts as follow up to her work on an IBM Owego Concurrent Engineering Quality Improvement Team ([1]).

2 Position

HO: Open Systems Infrastructure ≈ Horizontal Reuse

Horizontal reuse generally addresses reusable services across application domains whereas vertical reuse is considered specific to a domain. Both are needed to address the total application environment. The movement toward open systems can be considered to place specific constraints on horizontal reuse of infrastructure services. Open systems precludes use of proprietary service interfaces by specifying use of consensus-based open systems standards to improve application portability, interoperability and scalability. In general, COTS services are selected that have been validated against these standards. COTS products can be considered a form of black box reuse. The requirements of the specific application domain determine the subset of infrastructure service standards required to support application development.

Open systems technology and horizontal reuse initiatives address some of the same problems in general design as well as the "software crisis". If design is genericly defined as the process of removing ambiguity [2], then the open systems consensus-based process can be considered to have already constrained the software infrastructure design in the same way as a requirement to use black box software from a reuse library. In fact, open system places constraints across any horizontal reuse for infrastructure services. These components can be used to architect software systems, with various architecture specialists concentrating on domains such as management information systems, embedded realtime software applications, and computer integrated manufacturing. By considering horizontal reuse and OSE software technology as closely related answers to improving software productivity and reducing life cycle costs, research and resources can be leveraged to solve common problems of configuration management, classification, metrics, and application programmer use.

2.1 Open Systems Environment Services

The NIST Application Portability Profile (APP) [3] defines an OSE as a "computing environment that supports portable, scalable, and interoperable applications through standard services, interfaces, data formats, and protocols. The standards may consist of international, national, industry, or other open (public) specifications. These specifications are available to any user or vendor for use in building systems and products that meet OSE criteria." Furthermore, "an OSE supports applications through the use of well-defined components: a plug-compatible technology or building-block approach for developing systems."

The NIST APP provides an OSE reference model closely aligned with the Institute of Electrical and Electronics Engineers (IEEE) POSIX Working Group P1003.0 [4]. This reference model defines applicaton software, application platform, and external environment entities, as well as interfaces between entities to include application program interface (API) and external environment interface (EEI). The application platform entity closely parallels the concept of horizontal reuse by providing the following services:

- **Operating System** – kernal operations, commands and utilities, and realtime extension;
- **Human / Computer Interface** – client-server operations, object definitiona and management, window management, dialog support, and multimedia;
- **Software Engineering** – programming languages and language bindings and integrated software engineering envionrments (ISEE) and tools;
- **Data Management** – data dictionary / directory, database management system (DBMS), and distributed data;
- **Data Interchange** – document services, graphics data, and product data interchange;
- **Graphics** – services for creating and manipulating displayed images;
- **Network** – data communication, transparent file access, personal / micro computer support, and remote procedure call (RPC);
- **Integral Support (Security and System Management)** – security for all of the other service areas, fault mangement and control, configuration control, accounting, and performance monitoring.

The NIST APP concepts have also been used as the basis for other documents from the Defense Information Systems Agency (DISA). These include the three volume series on Technical Architecture Framework for Information Systems (TAFIM) that addresses architecture, implementation concepts, and standards profile [5], [6], [7]. Other documents from the Department of Defense (DoD) recommend OSE standards profiles for acquisitions, where a profile is a set of specifications that define a coherent and complementary set of services made availalbe to applications in a specific application domain. The application profile is the set of service requirements for the domain that drives the selection of the specifications. Therefore, the OSE profile for a project adds additional constraints on horizontal reuse of infrastructure services. The DISA/CIM Software Reuse Program has even proposed metrics for the coverage of reuse libraries to the TAFIM as well as for horizontal and vertical reuse support [8].

2.2 Generic Issues for OSE and Reuse

The NIST APP reference model and service classification can be used as a framework for classifying all infrastructure software targeted for application access as a horizontal reuse service whether or not OSE or proprietary specifications are selected. It provides common terminology and definitions to support discussion. The profile concepts of harmonization and coherence among services can be applied as well. Since there is a fuzzy line between domain analysis and the application profile subsetting of OSE specifications, the OSE service categories may also clarify vertical reuse approaches.

There is a list of generic operational and organizational issues for use of COTS OSE services and reuse repositories of developed software.

- **Service Information Organization** – how is the service information organized / classified?
- **Service Search and Retrieval** – how to find OSE COTS or any reuse service specification and access the code?
- **Dependency Management** – how to assure that dependencies among components and the parent / child dependencies within a component are addressed?
- **Management and Support Organization** – what are the inter or intraproject personnel and information organization, processes, and metrics?
- **Service Information Education and Communication** – how do application developers for a specific domain learn about the available services?
- **Tool Support** – how does the ISEE support access to the service information,
- **Configuration Management and Maintenance** – how are changes in versions addressed within and across domains?
- **Architecture Compliance Checking** – how is service usage inforced (automated vs. manual)?
- **Component Interoperability Testing** – how is testing done among selected components to assure verify harmony?

Furthermore, the technical implication of the application of object-oriented vs. functional approaches to service definition are significant to both OSE and horizontal reuse. In the OSE area, the Object Management Group (OMG) is addressing the issue through the Common Object Request Broker Architecture (CORBA) specification [9]. In general, the application software approach and specific development methods should be consistent with the service specification / representation methods, a technical issue also addressed by others [10].

2.3 Research Needs

The position of this author is that there may already be significant research done under the name of OSE technology or reuse that can be identified for potential application today. In addition, there is potential financial savings and improved return on investment (ROI) for current initiatives if these efforts are linked. For instance, issues of interoperability among repositories are already being addressed the the Reuse Interoperability Group (RIG) standardization efforts. This effort can be classified as a data interchange problem within the NIST APP framework.

However, while there is significant research already in areas of classification, configuration management and metrics; some of the hotter topics fall in the area of application developer access in conjunction with application rapid prototyping approaches. When prototyping is used in conjunction with simulation, preliminary infrastructure interoperability testing can be done early in the process. This prototyping and simulation can support architecture tradestudy analysis.

3 Comparison

There are several examples of efforts described in the literature that have already begun either addressing OSE and reuse issues or that reflect the overlap in the targeted problems. Several key examples from the commercial and federal segments include: AT&T BaseWorkX; Genesis and Avoca projects [11], Portable, Reusable, Integrated Software Modules (PRISM) and Central Archive for Reusable Defense Software (CARDS) [12]; and the Software Technology for Adaptable Reliable Systems (STARS) program.

While AT&T BaseWorX architecture for large scale reuse described by AT&T [13] does not reference the NIST APP framework only first published in 1991, the authors recommend reuse of software architectures and components based on many of the same standards recommended by the NIST APP. These include the Open Systems Interconnection (OSI) standards for networks and structured query language (SQL) for database management systems. To provide a framework for large scale horizontal reuse, the BaseWorkX provides an integrated application platform that includes "horizontal" or system level components so that developers can concentrate on the customer application. Both the client/server configuration for services and the manager/agent configuration for objects are supported by the framework. Since 1987, this platform has been used successfully on over 50 development efforts.

Batory and O'Malley also have addressed large-scale reuse in terms of database and network subsystems in the Genesis and Avoca projects [11]. They recommend the use of standard interfaces with their domain-independent model of hierarchical software system design and construction. The open architecture and interchangeable component approach is similar to the approach recommended by the NIST APP.

The Air Force Electronic Systems Center PRISM project ties the concepts of rapid prototyping, reuse, COTS and open systems architecture. The integration and COTS software packages for PRISM are cataloged in CARDS. While the library is considered domain-specific for command and control, COTS, open systems, and integration services apply to domains beyond command and control. Proof of concept prototype concepts are supported by the CARDS environment as well. The open systems architecture was selected to provide "plug and play" capabilities for ease of upgrade. In addition, the OSE services key to PRISM include DBMS, message processing and geographic information systems services. All of these are part of the NIST APP OSE framework.

The STARS goal is to "increase software productivity, reliability, and quality by integrating support for modern software development processes and reuse concepts within software engineering environment (SEE) technology" [14]. Demonstration projects underway as part of STARS include the Domain Specific Software Architecture (DSSA) technology projects [15] and the STARS Demonstration Projects [16]. The Air Force Demonstration project for Space Command and Control Architectural Infrastructure (SCAI) is based on many of the standards in the DoD TAFIM. Technology in the STARS program is also organized by service areas similar to those defined by the NIST APP. Also, the associated Asset Source for Software Engienering Technology (ASSET) reuse library, as well as other repositories sponsored by DISA and the DoD services, include Ada language bindings to some of these services in support of the DoD directives for use of the Ada language for application development. While reuse libraries do provide support for some of the OSE service interfaces, this position paper calls for a more strategic planned alignment. Such a plan also could be applied to initiatives such as the National Information Infrastructure (NII) [17].

References

[1] M. Gaska, "An Open Systems Profile for Concurrent Engineering," in *Proceedings of the 5th International Conference on Software Engineering and Knowledge Engineering*, (San Francisco, CA), pp. 271–4, 1993.

[2] D. Gause and G. Weinberg, *Exploring Requirements: Quality Before Design*. New York: Dorset House, 1989.

[3] N. S. P. 500-210, "Application Portability Profile (APP), The U.S. Government's Open System Environment Profile OSE/1 Version 2.0," tech. rep., June, 1993.

[4] I. W. G. P1003.0, "Draft Guide to the POSIX Open Systems Environment ," tech. rep., 1992.

[5] D. I. S. A. C. for Information Management, "Department of Defense Technial Architecture for Information Management, Volume 1, Implementation Concept, Version 1.1," tech. rep., October 21, 1992.

[6] D. I. S. A. C. for Information Management, "Department of Defense Technial Architecture for Information Management, Volume 2, Architecture Guidance and Design Concepts, Version 1.1," tech. rep., October 21, 1992.

[7] D. I. S. A. C. for Information Management, "Department of Defense Technial Architecture for Information Management, Volume 3, Reference Model and Standards Profile, Version 1.3," tech. rep., December 31, 1992.

[8] D. S. R. Program, "Software Reuse Metrics Workshop Proceedings," tech. rep., 23-24 February 1993.

[9] D. Belisle, "OMG Standards for Object-Oriented Programming," *AIXpert*, pp. 38–41, August, 1993.

[10] U.S. General Accounting Office Report to the Chairman, Subcommittee on Defense, Committee on Appropriations, House of Representatives, "Software Reuse, Major Issues Need to Be Resoulved Before Benefits Can Be Achieved," tech. rep., GAO/IMTEC-93-16, January, 1993.

[11] D. Batory and S. O'Malley, "The Design and Implementation of Hierarchical Software Systems with Reusable Components," *ACM Transactions on Software Engineering and Methodology*, pp. 355–398, October, 1992.

[12] D. Hughes, "Generic Command Center Speeds Systems Design," *Aviation Week and Space Technology*, pp. 52–53, March 8, 1993.

[13] R.P. Beck, S.R. Desai, D.R. Ryan, R.W. Tower, D.Q. Vroom, and L.M. Wood, "Architectures for Large-Scale Reuse," *AT&T Technical Journal*, pp. 34–45, Nov./Dec. 1992.

[14] D. A. R. P. A. (DARPA), "STARS – Accelerating the shift to megaprogramming," tech. rep.

[15] E. Mettala, "Domain Specific Software Architectures," in *Proceedings of STARS'92*, 8 December 1992.

[16] D. Bristow, "Air Force STARS Demonstration Project: Space Command & Control Architectural Infrastructure (SCAI)," in *Proceedings of STARS'92*, 9 December 1992.

[17] I. I. T. Force, "The National Information Infrastructure Agenda for Action," tech. rep., September 15, 1993.

4 Biography

Marilyn T. Gaska joined IBM Owego, now part of the IBM Federal Systems Company, in 1987. She currently is a Staff Programmer in SBIS Engineering and Technology, but has also worked in other fields to include healthcare prior to joining IBM. Currently she is a candidate for a Ph.D. in Advanced Technology, Systems Science, at Binghamton University, where she is working with Professor Donald C. Gause in the area of design. These part-time studies follow her second Master of Science is in Advanced Technology, Computer Science, from State University of New York at Binghamton in 1985. However, she received her first Master of Science on a Cornell Fellowship in 1979 following completion of a a Bachelor of Science degree at Cornell in 1978. She is also a member of the IEEE.