Special Report
CMU/SEI-95-SR-007

Proceedings of the SEI/MCC Symposium on the Use of COTS in Systems Integration

Edited by

Alan W. Brown

David J. Carney

Maureen D. McFalls

June 1995

Special Report
CMU/SEI-95-SR-007
June 1995

Proceedings of the SEI/MCC Symposium on the Use of COTS in Systems Integration

Edited by
Alan W. Brown
David J. Carney
Maureen D. McFalls

Software Engineering Institute
Carnegie Mellon University
Pittsburgh, Pennsylvania 15213

This report was prepared for the

SEI Joint Program Office
HQ ESC/AXS
5 Eglin Street
Hanscom AFB, MA 01731-2116

The ideas and findings in this report should not be construed as an official DoD position. It is published in the interest of scientific and technical information exchange.

FOR THE COMMANDER

(signature on file)

Thomas R. Miller, Lt Col, USAF
SEI Joint Program Office

This work is sponsored by the U.S. Department of Defense.

This work was created in the performance of Federal Government Contract Number F19628-95-C-0003 with Carnegie Mellon University for the operation of the Software Engineering Institute, a federally funded research and development center. The Government of the United States has a royalty-free government-purpose license to use, duplicate, or disclose the work, in whole or in part and in any manner, and to have or permit others to do so, for government purposes pursuant to the copyright license under the clause at 52.227-7013.

This document is available through Research Access, Inc., 800 Vinial Street, Suite C201, Pittsburgh, PA 15212. Phone: 1-800-685-6510. FAX: (412) 321-2994. RAI also maintains a World Wide Web home page. The URL is http://www.rai.com

Copies of this document are available through the National Technical Information Service (NTIS). For information on ordering, please contact NTIS directly: National Technical Information Service, U.S. Department of Commerce, Springfield, VA 22161. Phone: (703) 487-4600.

This document is also available through the Defense Technical Information Center (DTIC). DTIC provides access to and transfer of scientific and technical information for DoD personnel, DoD contractors and potential contractors, and other U.S. Government agency personnel and their contractors. To obtain a copy, please contact DTIC directly: Defense Technical Information Center / Attn: BRR / 8725 John J. Kingman Road / Suite 0944 / Ft. Belvoir, VA 22060-6218. Phone: (703) 767-8274 or toll-free in the U.S. — 1-800 225-3842).

Table of Contents

Abstract: The SEI/MCC Symposium on the Use of COTS (commercial off-the-shelf) in Systems Integration took place at the Software Engineering Institute (SEI) in Pittsburgh, Pennsylvania, on January 10-11, 1995. The symposium focused on two key points: the Department of Defense need for integrated systems, and its increasing reliance on acquiring software through commercial sources. The interrelationships between these points formed the conceptual basis for this symposium. These proceedings provide a record of the presentations and some of the main highlights from the discussions. The main body of this report is a set of notes from each of the panels.

Preface

The SEI/MCC Symposium on the Use of COTS (commercial off-the-shelf) in Systems Integration took place at the Software Engineering Institute (SEI) in Pittsburgh, Pennsylvania, on January 10-11, 1995. The symposium focused on two key points: the Department of Defense need for integrated systems, and its increasing reliance on acquiring software through commercial sources. The interrelationships between these points formed the conceptual basis for this symposium. Speakers were invited from the Department of Defense, industry, and academia to present their views. The symposium was organized as a series of panel sessions, each with several short presentations, followed by open discussion between the panel members and the audience.

These proceedings provide a record of the presentations that were made at the symposium, and some of the main highlights from the discussions. The main body of this report is a set of notes from each of the panels. For a hard copy of the slide presentation materials used by each presenter, contact Lisa Massucci (lm@sei.cmu.edu or 412-268-6755)

Many people from the SEI contributed to the success of this conference on both a technical and administrative level. In particular, we would like to recognize the contributions of Maureen McFalls, who carried much of the burden of organizing the conference and contacting speakers; Mike Mattison, who assisted her in providing contacts; Peter Feiler, Alan Brown, and David Carney, who were responsible for the technical agenda; Donna Baird and Lorraine Nemeth for providing invaluable administrative assistance and support; and the scribes who recorded the discussions that took place during the symposium.

We also thank Meg Wilson and Rob Smith of MCC for their administrative help and support in organizing the symposium.

Foreword

The purpose of this symposium was to understand the lessons learned from those who are involved in the use of commercial off-the-shelf (COTS) products, to frame a research agenda and to investigate the potential for broad collaboration in addressing principal issues. Executive acquisition officers from the government and representatives from industry and academia attended to present their points of view in open panel discussions.

Both the Software Engineering Institute (SEI) and Microelectronics and Computer Technology Corporation (MCC) are entering their second decades. During their start-up phases, each organization focused on different aspects of software. Since then, the technology and business environment has changed significantly. There are strong indicators that systems integration of commercially available software will be important to our respective customers.

This symposium provided an opportunity to share the community's insights on the use of COTS in systems integration in a forum structured to stimulate ideas, challenge assumptions, and encourage meaningful dialogue.

Larry Druffel	John W. McRary
Director, SEI	Chief Executive Officer and President, MCC

1 Overview of the Symposium

The high costs and low productivity of many software development and maintenance projects are well documented. At the same time, budgets for software development with both government and industry are coming under increasing scrutiny. A number of organizations, including U.S. government organizations, realize that one way to reduce the burden of development and maintenance is by constructing systems from collections of existing components. Furthermore, pre-existing components can be purchased as commercial off-the-shelf (COTS) products from third-party suppliers.

Considering software development as a process based on assembling collections of existing components is important because it raises the possibility of several gains and improvements. These include

- reducing software development costs as a consequence of constructing large parts of a system from existing components that have been previously developed and tested
- reducing the expense of maintaining software by delegating maintenance of COTS components to their suppliers
- improving reuse across programs through reducing individual purpose-built, “stove-pipe” software development in favor of larger purchases of existing products to be used consistently across the whole organization
- promoting a competitive component marketplace in many technical domains, with the result that system integrators will be able to choose from a range of components provided by third-party suppliers

This view of systems integration using COTS components is an attractive one. However, there are many potential roadblocks to its practical realization. These barriers vary considerably in nature, exhibiting technical, organizational, social, and economic dimensions. Typical issues being faced include

- acquisition guidelines and regulations that preclude or hinder the use of COTS components when developing a new system
- lack of visibility and access to COTS component internals to establish key component qualities such as security, reliability, and performance
- system instability during maintenance due to the frequency, variability, and inconsistency of periodic new releases of COTS components
- loss of schedule control during development and maintenance when relying on third-party products

- legal implications to COTS component vendors if system failures occur, leading to financial or human losses

Before systems integration based on COTS components can have more predictable, widespread effect on the software engineering community, these and many other such barriers must be addressed, and solutions developed.

The goal of this symposium was to bring together participants representing a wide variety of perspectives to highlight the importance of this new approach to system integration. In doing so, this community was able to share experiences that had been gained thus far, to identify major topic areas that require further attention, and to initiate a program of work that could begin to address these areas.

More than 130 people registered for the symposium, representing a wide range of industry, government, and academia. These attendees included a number of industry program managers, government policy makers, and research institute directors.

The symposium was organized around several major themes, with invited panelists providing a range of perspectives on each theme. Lively debate and discussion on those themes then ensued involving panelists and the audience. The themes of the symposium were

- Department of Defense Needs in COTS and Systems Integration

  Issues discussed included a view from each of the armed services as to their approach to the greater use of COTS components in systems integration.

- A Technical Perspective on Systems Integration

  Issues discussed included an identification of major technical barriers to the use of collections of COTS components.

- A Commercial/Business Perspective

  Issues discussed included ways to judge and approach the issues of cost effectiveness using COTS components, and how the COTS component market place is viewed from a commercial perspective.

- Systems Architecture and COTS Integration

  Issues discussed included positive and negative experiences with the use of COTS components in building large systems, how the relationship between government and industry is affected by a movement toward greater use of COTS components in systems, and the effect of using COTS components on such system qualities as reliability, performance or maintainability.

- Towards Open Systems

  Issues discussed included how open systems, standards, and COTS components all relate to each other in integrating large systems, how to deal with de facto versus de jure standards given current government approaches to systems acquisition and development, and how to decide when the use of COTS components is inappropriate.

- Acquisition Regulations and a COTS Approach

  Issues discussed included how the use of COTS components is precluded or discouraged today, how current acquisition regulations can be modified to make better use of COTS components, and the legal or licensing issues that must be changed or re-interpreted in acquisition regulations to ensure greater use of COTS components.

These proceedings provide a record of the major presentations and items of discussion. As such they provide an interesting and insightful overview of the COTS and systems integration problem from a range of important perspectives. A number of success stories, problems, and potential future solutions are discussed. We believe that these will provide the reader with an excellent overview and a realistic snapshot of the current state of the practice with respect to systems integration of COTS components.

2 Symposium Agenda

Tuesday, 10th January 1995

9:30 **Welcoming Remarks**
Larry Druffel, Director of the SEI

9:30 **Introduction and Presentation of Honors to Congressman John P. Murtha**
Robert Mehrabian, President, Carnegie Mellon University

9:45 **Keynote Address**
Congressman John P. Murtha

10:00 **SEI/MCC Collaboration**
John W. McRary,
President and Chief Executive Officer of MCC

10:15 Break

10:45 **Summary of Defense Science Board Task Force on Commercial Software**
Larry Druffel

11:00 **Department of Defense Needs in COTS and Systems Integration**

Panel Discussion
Moderator: Thomas C. Brandt, SEI
Panel:
- Blaise J. Durante, Office of the Assistant Secretary of the Air Force for Acquisition
- Rear Admiral John Hekman, Naval Information Systems Management Center
- Robert J. Kent, ESC Software Center

12:30 Lunch

1:30 **A Technical Perspective on Systems Integration**

Panel Discussion
Moderator: Julia Allen, SEI
Panel:
- Barry Boehm, Center for Software Engineering, USC
- William L. Scherlis, School of Computer Science, CMU
- William A. Wulf, Department of Computer Science, UVa

2:45 Break

3:00 **A Commercial/Business Perspective**

Panel Discussion
Moderator: John W. McRary, MCC
Panel:
- Kent D. Bimson, SAIC
- Claude M. Del Fosse, Software Productivity Consortium
- Doris Tamanaha, GM Hughes Electronics

Wednesday, 11th January 1995

9:00 **Systems Architecture and COTS Integration**

Panel Discussion — Moderator: Neal Walters, Loral Federal Systems

Panel:
- Harry Crisp, Naval Surface Warfare Center
- David Garlan, School of Computer Science, CMU
- John Machado, Technology and Engineering Group, SPAWAR

10:30 Break

11:00 **Towards Open Systems**

Panel Discussion — Moderator: Patricia Oberndorf, SEI

Panel:
- Michael A. Duniho, National Security Agency
- Joseph Hanratty, Office of the Secretary of Defense
- Jon Stonecash, Computing Devices International

12:30 Lunch

1:30 **Acquisition Regulations and a COTS Approach**

Panel Discussion — Moderator: Scott L. Reed, SEI

Panel:
- David Nordean, Air ASW Assault and Mission
- Peter A. Kind, LTG (Ret)
- Richard Knaggs, The Boeing Company

3:00 Break

3:30 **Where Do We Go From Here?**

Panel Discussion — Discussion Leaders:
- Peter H. Feiler, SEI
- Robert J. Smith II, MCC

3 Department of Defense Needs in COTS and Systems Integration

Focus: Overall issues for the DoD that relate to COTS, system acquisition, and system integration

Moderator: Thomas C. Brandt, SEI

Scribe: Gibbie Hart, SEI

Brandt introduced the session by setting the focus on two major topics. The first was the overall question of the DoD's needs in COTS and systems integration. The second topic concerned acquisition reform within the DoD: does this have any reality or not? In introducing the session, Brand indicated one change of speaker: Dennis Turner, CECOM, Director of the Software Center for the Army was unable to depart Newark airport due to a construction mishap, and Bob Kent, ESC Software Center for the Air Force graciously agreed to fill in.

Rear Admiral John G. Hekman, Commander of Naval Information Systems Management Center

Hekman described the Navy's commitment to acquisition reform and streamlining, and the movement away from government-only specifications. He noted such measures as the recently reconstituted the Software Executive Officials Consulate, where individuals from the laboratories as well as others at flag and SES levels are brought together to consider the problems associated with managing software and changes. He also noted that the Acquisition Reform Council and the Acquisition Oversight Council focus on software issues, an indication of the emphasis that the Navy is trying to bring to the subject of software management and COTS.

Hekman's presentation then touched on four key areas:

1. background: exploring software costs
2. the Defense Science Board (DSB) Study and ongoing efforts
3. Department of the Navy COTS paradigm shift
4. the COTS challenge

Hekman acknowledged that most people in the audience were probably familiar with some of the data and charts he used in exploring software costs, and that some were actually created by people in the audience (e.g., Barry Boehm).

To provide some background about software costs, Hekman described a significant acquisition, one that had a strong COTS component. The use of COTS had several implications especially for testing: "We wondered," Hekman said, "what would happen if we brought in a group of individuals whose sole job was to break [the software]" The result was a surprise to the Navy and to industry as well. "The good thing is that the end result was a better solution

and much more highly reliable software. The bad news is that the software was bad ... in many of those domains where it is considered that a failure is not fatal." Hekman noted that the testing community is now building good testing criteria for software intensive systems. "Program managers will know up front what they need to test, and what's going to be looked at when they go through the testing scenarios; we don't have that today."

Hekman spoke about the DSB study as it related to COTS, architecture, and reuse, the areas that the Navy volunteered to lead. He reviewed several key recommendations from the study, and the recommended assigned responsibilities for each. The DSB recommendations he focussed on were

- B.1 Define software architecture
- C.5 Develop expertise in analysis of domain software design
- D.1 Require trade studies and analysis of the use of COTS
- D.2 Establish "customer friendly" "component stores"
- D.3 Increase tech-base funding for security audit tools
- D.4 Capitalize innovative cost-effective techniques for acquiring COTS, such as enterprise licenses
- G.3 Initiate demonstration programs (e.g. ATDs)

On the issue of the Navy's paradigm shift toward COTS, Hekman noted that in May 1993, the Navy established a policy that selection of a government solution and not a COTS solution constituted disapproving the COTS solution. This started the change within the Department of the Navy, which is moving towards open systems and is instituting a systems engineering process. It is now taking a proactive stance in the standards boards, and wants those standards to satisfy Navy needs.

Hekman saw the COTS challenge is as follows:

- Defining "off-the-shelf" software (COTS?, MOTS?, NDI?, GOTS?)
- COTS applicability for real-time systems
- Business relationships and market leverage
- Life-cycle support/integration/costs

Hekman noted that a balanced approach to COTS solutions is possible, but pointed out that a solution for real-time systems does not yet exist. Integration is also a concern, but is achievable, as are the improved life-cycles costs associated with using COTS solutions.

Robert Kent, ESC Software Center, Hanscom Air Force Base

Kent began as follows: "We have a vision at ESC in terms of COTS, reuse, and architecture; and that is to not build any more software. It is to have in place continuously evolving architectures for our product lines in which we continuously qualify COTS products so that at any time, we have an instantiation of that particular systems so that we can instantly serve our custom-

er." This approach gets rid of instant source selection, instant specifications, and the necessity of integrating COTS products on an instant program. Kent pointed out that COTS integration is not an easy job. ESC sees COTS integration, architecture, and maintenance as a continuum that radically changes acquisition processes.

The following definition for COTS is from the Federal Acquisition Regulations (FAR):

> "Commercial product" means a product, such as an item, material, component, subsystem, or system, sold or traded to the general public in the course of normal business operations at prices based on established catalog or market prices (see 15,804-3(c) for explanation of terms).

There are some problems with this definition. For example: Has the product been sold on the open market for some period of time competitively? For how long? When was it first considered as COTS? The day the glossy brochure came out? The day it was shown at a trade show? Note also that the FAR definition says "commercial" and not COTS/MOTS/GOTS, etc. ESC rules say that they will only use C/GOTS if there is an identified maintainer with an 800 number to call and get help.

Kent then discussed how testing is a problem with COTS. The ESC facility is trying to qualify COTS or capture the attributes of performance of COTS products within our domain-specific architecture; this is difficult and requires taking some risks. Viruses, and maintenance are also large problems. Since a system might include forty different COTS products, what does one do about the issue of maintenance? Give the customer forty '800' numbers to call? Hire an integrator or create a new class of products? ESC is struggling with all these issues.

Kent then addressed the issue of "product lines." What are product lines? A grouping of systems that

- would be predominately COTS based
- could software be reused between systems
- could embrace similar information technology standards
- would benefit from a product line facility
- could be provided by pre-selected product line "OMNIBUS" contractors
- would benefit from re-use library concepts
- could use similar/common architectures

A key goal for ESC is to identify product line boundaries; General Franklin's position is that ESC programs are general variations of a theme, such as command and control centers, communications systems, intelligence centers, etc. These product line systems can be identified and can be represented by a generic architecture, or domain, to facilitate software reuse and rapid prototyping. A team, which included the SEI and ESC, looked at a large number (165-200) of programs at ESC and made a first cut at product line attributes. Kent suggested that

one could take this set of attributes and by substituting “air traffic control” to “battle field management” change the set of product line attributes from an Air Force set to an Army set. They are not that different.

ESC would maintain product lines by evolving architectures and continually qualifying COTS products so that at any point in time, they would have contracts in place that would give a customer, a user, an instant solution. It may not give a 100% solution, but it would give the customer a product in six months to a year rather than the several years it takes today. So if a customer can stand for a less than 100% solution, this technique would be very beneficial and saves money and time. Some software would still have to be developed; but overall, this represents a major savings in time and costs. According to Kent, “the biggest problem is cultural, since this is really a radical change in how we do business and address architectures.”

Blaise J. Durante, Deputy Assistant Secretary, (Management Policy & Program Integration) SAF/AQX

Durante has been a member of the Senior Executive Committee and was recently promoted to Deputy Assistant Secretary for Management Policy and Program Integration in the Office of Acquisition Executive for the Air Force.

Durante’s talk centered around four points:

1. commercial practices versus best practices
2. promoting best practices through reforming our use of military specifications and standards
3. promoting best practices through software reuse
4. PRISM: a success story using COTS

On the first point, Durante indicated that current interest in the Pentagon related to commercial practices. But he noted, “We have to look at the situation and apply the best practice. Not all commercial products are the best. Lots of times commercial is the best, but we have to be careful. It is not always a panacea. Good practices reside in both the government and commercial sectors.”

Durante also stressed that

- We have to get away from assuming that “commercial” means better.
- The challenge to the DoD is to determine and say what is “best.”
- Risk needs to be looked at with regard to functionality, schedule, life-cycle cost and user’s needs.

With regard to specifications and standards: “The new policy emphasizes ‘performance’ versus ‘how-to specifications,’” said Durante. The secondary emphasis is on non-government standards, which avoids re-inventing the wheel and recognizes those practices that drive the market. “If there is something out there,” he said, “use it. You don’t need a waiver to use it now.

You need a waiver to use the government specification now (which is a reversal). This is a real cultural change. The DoD no longer drives the market; someone else is driving the market and setting the standards." However, Durante cautioned that in some cases, the government has set the standards that industry is using, and these should not be eliminated.

Durante then addressed the acquisition strategy for reuse and how to implement reuse under COTS. Acquisition Strategy Panels should explore the full or partial reuse of existing software, including COTS. "The way to get people's attention is to put reuse into RFPs and source selection," he said. He also suggested making it an evaluation criterion on those areas because RFPs can require architecture that facilitates reuse. "We can also reward contractors for schedule and cost reductions attributed to software reuse," said Durante. "If software reuse is not feasible, consider requiring new software that facilitates future reuse as well as reuse capabilities."

Durante then cited PRISM as an example. "PRISM stands for portable, reusable, integrated software modules... It is 80% of a core program for the C2 centers," he said, "and the only thing we have to do is get the additional 20% developed. We are having trouble getting people to buy it because of the NIH; and it's a 95% solution instead of a 99% solution, even though the last 5% is usually wasted time and money." Durante is interested in this because it is a major cost-savings and the budgets are flat.

Discussion

Q: Do you have any case studies of any programs that have used your services?

A: (Kent) We tried the Omnibus notion on our procurement community a while back and we got beat up. Mainly because we could not prove the notion that 80% of the command center was in fact generic. Because we couldn't prove that then, what they said we were doing was pre-selecting, and therefore violating the competition and contracting act. We have run PRISM since that period of time. We have put into the field a couple of programs that we have assisted. One is the combat weather system and an emergency com system for STICOM. All of those were done in months, rather than years, and at one-tenth the cost. We recently did an exercise for ACC to put in the next generation U.S. air defense system, which was built in the 1960s and needs replaced. It was estimated to be a seven-year acquisition for $300 million. Kent has submitted a proposal for $38 million in 28 months. and the user loves the solutions.The solution is a variation which assumes COTS and COTS workstations. We have one product line which is the command center.

Q: Do you have evaluation criteria for COTS software?

A: Database and message systems: Not many have been on the market for the last 18 months; also we consider popularity. Hekman indicated that their market survey concept is to look at the requirements, the products, the marketplace, and the solutions, and see

what is being offered to perhaps redesign a requirement prior to milestone 0 so we aren't developing a 100% solution. Perhaps there is another solution that saves time and dollars and gets the highest level of performance at the lowest cost.

Q: How will you deal with trends towards consumer and commodities market which results in a new release coming out every six months?

A: (Durante) Software needs to be rehosted on the same platforms because platforms are not going to be changing more than every five to six years.

(Kent) We expect to have a first line filter, as the vendor will have to bring the software to them to be requalified, and it will be on a pay-as-you-go proposition. The vendor will have to take his widget and pay for the manpower to get it requalified on the domain specific architecture and then onto the buy list. Since the vendor is paying, he thinks this will minimize how often people are going to bring new things in to get qualified.

Other questions and discussion centered around 800 numbers and help services, open systems, vendors leapfrogging their competitors' functionality, switching products, and licensing and training issues. Most agreed that there are questions and issues that remain to be solved, but that the cost-savings and need to move towards acquisition reform, COTS, and systems integration is key to success.

4 A Technical Perspective

Focus: Technical issues in the COTS approach to systems integration

Moderator: Julia Allen, SEI

Scribe: Reed Little, SEI

Barry Boehm, Center for Software Engineering, USC

Boehm began by noting that the use of COTS software is changing the development process. According to the old process, system requirements drove capabilities. In the new process, capabilities will drive system requirements. However, Boehm cautioned that in the new paradigm, it is not a requirement if you cannot afford it.

Boehm spent some time examining the implications of the use of COTS components on lifecycle models. For example, use of COTS components have caused a revision to the spiral model. Additions include

- add COTS possibilities to alternatives
- analyze whether COTS meets requirements

Boehm then stressed the need to analyze the cost and risks of using COTS, illustrating this point with a number of examples of COTS cost and risk modeling.

Finally, Boehm noted that it is extremely important that the COTS software matches the architecture of the overall system. He continued with a short exploration of this point using the work of David Garlan on “Architectural Mismatch” as an illustrative example.

William L. Scherlis, School of Computer Science, CMU

Scherlis believes that system developers want to find conventional architectures and available components in the marketplace. Developers rely on a component marketplace — even if this marketplace is internal to an organization.

In fact, Scherlis noted that the competitive marketplace drives the buy versus build decisions. There are several aspects to making these decisions:

- Specifying components. It is important to validate needs as early as possible, to employ prototyping approaches, and to be as flexible as possible.
- Locating components. Help in this task may come from the standards community, who are trying to organize and categorize components.
- Evaluating components. The usual evaluation approaches apply — form, fit, and function.

- Adapting components. Important decisions need to be made concerning how to modify, and who performs and maintains those modifications.

There are costs and risks involved when making the above decisions. The costs include what it takes to make the decisions in addition to the cost of the actual COTS software incorporation.

Commercial software development has frameworks to enable quick systems development. These frameworks allow the developers to make “big steps.” Included in these framework are various languages, such as

- commercial oriented languages — e.g., Visual Basic
- domain specific software languages — e.g., SQL, HTML
- glue languages — e.g., TCL, PERL

Use of these frameworks do imply compromise on system requirements. For instance, all Macintosh applications look like Macintosh applications, as they conform to a certain “look and feel”. Scherlis noted that evolution of a framework is problematic; making irreversible changes to an architecture is a major challenge.

Even with these frameworks, the industry still needs to better address a number of issues. Help is needed from technology. This leads to a number of specific technology goals that will assist in the key decision making areas identified earlier:

- specifications — flexible, rapid requirements formation, validate early, not necessary to fully document, do prototypes
- components — supply network, standard product lines, standards, a component can be one of several different types of things
- evaluation — how to get components with a warranty: interfaces, functionality, performance
- adaptation — tailor component to user need, architecture evolution, stay away from the modified-off-the-shelf (MOTS) sink hole

Scherlis then identified several possible approaches to COTS component modification:

1. user modification through component configuration or tailoring
2. modification through partially revealing component internals
3. supplier tailoring based on individual user needs
4. supplier evolution of components in response to general market trends and user needs

Finally, Scherlis identified a number of technical initiatives that will be important to this field in the future. These were

- continue language research and direct it toward new types of languages, e.g., telescript

- support for evolving technologies, e.g., executable specifications and prototyping
- support for heterogeneous systems in order to assemble systems of components in a rapidly changing marketplace

William A. Wulf, Department of Computer Science, UVa

Wulf presented an overview of a workshop held in the UK in 1994 entitled “Software 2000”. Attendees were primarily COTS developers. However, they did not talk about using their system the way we at this symposium are envisioning. In fact, Wulf suggested that they are not developing components to be interconnected.

Wulf identified a number of major themes from the Software 2000 workshop that may have relevance to COTS and systems integration.

- The Global Information Infrastructure (GII). In the future, distribution will not be the important issue due to the existence of GII. However, business models change when technology changes. The GII will make us rethink the business models currently in use.
- Software engineering. No foreseeable quantum changes in software engineering were expected in the next few years.
- Public quality perception. Wulf noted that the workshop was held before the Intel Pentium problem. However, the workshop attendees believed that the general public is not ready to pay for higher quality. Price with reasonable functionality is the key.
- Legal issues. Where is information? The transitory, transparent, and pervasive nature of information is causing major legal problems to the community. These will heighten in the coming years.
- Education. More and more programming will be by non-computing professionals. Similarly, most systems will be assembled with COTS components, producing small systems for small work groups. Major challenges remain in educating the people who will be building and assembling such systems.

Discussion

Q: Would there be any difference in results if the Software 2000 workshop were held in the U.S.?

A: (Wulf) No marked differences.

Q: Adapting a COTS product that is not frequently adapted implies much risk. What about expanding a product outside its “range of applicability”?

A: (Scherlis) There is a “zone of safety” of applicability for COTS applications, and the adaptor must know what that zone is.

Q: How does one determine the zone of high risk?

A: (Scherlis) There are a range of such risks including locating the appropriate components, does the component have a future.

(Boehm) One needs to mitigate the risk — confer with COTS vendors and determine where they are heading and gain more capability.

Q: COTS products are big — how does one integrate parts of products?

A: (Wulf) The size of the products is a direct consequence of low cost of the software. The users would rather have large number of independent components rather than one glob. Market share provides more money for vendor investment for bigger globs.

(Boehm) Open interfaces are de-motivating for a vendor to provide. Word processor vendors want to sell their bundled spelling checker rather than providing an open interface so the users could use another vendor’s checker.

Q: Do suppliers exist that think of themselves as COTS suppliers?

A: (Wulf) Database, small component, and object-oriented library providers do exist. Microsoft does not consider itself as a COTS supplier the way we are looking at COTS in this symposium.

Q: The Defense Science Board assessed the problems for all systems to be issues such as being hard to integrate, hard to get requirements correct, and hard to modify. These are the same problems we are now saying we have for COTS. What is the distinction of a COTS-based approach?

A: (Scherlis) Attention to requirements is always needed, but we can be flexible in some uses of COTS.

(Boehm) COTS are built for diverse users; DoD systems are for special use. The cost model utilized by the COTS vendors produces robust things but not perfect things.

Q: While most software used to be for the government, that is not the case now. The U.S. government is not driving how we do software development anymore. There is no sense in writing specifications and developing software the old way?

A: (Wulf) Correct.

Q: There are constraints of lower cost, decreased time to deliver, and lower risk. Will COTS do that?

A: (Boehm) Currently there is a culture clash happening. The traditional approach is to have the requirements fixed before building the system. The best COTS-based approach is to look at the available technology and tailor requirements based on what's available. The government procurement system does not yet work this new way.

Q: While COTS should reduce cost and provide faster fielding of systems, there is still a maintenance problem with systems developed using COTS. Is the life-cycle cheaper utilizing COTS?

A: (Scherlis) With respect to maintenance one has to trust in what the vendor will do when new COTS versions come out. This implies a re-engineering of the glue code, with all the same old problems.

Q: If an aircraft with software developed utilizing COTS falls from the sky, do the maintainers call an 800 number provided by the COTS vendor?

A: (Boehm) A major fraction of the type of DoD systems we are discussing here (embedded real-time systems) will not use COTS. There is a small niche of DoD-specific systems which will have COTS. The DoD must sort out where COTS is high risk and where COTS can be safely used.

(Scherlis) The term COTS is used in a broad way. Developers of the PRISM product can develop their own product line. We are not slaves to shrink-wrapped COTS software. The vendors can develop an approach based on component product lines.

Q: Communication system companies are a large number of vendors but probably don't think of themselves as COTS vendors, although some customers probably do. Based on that, who actually thinks of themselves as COTS vendors/suppliers? We tend to think only of end applications when thinking of COTS.

A: (Wulf) How many vendors have on their front burner interfaces for their products?

(Scherlis) There is customer demand for interfaces. The vendor emphasis is to add value and hook the customer.

(Wulf) Look at the Lotus lawsuit over the pen interface. Vendors want to differentiate themselves in the market place.

Q: There are viable markets for vendors building niche infrastructure type products that can be used by large numbers. It depends on the size of the market, look at health care and insurance domain products. Where are missile systems products? Was anything said about niche markets at the Software 2000 Workshop?

A: (Wulf) No.

(Boehm) There is a phenomenon driving companies. They have the opinion “If we are not #1 or #2, we’ll leave the market.” They focus their investments on being first or second in a given market.

Q: Is the military technology market still big enough to affect the COTS vendors?

A: (Scherlis) There are a lot of niches where the military technology can have an effect.

5 A Commercial/Business Perspective

Focus: Ways to judge and approach the issues of cost effectiveness, reliability, quality and cycle time.

Moderator: John W. McRary, President and Chief Executive Officer of Microelectronics and Computer Technology Corporation (MCC)

Scribe: Maribeth Carpenter, SEI

McRary introduced the panel topic as a counterpoint to the previous panel on DoD needs. This panel was to address needs from the business perspective. The panel members had coordinated the content of their presentations, with the exception of the last-minute addition of Robert Smith, Vice President of Information Technologies at MCC.

McRary suggested that one definition of COTS might be “cost of the solution.” Expectations must line up with the ability to pay. Businesses are interested in POTS, “profit of the solution”, or the pricing/capitalization of reusable components.

McRary identified 4 themes to run through the presentations:

1. COTS software is sold as an immature product.
2. There are major product configuration and integration issues. Will there be incentives/benefits to outweigh the difficulties?
3. How will industry processes change to use COTS?
4. How does a reuse paradigm affect pricing?

Kent D. Bimson, SAIC

Bimson was at Lockheed Research prior to joining SAIC. His talk was entitled “Issues in Integrating COTS Products,” by SAIC and the Orlando ASSET Group.

The talk reflected Bimson’s personal experiences and addressed the question “What does it mean to take COTS and integrate it into the customer solution?”.

There is a familiar cartoon where two guys are manacled feet, hands, and neck to the wall. Their clothes are in shreds. Their beards are to their knees. Their bodies are bony skeletons. One guy says to the other, “Now here’s my plan.” Unlike this optimistic fellow, with COTS we don’t know how to get there from here. It is difficult to move through rapid growth.

There is a cost associated with being first to market. First-to-market products spend more for early entry and often charge more money as a result. Examples are early expert system shells and mainframe project management systems. When bringing research to market, there is always the threat of low-cost market entry of later products. Others enter the market more cheaply by using your coattails.

COTS products often do not provide access to data. An example is project management systems that do not export or import data (e.g., cost data) in standard formats. This resulted in downselecting to project management products with data access for value added applications.

Contrary to the open systems paradigm, stand-alone COTS products (e.g., expert systems products) generally cannot be embedded in other applications. In this instance, we were led to port the application from one that couldn't be embedded to one that could. The crux of the matter is the cost of using COTS products. We either pay at the front end or pay at the back end. We are hemmed in by acquisition requirements. COTS integration is not the solution.

Another issue is COTS products that do not run across standard platforms, for example, hypertext products that run only on one operating system. This led us to port the application to another hypertext product that could run in multiple operation system environments.

While using COTS within other COTS can bring you to market early, the cost can be high. Price expectation is being set by the game makers — no $5,000 products. We had hopes of using a graphic user interface (GUI), another company's product, embedded in an expert system. We had to pay a copy royalty to the other company. We eventually used another GUI product that was not integrated into an expert system.

Bimson shared these experiences from the business perspective to show that integrating COTS products engendered problems as well as solutions.

Claude M. Del Fosse, Software Productivity Consortium

Software Productivity Consortium (SPC) has recently run a study on COTS. Claude Del Fosse shared the conclusions. His talk was entitled "Putting COTS Software to Work", as presented at a previous symposium on the use of COTS in system integration. Participants were large defense electronics and aerospace contractors such as Boeing, Lockheed, Martin Marietta, and Grumman.

The situation is that companies are being asked to use COTS in software. Sample statements to that effect come from Emmett Paige, Jr.: "We will use commercial off the shelf software as much as possible. We will depend on the marketplace for life cycle maintenance and support," as well as SPC member company executives.

Many systems have successfully used COTS. The Boeing 777 has 4 million lines of COTS distributed over 1000 processors. They reduced development and maintenance costs and improved product portability and enhancement.

There was an executive round table at SPC on October 19-20, 1994, with 50 participants from SPC member companies at the vice president level. They discussed key issues in COTS software:

- Definition of “off the shelf” software

  COTS — commercial off-the-shelf?

  MCOTS — modified commercial off-the-shelf?

  NDI — non-developmental item?

  GOTS — government off the shelf?

- COTS applicability for real-time systems. An exact functional fit is improbable. Tailoring is often necessary. Is there enough documentation? Undocumented features may result in anomalous behavior.

  Verification and validation is difficult due to COTS development process. It is difficult for the integrator.

  Is COTS suitable for customized, mission-critical systems? For a COTS vendor, a system crash is not presumed to be fatal. DoD development practices are not used. The vendor’s product is directed at thousands of work station customers, not a few big customers. The Boeing 777 used Microsoft products. To Bill Gates, Boeing was too small a customer to meet with.

- business relationships. It is important to establish good business and technical relationships early. Determine the focus of the vendor to evaluate loyalty to your business.

  Anticipate support challenges: e.g., support contract is vague; bug fixes vs. enhancements; time lag of bug fixes (next vendor product release cycle may not fit needs).

  Define licensing strategies (pass-through rights — who owns the product after modification?)

  Look for lower tier suppliers (vendor subcontractors). They are typically the weak link in support.

  Do contingency planning to reduce risk. Vendor dependencies should be minimized. Look for multiple sources of systems and open architecture products. Avoid proprietary interfaces.

- life-cycle costs. What is the life expectancy of your system? Of the COTS tool? Of the COTS vendor?

- documentation, acceptance, and maintenance. What is the quality of the available documentation and training? Who maintains the modified COTS after version updates are released? Is there adequate configuration management?

- integration and performance. Beware of COTS performance claims. Advertised capabilities may not be available until the next product release. Complete product evaluations are required. Vendor demos are not sufficient. A good systems engineering approach is required. Successful product integrators are generally SEI CMM level 3 organizations. The system integration and testing effort is often underestimated.

- Impact of lower-tier suppliers. Understand COTS dependencies on lower-tier COTS suppliers.

Finally, Del Fosse reminded the audience that they are the ultimate beta test for products designed for less sophisticated users.

Doris Y. Tamahana, Hughes Aircraft Company

Doris Tamahana has experience on both the commercial and DoD sides. She has particular interest in reuse.

A central message of Tamahana's presentation was that COTS products are making a difference now. In the C3I domain, significant systems are being fielded today that contain a high percentage of COTS software.

A particular 36-month contract success that she used to illustrate her point credits good business relationships and buy-in to changing requirements. Every 6 months the requirements were revised. The customer admitted that the requirements were not initially known fully and had to evolve. Prototyping of requirements on COTS products was used to validate the requirements. Phase 1 provided the initial operational delivery; phase 2 involved incremental enhancements. Configuration management turned out to be a problem with 2 different versions of the system.

COTS has been viewed by some as the current FOY (Fad of the Year). The software engineering community talks about COTS insertion, COTS "megaprogramming," COTS-based systems, etc., but without standard definitions of these terms. Tamahana suggested the following basic definition: COTS — hardware and software products and services developed to commercial practices for broad public and private sectors.

The "FOY" status of COTS indicates some real underlying problems. There are some overly simplistic interpretations, for example:

- "I can have my COTS and modify it too..."
- "When you use COTS it is always cheaper, so why are you charging me for...?"

There are many "GOTCHAs": Some examples include:

- "It's only a little change; ask the vendor to make it for us" (To the vendor, you become a market segment of one.)
- "I've seen all the features I want in the COTS out there; don't worry, just find us the right combination..." (feature compatibility, integration problems)
- "I know it's COTS, but I still want a blinking cursor." (modified COTS is not COTS)

There is a need for process definitions for COTS usage. COTS usage tends to be high in support of the program management and life cycle processes, and the systems and software engineering processes. For COTS processes themselves, we need process definitions for COTS insertion, COTS development, COTS tracking, and COTS evaluation.

Tamahana discussed four categories of COTS insertion:

1. stand-alone: used as is either as part of the product or as a development tool;
2. modified: modifying COTS is not COTS;
3. COTS linked by shared data: a current practice with problems;
4. COTS open systems: a future practice with problems.

Potential users should keep asking the question whether COTS software products are really a solution. From a business perspective, one might go to COTS products to achieve more concurrency with a new life cycle model.

COTS works best in an environment of flexible requirements management. If the system is over-specified, it will be hard to find a COTS "fit." There is a learning curve in converting to a COTS process, and major vendor updates can add unexpected incremental cost. You may not be able to influence the COTS vendor because vendors are market driven.

In summary, Tamahana suggested we need to take care to balance the COTS solution against a custom solution. COTS presents us with both risks and opportunities.

Robert Smith, Vice President of Information Technologies at MCC

Smith spoke from the perspective of a software entrepreneur.

Since DoD software is becoming a smaller part of the market, Smith advised that we need to understand the vendor's perspective on risks and investments. We need a win-win situation, mutual opportunities for mutual benefits. Currently vendors don't care about the DoD and government market because the DoD is a small part of the software market and the vendors want to play to large growing markets.

Discussion

A participant suggested that the session raised a number of problems and issues. Fred Brooks wrote the "No Silver Bullet" article — we should put COTS on the list of no silver bullets.

Another suggested we should encourage vendors who come to the DoD with custom solutions to create COTS instead.

Q: The government owns lots of software. What if it were commercialized? In violation of “non-competitive” position?

A: There have been programs to make DoD software licensable. NIST has cost sharing initiatives.

Q: To what extent do we accept “second best” architectures and requirements to use COTS? What can you afford?

A: It’s a pragmatic business decision.

6 Systems Architecture and COTS Integration

Focus: Examining the effects of COTS component use on systems integration strategy.

Moderator: Neal Walters, Loral Federal Systems

Scribe: Paul Clements, SEI

Walters' organization is working on process issues and integration strategies for COTS-based MIS systems.

The original dream of COTS was lower effort and improved performance for the customer and the integrator. Of course, reality has shown that in fact higher development efforts are required. The first wave of optimism over COTS has passed, and we are recognizing the need to apply science and engineering discipline to the problem, much like software in general 25 years ago.

Walters suggested that the problems include the fact that the COTS component never does quite what is desired; it is not integratable (that not being a goal of the vendor), and it isn't Ada-friendly. It's hard to work with vendors; developers have in effect lost control. Vendors don't know or care about government acquisition policy, documentation standards, etc.

Early lessons learned include the need to develop better process measures and metrics, and to improve process models based on COTS acquisition. Hands-on evaluations are also important, as is early prototyping. Customers must learn requirements flexibility; towards this end, requirements rationale must be captured. Given a so-called requirement that rules out a COTS component, is it really necessary or just the result of an engineer discovering something nifty at a trade show? Finally, reusing successful products is key.

Customers need to learn integration skills, including the fact that vendor management is different from subcontractor management. They must also learn COTS-based cost estimation skills. (Loral has developed a cost model based on function point counts, modified for COTS-integrating projects.) Change management (such as when a vendor's company disappears) is essential.

Finally, Walters believes that new life-cycle models for COTS integration projects are needed. (Loral has several, three of which were presented.)

In any project, the challenge of architecture is to understand and work with constraints, and using a COTS component presents a set of constraints. In a naive view of COTS-based architectures, we picked components and wrapped them with glue code. However, this architecture is extremely brittle, with respect to changes in products and especially with respect to changes in business rules brought about by technology advancement. More robust architectures are required.

Harry Crisp, NSWC

Crisp presented an overview of the AEGIS weapon system acquisition program being run by NSWC. His domain, typified by AEGIS, is that of complex systems, whose components cross technology domains, featuring increasing automation to support human-in-the-loop applications. The systems are usually real-time to varying degrees, have a tight integration between hardware and software, and often need to be able to demonstrate high dependability and assurance.

The architecture for AEGIS is driven by the heart of the ship, the SPY-1 phased array radar, and its associated weapon control system.

This is the development environment into which NSWC is seeking to integrate COTS components. Of interest to the workshop were the plans for Build Level 6 to include COTS processing elements, and for Build Level 7 to be a distributed network-based workstation-oriented architecture. Plans for a 21st-century surface combatant include an open architecture for the entire ship, not just its weapons systems.

According to Crisp, use of COTS components opens up the acquisition process, in the sense that it motivates otherwise-isolated cells in the Navy to talk to each other, in order to pursue common solutions to acquiring COTS-based systems.

David Garlan, School of Computer Science, CMU

Garlan spoke about the specific things that can go wrong when trying to assemble systems from off-the-shelf components. In building Aesop, an environment for building architecture-based environments, several problems were encountered by attempting to import components. The components include

- object-oriented database
- toolkit for graphical user interfaces
- RPC mechanism
- event-based tool integration kit

Developers expended approximately ten times the expected effort to build the system. The problems included

- large code size. The components were a few megabytes each, and there was no code sharing; whole libraries had to be imported, even if they weren't used.
- poor performance. Interaction between tools and the database was poor via the RPC facility.
- modifications to the components were required: e.g., the event loop.

- replacement of functionality was required.

Garlan believes that the problems were caused by different assumptions made by each component. From an architectural point of view, these assumptions were

- about components (who owns thread of control)
- about connectors (patterns of communication and data communicated)
- about topology (e.g., the database assumed communication with all other components, and no communication among the other components)

To solve these problems, Garlan suggested a number of avenues of future work. First, we need better documentation of architectural assumptions. Second, we need to consider connectors (glue) as first-class architectural citizens. Third, we need better architectural analysis tools, the analog of compiler type-checkers, to see if the assumptions made by components are in fact being followed. Fourth, we need to instill a culture of orthogonality, to encourage vendors to supply products built as separable parts. Finally, we need a better understanding of solution strategies, as might be obtained by taxonomizing types of architectural mismatch and their solutions such as wrappers.

John Machado, SPAWAR

Machado's area of expertise is in the acquisition and development of technology to support embedded computer systems, or mission-critical computing resources (MCCR). The DoD is moving to adapt open systems as the paradigm for MCCR procurement. In order to make sure that DoD needs are being served by industry, the DoD must take a more active role on industrial standards committees.

The MCCR world can be divided into five levels:

1. mission
2. operation (e.g., joint strike)
3. system (e.g., F-22)
4. functions (e.g., navigation)
5. infrastructure (e.g., operating system)

Corresponding to each of these is an architecture, which describes components and flow of information among the components. For example, the operation "strike" has an architecture composed of systems, which exchange information in order to carry out the operation. Program managers who make procurement decisions reside at the system level.

The program manager looks to the level above (here, to the operations) to see what information is provided to his system, and what information his system is expected to provide to other systems in the operation. He also looks one level below (here, to the functions) to see what functions are available with which he can construct his system.

The process repeats at each level.

Discussion

Q: If the COTS components in a system change, when does the architecture have to change? How do you know when an architecture must change?

A: (Machado) Architectures are like standards; they are kept and used as long as they are useful. Now we know that both things (standards and architectures) evolve, and newer standards efforts are taking evolution into account.

Q: The barrier to COTS integration is interfacing; the components are not open; there's no source code usually delivered; non-disclosure agreements get in the way. How can we motivate vendors to help with all of this?

A: (Garlan) It can be argued that it's in the vendors' interest to document the kinds of information that allow open integration to take place. It opens up new markets for their products. The problem is that there's no good way to document the interfaces, assumptions they make, etc. A non-solution is the adoption of a standard, such as CORBA. We need to find ways to say what's important, and bring the solution strategies (wrappers, etc.) to a disciplined engineering stage. (One criteria for components to go into Aesop was accessibility to source code, by the way.)

Q: Are vendors aware of these problems? Why don't they produce compartmentalized products? They all want to sell top-to-bottom complete solutions.

A: (Walters) They're naive with respect to that. They also have specific markets towards which they orient their products, and for those markets compartmentalization isn't required.

Q: A comment is that markets are immature with respect to deploying compartmentalized versions of COTS. We need to articulate the size of the potential market for components to motivate vendors.

A: General agreement.

Q: How successful was the use of open standards in the Aegis work?

A: (Crisp) It's too soon to tell. The goal is an open systems architecture; we have a few years to go. The advanced development model is a small part of Aegis, and it's about half-way prototyped. There has been good success so far, but no true real-time operation yet. We're optimistic.

Q: What is the current state of the art in availability, performance, etc., that we can specify at the infrastructure level?

A: (Machado) Standards embody what we know well. Embedded systems are state of the art, because that's how we gain advantage over the enemy. Therefore, standards do not tend to apply, and this is an ongoing, not temporary, situation. We must continue to act with standards committees to bring standards up to state-of-the-art levels.

Q: (follow-up) Yes, but where are we now? If we can't even integrate components, we don't have a hope of integrating components with state-of-the-art performance, availability, etc.

A: (Machado) The Navy's Next Generation Computing Resource program, begun in 1987, had an impact on standards in the way mentioned above. The Navy aligned itself with other major concerns, such as the telephone company. The phone company puts its switches on street corners, which makes them need to be rugged, salt-resistant, etc., similar to Navy systems.

A: (Garlan) In other domains, it's possible to mandate standards. The problem sometimes is the multitude of standards.

7 Towards Open Systems

Focus: Consider open systems and their realization within the context of the use of COTS components.

Moderator: Patricia Oberndorf, SEI

Scribe: David Carney, SEI

Oberndorf opened the session with some comments and observations about the topic of open systems. She noted at the outset that there is a major terminological problem when open systems are mentioned; an audience of some fifty people would probably come up with double that number of definitions of open systems.

She also noted that the previous sessions of the Symposium had made the implicit assumption that the promise held by widespread COTS use was a beneficial one. But there were as yet no rules, nor any basis for evolution. Presuming that the 'old rules' are out — we don't create a Mil-Spec and then write everything from scratch any more — then what are the new rules? One answer that has been sounded is to "use an open systems approach." However, that statement is currently not well-defined, and until greater agreement on what open systems actually means, the promised benefits of a COTS approach will not be achieved.

As a point of departure for the open systems session, each of the panelists had been asked to focus on some key issues. The first was the relationships that exist between open systems, COTS, and standards. A second was: What is the relationship between de jure and de facto standards? A third issue was: When is it appropriate not to use COTS? When is it more appropriate to build the system yourself?

Joseph Hanratty, Office of the Secretary of Defense

Hanratty briefly discussed the new DoD policy toward open systems, and then focussed on the key open systems issues posed by Oberndorf.

Regarding the DoD policy that was recently issued, the message is clear: an open systems approach will be used in acquisitions of weapons systems electronics. This affects new acquisitions as well as modifications of existing ones. It also applies not only to digital acquisitions, but also to all analog components (e.g., power supplies, connectors). The governing policy is to increase and institutionalize use of COTS. The key actions needed to accomplish this are to coordinate the activities of the services and the DoD; to educate various players through such mechanisms as the SEI's course on open systems; and to coordinate the DoD activities with the activities of the various standards bodies. All of these activities will be focused on refining the new DoD policy toward COTS and open systems.

Hanratty's main presentation dealt with the questions and issues that Oberndorf had made to the panelists. To the question of how open systems and COTS relate in integrating systems, he noted that integration occurs through components, through standards, and through architecture. Integration through components tends to result in "point-to-point" solutions; these are not generally useful, and lead in the worst case to chaotic systems. In integration through standards, open systems are an enabler for using COTS components; this, in turn, depends on the COTS components conforming to the standards in question.

There are four main integration strategies in which open systems and COTS are significant: "grafting," substitution, a hybrid strategy, and the use of architectural models. The first three of these are useful for legacy systems, and the last is for new procurements. For grafting, Hanratty compared this to the type of action where a limb is grafted onto a tree; an actual example of this is the Navy's AYK-14 computer procurements. Substitution is a more open systems approach, in which one component is removed and another replaces it; the presumption is that some standardized interface permits this substitution. In actual practice, both of these will rarely be used, leading to a pragmatic, hybrid approach. When a system can be designed from scratch, as with the Navy's new attack submarine program, then architectural models can be used to achieve an open system and to incorporate COTS components in it.

To the second of Oberndorf's questions, namely, the relationship between de facto and de jure standards, Hanratty responded that one key for choosing a standard is timing. The opposite poles of immaturity versus obsolescence will indicate for or against a particular standard.

To the third question, when is a COTS or open systems approach wrong, Hanratty responded first that a COTS approach might be wrong based on cost or performance; the risks stem from the loss of control, since the DoD moves from being a producer to being a consumer. The mitigation for this risk is for the COTS components to be standards-based. As to when an open systems approach might be wrong, Hanratty pointed out that it is rarely, if ever wrong. He did admit, however, that there are certain characteristics of a closed system (lack of modularity, short life expectancy, "one-of-a-kind" system) that could suggest that the open systems approach is not necessary.

Hanratty summarized his presentation with the following concerns and questions:

- When is it cost-effective to modify a system? How can a program manager know this? What criteria does he use to know this?
- How do we know if a particular COTS product really has market acceptance?
- How do we know if an interface description is adequate?
- How can we extend modeling and simulation tools to help with these problems?

Michael Doniho, National Security Agency (NSA)

Doniho discussed the question of open systems from the viewpoint of the National Security Agency (NSA), which has certain differences from the DoD perspective.

First, when the NSA considered the Perry memo, the realization was that it is merely common sense. The NSA has historically been interested in commercial components (e.g., M204). NSA was one of the first Unix sites outside AT&T, and continually does cost-benefit analyses for use of COTS on all acquisitions. The real issues for this conference are fourfold: cost reduction, interoperability, affordable technology, and software reuse (the last being a widely misunderstood notion).

To answer Oberndorf's set of questions, Doniho first defined the key terms as

- open systems — plug-compatible software modules; also accessible interface standards
- standards — agreements among software developers
- COTS — shrink-wrapped software

The relationships among these three are that open systems need standards, and that COTS software should be used in open systems. Said more simply: COTS require open systems, which in turn require standards.

To the second question, Doniho responded that the choice is for de facto standards every time; de jure standards that are not being used are not standards (using the definition of standards as "agreements between developers" unused standards can not reflect agreements).

Doniho separated the third question into "When is COTS wrong" and "When is open systems wrong." For the former, Doniho itemized a number of possible conditions:

- when COTS software is not available
- when it doesn't work in the required environment
- when ti lacks essential features that cannot be easily added
- when it costs too much or the pricing structure is impossible
- when free software is available that will do the job
- when government-owned software is cheaper
- when standards either don't exist or are unreliable

Doniho then illustrated these points by describing software currently in use at NSA. One mapping system ("Oilstock") was developed in-house, while another ("Beavermap") was purchased from an outside vendor. The cost and benefits indicated that for this particular problem, it was more economical for the software to be developed rather than to use COTS.

As to the latter question of when is an open systems approach wrong, Doniho pointed out that we need to consider various perspectives. For instance, for a tool vendor, if an open systems approach does not eventually provide greater market share, then there is little reason for such an approach to be adopted.

Doniho illustrated this issue with a series of examples from two research topics currently underway at NSA and MCC. One, called CARNOT, is part of an effort to integrate heterogenous COTS databases by means of intelligent agents. The second, CYC, is a global reasoning engine; although not yet a commercially viable, this project has the potential to be so, especially in the context of the exponential growth of information management needs.

In summary, Doniho believes that COTS in general is headed toward greater modularity; in his opinion, the state of COTS is similar to that of the electrical engineering community 20 years ago.

Jon Stonecash, Computing Devices International (CDI)

Stonecash noted at the outset that he is in many ways dissimilar to most other participants at the Symposium, since he is not connected with the DoD, but is working on commercial aviation issues. However, the key problem in that domain is interfacing with software that is very similar to DoD software (e.g., on-board systems), and thus his views probably have much in common with the rest of the panel. The major problem for CDI's work at the moment is in determining the status of on-board systems; this is essentially an integration problem.

One aspect of this problem lies in the question of certification. For instance, he would definitely not want certain popular software packages (he used Windows as the example) performing safety critical roles on commercial aircraft. But this is an example of software that is good enough for some jobs while not being good enough for flight-critical roles.

One solution is to have a hybrid mix, with some software used for aircraft avionics, and other COTS applications used for non-critical purposes; a "firewall" protects the flight-critical software. The problem then shifts to certifying the firewall, since not a single bit of code can be changed.

In applying this solution to commercial avionics systems, CDI took some special steps to "sanitize" the COTS components; for instance, when using "Windows," the Solitaire module was removed.

Stonecash reminded the audience of the famous pair of questions once asked by Bill Gates: What if computing were free? What if communications were free? Stonecash added a third one: What if *development of new applications* were free? Stonecash sees a trend toward this in many aspects of commercial software. If this trend is true, then there must be a change in the proportion of time needed for new development compared with the time needed for documentation.

Discussion

Q: Does NSA see value in POSIX as an operating system standard?

A: (Doniho) We've looked at these, but they don't answer all of our needs.

Q: With regard to CARNOT and CYC, since you (NSA) are largely driving their development, can you really describe these as COTS? If so, who are the other customers for them?

A: (Doniho) We believe that they will be widely used in the commercial world.

(Stonecash) We can corroborate that; CDI is very interested in their development. CDI is not currently a member of these projects at MCC, but is seriously looking at CARNOT. If it does what it claims to do, then it will be very valuable to us.

Q: Doesn't the policy of using freeware off the Internet make NSA vulnerable?

A: (Doniho) We only use source code freeware; we mitigate the danger by inspection.

Q: Isn't it the case that as we move forward to a commercially-based world, we should agree on our expectations?

A: (summary of all comments) We must somehow give the COTS vendors a language by which they can describe their products. The problem then becomes: How do you judge a product by its description?

Another issue (in terms of expectation) is that there is a problem with the technological horizon: both CARNOT and CYC are up to a decade away from maturity. There is an unavoidable friction between mission requirements and capabilities of COTS tools. We need to start now to push both the standards community and the COTS community toward COTS satisfaction of such requirements.

Joining the standards community is critical: if the DoD doesn't participate, it will never get what it wants.

8 Acquisition Regulations

Focus: Examine current acquisition practices and regulations with respect to the ways in which they encourage or discourage the use of COTS components in systems integration.

Moderator: Scott Reed, SEI

Scribe: Kurt Wallnau, SEI

Reed began by noting that the Federal Acquisition Streamlining Act recently passed, known as Public Law 103-355. Included among its provisions:

- The use of COTS is required.
- Requirements must be defined in a way that COTS can be procured.
- COTS providers can compete in any acquisition.

Reed noted that a host of difficult issues remain to be resolved, including issues of warranties, upgrades, and vendor relations.

Also noted was the interplay between COTS and the peculiar requirements imposed by DoD systems such as weapons systems — since the DoD is no longer the dominant market force in computing technology, there is the potential for a divergence between the technologies that will become available as COTS components, and the sometimes more stringent requirements imposed by DoD battlefield missions.

David Nordean, Air ASW Assault and Mission

Nordean began his presentation by distinguishing between *policy* and its *realization in practice*: the gulf between a pronouncement of policy and the point where institutional conservatism is overcome so that the policy is actually implemented, can be quite large. This point ran as something of a sub-text throughout the presentation. For example, Nordean claimed that budgetary restrictions dictate, as a matter of necessity, a non-developmental item (NDI) approach to the kinds of systems he develops. However, even where necessity and common sense dictate an NDI approach, he has had to struggle with and overcome a fair degree of institutional resistance within the acquisition system to take the indicated NDI approach.

Throughout the presentation, Nordean equated the notion NDI with COTS component — or, at least, the terminology was used interchangeably. The implication is that from the program manager's perspective, limited budgets mean that, in practice, systems can only be built if there is a reliance on the use of previously-developed components, commercial or otherwise.

Nordean illustrated what he considered to be a COTS approach in terms of the system he is currently managing — a survivable C2 system for the ballistic submarine fleet. He described the system in terms of its constituent components — mostly commercially available hardware (buses, hard drives, etc.).

He then asked the audience whether they believed this system, as described, was an NDI system. The audience, for the most part, expressed confusion about the answer: most answered "don't know." Nordean then indicated that he was not surprised by the confusion, and attributed this to the general lack of knowledge about what constitutes COTS and NDI components. According to him, this system would be classified as an NDI system.

Nordean then went on to describe some of the technical issues involved in developing a system based upon COTS components rather than DoD-specified components. One advantage he sited concerns system growth potential. There is an acquisition guideline that indicates that procured systems should accommodate 50% growth. He claims that the COTS-based system he described in the earlier portion of his presentation has sufficient excess capacity to allow over 200% growth. He attributed this to the fact that the DoD is no longer driving computing technology. He concluded from this that the DoD must learn to follow and exploit the rapidly evolving computing industry.

On the minus side, Nordean discussed some of the more difficult issues relating to the use of COTS. He noted that various DoD-specific requirements would not in all likelihood be met by COTS components, and as a concrete example illustrated this point with electro-magnetic pulse (EMP) hardening as a DoD requirement. In his own program this problem was bypassed by "hardening" the airframe rather than the components — and this, in turn, led to a technically superior solution when compared with stipulating EMP requirements at the component level. As an example of an unresolved technical issue related to DoD mission-specific requirements on COTS components he noted that most COTS components have a temperature operating range with a low-end of around 0°C. Unfortunately, some DoD missions will expose the components (within the airframe) to significantly colder temperatures.

Peter A. Kind, LTG (Ret)

Kind limited his presentation to making three points: the need for architecture, standard data, and a revised acquisition approach centered on specifying desired functionality. These points were addressed in turn.

Kind was not explicit about what he meant by architecture, but identified the importance of architecture in terms of its use to

- frame discussions and identify key concepts and terminology
- establish a foundation for configuration management

Kind indicated that, if anything, the use of COTS increased the importance of architecture. No specific rationale for that claim was provided — presumably it is due to the increased reliance on “plug-in/plug-out” style architectures that stem from COTS-oriented systems, thus resulting in a more dynamic system in terms of possible configurations.

The need for standard data was also cited. As an illustration, Kind described a real-life situation where the computer symbols used by two closely-related systems to describe a truck were in fact different. While problems such as these are not unique to COTS-based systems, the point was that the full benefit of COTS can not be achieved until some of the more fundamental issues pertaining to data standardization are solved.

Finally, Kind indicated that new approaches to acquiring systems were needed to provide contractors with sufficient flexibility to propose COTS solutions. Overly-specified requirements can, in practice, prevent the use of COTS components — for example, stipulations of component EMP characteristics in Nordean’s system would have made a COTS approach infeasible. Instead, General Page proposed a DoD acquisition approach that works by describing the desired functionality of the system, and then focusing DoD attention on developing a rigorous testing/acceptance approach.

Richard Knaggs, Boeing Aerospace Corporation

Knaggs itemized a number of problems and recommendations for using COTS in acquisition:

- The use of MIL-SPECs precludes the use of COTS. As an example, Knaggs cited a situation where delivery of a system was refused because the documentation provided by the COTS component vendor did not comply with DoD Mil-Std-2167 documentation requirements.
- Different development standards are needed for developmental and non-developmental software. This can be thought of as a generalization of the problem alluded to above regarding the use of MIL-SPECs. In addition to documentation, other factors relating to software development standards, e.g., testing and reliability issues, need to be different where COTS components are concerned.
- The “1-800” concept — where the customer has a single point of contact for problem reporting — is impractical in a COTS-based system where a large number of commercial software components may be used. The integration contractor will not have sufficient expertise in each product.
- Do not require the integration contractor to purchase all maintenance upgrades of COTS components. In practice, the system will never stabilize and the cost of re-integration will far outweigh the marginal benefits of the component upgrade.

- Allow contractors to submit alternative proposals. The idea is that alternative proposals would provide contractors with the flexibility to propose solutions that would violate some stipulations of the solicitation but which, on balance, might be technically superior (due to use of COTS) than the strictly-compliant proposal.

- The use of detailed functional specifications (for example, the Federal Acquisition Regulation (FAR) 10 purchase description "brand name or equivalent") must be avoided. The point is that detailed functional specifications for COTS components will either match only a single component, and this would be sole-sourcing in disguise, or if two components matched the same detailed specification, then this would indicate that some form of copyright infringement may have occurred.[1]

- Multi-level security is still an unresolved problem for COTS components.

As a closing point, Knaggs raised the issue of whether COTS-based systems cost more or less than non-COTS systems. While not expressing an opinion on the matter, and he did not distinguish between development maintenance costs, he indicated that the very question may reveal that those who ask the question may not be considering the *whole system*. Presumably, a whole-system perspective might include factors such as the degree to which a system is adaptable to changing technology — arguably a characteristic of COTS-based systems.

Discussion

Q: How is a "source line of code" count for COTS software products determined?

A: (Nordean) Vendor estimates can be used if necessary. But this figure is not significant. The number was provided during the presentation because it is part of the existing acquisition culture and for no other reason—perhaps another indication of the changes that must occur in current practice and perception in order for COTS approaches to be practicable.

Q: Taking a COTS approach, what are the implications of allowing contractors to routinely submit proposals that are, in effect, 85% solutions. This might arise, for instance, if alternative proposals are permitted.

A: (Nordean) The government is simply looking for "best value," and whether the proposed solutions were 85% or 100% was not the real point.

Q: (Follow on) Does this approach not provide a fertile basis for protests?

1. The underlying premise seems to be that no two COTS components are exactly alike—that it is their differences that establish their competitive characteristics, and these differences will always be manifested at the level of detail described by detailed functional specifications.

A: (all) The panelists agreed that some form of acquisition reform that is based on COTS is needed, although exactly what reforms are needed remains unclear. The notion of specifying functionality rather than specifying solutions is needed; also, one can consider specifying system requirements in terms of available COTS technology, rather than specifying system functionality without regard to COTS technology.

9 Where To Now?

Focus: Obtain input and suggestions from the audience concerning possible near term plans for making progress in this area, advising the SEI and MCC on the role that they may play in this area, and recommending possible topics for future workshops.

Discussion Leaders: Peter Feiler (SEI) and Rob Smith (MCC)

Scribe: Alan Brown, SEI

Peter Feiler, SEI

Feiler began by stating the role of SEI and MCC: both have the role of helping the community to improve their practices. This session is aimed at providing input to both organizations as to they can help in the domain of system integration of COTS components.

However, Feiler recognized that the work on systems integration of COTS components is not taking place in isolation. There are many related topics in which considerable work has, and is, taking places. Examples of these topics include the current SEI work on software architectures, domain analysis, and integration of Computer-Aided Software Engineering (CASE) tools.

Following the conclusion of this symposium, Feiler stated that the next steps will include

- publishing a set of proceedings that describe the presentation and discussions that took place (i.e., this report)
- development of an action plan that outlines a set of tasks to be carried out by the SEI and MCC and the relationship of those tasks to initiatives currently taking place elsewhere
- planning another workshop within the domain of systems integration of COTS components

Rob Smith, MCC

Having listened to the presentations and discussion over the last two days, Smith extracted some of the major themes of the symposium. His analysis provided the following list of topics requiring much greater attention in the future:

- providing experience reports of lessons learned by system integrators to help others see what worked and what didn't
- in depth analysis of a range of technical issues connected with describing appropriate system architectures, standards and their use, and so on
- investigation of how to measure system qualities (the "ilities") for systems composed of COTS components

- further investigation of business issues related to the use of commercial COTS components, including licensing concerns, establishing appropriate feedback relationships between COTS vendors and users, return on investment from using COTS components, and so on
- examination of cultural issues associated with the move to this new way to develop and maintain software
- describing approaches to manage the evolution of systems containing COTS components

Smith suggested that it would be useful to hear more themes and issues from the audience to expand this list.

Discussion

One suggested future action was to invite representatives from key COTS suppliers to try to learn more about their perspectives on the problems of integrating their products with those from other vendors. In particular, it would be interesting to hear their views of the marketplace in COTS components, and the marketing rationale for some of their technical decisions. This suggestion was expanded with the thought that representatives of both small and large vendor organizations should be included in case their views were radically different.

Some discussion then followed concerning the nature of the components supplied by vendors. One opinion was that shrink-wrapped products are somehow different than components meant to be integrated with others. Other opinions were that these kinds of products must be treated the same — the inability to integrate shrink-wrapped products was a major part of the problem being faced by system integrators.

A specific suggestion was made with the area of standards for document interaction. In particular, the suggestion was made that the DoD must become a much more active player in the debate between standards such as OpenDoc, OLE, and others.

Another area of interest was to create some guidelines on how COTS vendors should document the major interfaces of their components to make it easier for system integrators to choose between components that support similar interfaces. Some experiences expressed during the symposium were directly relevant here.

There appears to some fundamental difference between the applicability of COTS components in safety critical systems as opposed to MIS-like systems. There are much more stringent requirements on the former, so the lack of control and predictability makes COTS use more difficult. A number of others picked up on this distinction, describing their frustrations in using COTS within safety critical applications. Problems they raised included obtaining safety critical information from COTS vendors, maintenance of COTS components, and certification of components.

A specific suggestion then followed concerning the setting up of some certification guidelines, and ideally an organization to certify COTS components. The notion of “qualified products” may be a good start in this direction. It was suggested that SEI or MCC may be the place for such work.

The discussion moved to government acquisition regulations. It was suggested that changes in progress would have an important impact in this area. There needs to be a way to represent the concerns of the system integrators into these changes, and into the interpretation of any regulations defined. Other committees and groups are performing such a role. Perhaps the role of this group could be to coordinate and assist in their efforts.

One suggestion was that three separate goals could be defined, and we should choose which one we are attacking. These goals are

- how to use COTS components in a cost effective way in safety-critical applications;
- influencing COTS vendors to improve their components, their interfaces, and their documentation;
- qualification of COTS components for use in application domains.

It was suggested that one class of people that particularly need help and guidance are those involved in the writing and interpretation of Requests For Proposals (RFPs). They need some help with understanding the full life-cycle with which COTS components can be used — from before RFP generation through into system maintenance. They need to know what key decision points exist, how use of COTS components changes the life-cycle, how cost models are impacted by use of COTS components, and so on. It would be good to produce examples and guidance for this class of people, as it is here that much of the problems and over-expectations begin.

Further analysis of cost models was also suggested. This would help in understanding the buy versus build decisions that many people are facing. Some form of strawman model of the use of COTS components as part of a system life-cycle might make the problem more concrete.

It was suggested that another impact of the use of COTS components had not yet been brought up. Some people are interested in the effects on process maturity of the use of COTS components rather than building the software yourself. For example, some suggest that if you mainly integrate existing components then you don’t need to worry all that much about process maturity. This could be addressed by the SEI in terms of the Capability Maturity Model (CMM).

The final comment made by the audience was a plea for realism in any follow-on work. In particular, it was suggested that many of the problems lie in the details and specifics of the problem of system integration using COTS components. Any future workshop should look at some detailed, specific problems, and suggest concrete solutions and best practices. Examples may include how to detect, report, and fix errors that occur in systems composed of multiple COTS components.

REPORT DOCUMENTATION PAGE

1a. REPORT SECURITY CLASSIFICATION
Unclassified

1b. RESTRICTIVE MARKINGS
None

2a. SECURITY CLASSIFICATION AUTHORITY
N/A

3. DISTRIBUTION/AVAILABILITY OF REPORT
Approved for Public Release
Distribution Unlimited

2b. DECLASSIFICATION/DOWNGRADING SCHEDULE
N/A

4. PERFORMING ORGANIZATION REPORT NUMBER(S)
CMU/SEI-95-SR-007

5. MONITORING ORGANIZATION REPORT NUMBER(S)
N/A

6a. NAME OF PERFORMING ORGANIZATION
Software Engineering Institute

6b. OFFICE SYMBOL (if applicable)
SEI

7a. NAME OF MONITORING ORGANIZATION
SEI Joint Program Office

6c. ADDRESS (city, state, and zip code)
Carnegie Mellon University
Pittsburgh PA 15213

7b. ADDRESS (city, state, and zip code)
HQ ESC/ENS
5 Eglin Street
Hanscom AFB, MA 01731-2116

8a. NAME OFFUNDING/SPONSORING ORGANIZATION
SEI Joint Program Office

8b. OFFICE SYMBOL (if applicable)
ESC/ENS

9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER
F19628-95-C-0003

8c. ADDRESS (city, state, and zip code))
Carnegie Mellon University
Pittsburgh PA 15213

10. SOURCE OF FUNDING NOS.

PROGRAM ELEMENT NO	PROJECT NO.	TASK NO	WORK UNIT NO.
63756E	N/A	N/A	N/A

11. TITLE (Include Security Classification)
Proceedings of the SEI/MCC Symposium on the Use of COTS in Systems Integration

12. PERSONAL AUTHOR(S)
Edited by Alan W. Brown, David J. Carney, and Maureen D. McFalls

13a. TYPE OF REPORT	13b. TIME COVERED	14. DATE OF REPORT (year, month, day)	15. PAGE COUNT
Final	FROM TO	June 1995	44

16. SUPPLEMENTARY NOTATION

17. COSATI CODES		
FIELD	GROUP	SUB. GR.

18. SUBJECT TERMS (continue on reverse of necessary and identify by block number)
COTS, commercial off-the-shelf, systems integration, integrated systems, symposium, Software Engineering Institute, SEI, MCC

19. ABSTRACT (continue on reverse if necessary and identify by block number)

The SEI/MCC Symposium on the Use of COTS (commercial off-the-shelf) in Systems Integration took place at the Software Engineering Institute (SEI) in Pittsburgh, Pennsylvania, on January 10-11, 1995. The symposium focused on two key points: the Department of Defense need for integrated systems, and its increasing reliance on acquiring software through commercial sources. The interrelationships between these points formed the conceptual basis for this symposium. These proceedings provide a record of the presentations and some of the main highlights from the discussions. The main body of this report is a set of notes from each of the panels.

(please turn over)

20. DISTRIBUTION/AVAILABILITY OF ABSTRACT
UNCLASSIFIED/UNLIMITED ■ SAME AS RPT □ DTIC USERS ■

21. ABSTRACT SECURITY CLASSIFICATION
Unclassified, Unlimited Distribution

22a. NAME OF RESPONSIBLE INDIVIDUAL
Thomas R. Miller, Lt Col, USAF

22b. TELEPHONE NUMBER (include area code)
(412) 268-7631

22c. OFFICE SYMBOL
ESC/ENS (SEI)

ABSTRACT — continued from page one, block 19