



**CarnegieMellon**  
**Software Engineering Institute**

---

Pittsburgh, PA 15213-3890

**SEI Monographs on the Use of  
Commercial Software in Government  
Systems**

**COTS and Open Systems**

Patricia Oberndorf

*February 1998*

## About this Series

Government policies on the acquisition of software-intensive systems have recently undergone a significant shift in emphasis toward the use of existing commercial products. Some Requests for Proposals (RFPs) now include a mandate concerning the amount of COTS (commercial off-the-shelf) products that must be included. This interest in COTS products is based on a number of factors, not least of which is the spiraling cost of software. Given the current state of shrinking budgets and growing need, it is obvious that appropriate use of commercially available products is one of the remedies that might enable the government to acquire needed capabilities in a cost-effective manner. In systems where the use of existing commercial components is both possible and feasible, it is no longer acceptable for the government to specify, build, and maintain a large array of comparable proprietary products.

However, like any solution to any problem, there are drawbacks and benefits: significant tradeoffs exist when embracing a commercial basis for the government's software systems. Thus, the policies that favor COTS use must be implemented with an understanding of the complex set of impacts that stem from use of commercial products. Those implementing COTS products must also recognize the associated issues—system distribution, interface standards, legacy system reengineering, and so forth—with which a COTS-based approach must be integrated and balanced.

In response to this need, a set of monographs is being prepared that addresses the use of COTS software in government systems. Each monograph will focus on a particular topic, for example: the types of systems that will most benefit from a COTS approach; guidelines about the hard tradeoffs made when incorporating COTS products into systems; recommended processes and procedures for integrating multiple commercial products; upgrade strategies for multiple vendors' systems; recommendations about when not to use a commercial approach. Since these issues have an impact on a broad community in DoD and other government agencies, and range from high-level policy questions to detailed technical questions, we have chosen this modular approach; an individual monograph can be brief and focused, yet still provide sufficient detail to be valuable.

## About this Monograph

This monograph offers a practical rather than theoretical approach to the issues of COTS and open systems. While there are several potential issues with COTS and open systems that need to be considered when making actual decisions, this paper is principally aimed at clarifying the general understanding of what each is. Thus, it does not attempt to address the drawbacks and pitfalls of using either COTS products or an open systems approach, but rather is intended to reveal the conceptual relationship between COTS products and open systems.

# COTS and Open Systems

## 1 Introduction

Making greater use of commercial-off-the-shelf (COTS) products is becoming increasingly popular. Everyone from industry executives to Congress is suggesting that leveraging commercial capabilities will save time and money while improving the performance of software-intensive systems. At the same time, use of an “open systems” approach to develop systems has been gaining popularity, with visions of systems that are “plug-and-play” compatible, where components from one supplier can be easily replaced by those from another supplier.

Advocates of such “open” systems often confuse the use of an open systems approach with the use of COTS products, making it difficult for the average manager or engineer to know just what he/she should be doing to develop (and maintain) systems more effectively. Further confusing the problem is that “open systems” is a concept that is defined in various ways by different people.

These two concepts—the use of COTS products and the creation of open systems—are closely related and complementary, although definitely not synonymous. The purpose of this monograph is to clarify what each is, explain the differences between them, and explain what their relationship is.

## 2 What Is “COTS”?

For the government, increased use of commercial products holds the hope of getting, at a reasonable cost, something that already performs the functions needed by government systems. Such an approach minimizes the difficulties of developing government-unique system components; it holds out the promise of fast, efficient acquisition of cheap (or at least cheaper) component implementations.

To encourage this approach, then-Secretary of Defense William Perry directed in June 1994 that DoD acquisitions should make maximum use of performance specifications and commercial standards, thus increasing the opportunities to make use of commercial products. This was followed by the Federal Acquisition Streamlining Acts of 1994 and 1995 from Congress, which directed the increased use of commercial items; these have since been incorporated into the FAR to help implement the new approach. New and rewritten DoD policy documents (e.g., DOD Directive 5000.1 and DOD Regulation 5000.2) implementing the ideas have also been released.

The motivation behind these directives and laws has several sources.

- Government organizations traditionally spend a great deal of effort defining (to the lowest level of detail) the desired characteristics of systems and how the contractors are to build those systems to achieve those characteristics. Thus many resources are expended developing systems and components that often already exist—or exist in good enough form with nearly the same capabilities—elsewhere. The prevailing (and time-consuming) approach is to develop everything from the ground up every time, yielding a unique system each time. The

result is systems that are very expensive, with only one customer to bear the development and maintenance costs over the life of the component or system.

- To make matters worse, commercial organizations are investing a great deal in advancements in technology, but the unique qualities of systems developed the traditional way make it impossible for these systems to take advantage of the advanced technologies.
- The long lead times of unique systems also work against capitalizing on new advances; historically the DoD has fielded technology that is more than ten years old.

Shifting to an approach based on leveraging the commercial marketplace offers several attractions. The hope is to lower costs by sharing them with other users, amortizing them over a larger population while taking advantage of the investments that industry is making in new technologies.

## 2.1 Defining “COTS”

The term “COTS” is the acronym generally used to describe commercial products. It commonly refers to things that one can buy, ready-made, from some manufacturer’s “store shelf” (through a catalogue or from a price list). This usage, however, is imprecise and not universally accepted. The government, which needs a precise definition for procurement, has defined the term “commercial item,” a definition we use in this monograph. The full text of this definition can be found in Part 2 of the current Federal Acquisition Regulations (FAR); the following is a summary.

*A commercial item is*

- (1) property<sup>1</sup> customarily used for nongovernmental purposes and has been sold, leased, or licensed, or offered for sale, lease or license to the general public;
- (2) any item evolved from an item in (1) through advances in technology and is not yet available commercially but will be available in time to satisfy the requirement;
- (3) any item that would satisfy (1) or (2) but for modifications customarily available in the commercial marketplace or minor modifications made to meet Federal Government requirements;
- (4) any combination of items meeting (1) - (3) above;
- (5) services for installation, maintenance, repair, training, etc. if such services are procured for support of an item in (1), (2), or (3) above, as offered to the public or provided by the same work force as supports the general public, or other services sold competitively in the commercial marketplace;
- (6) a nondevelopmental item developed exclusively at private expense and sold competitively to multiple state and local governments.

---

<sup>1</sup> “Property” in this definition explicitly excludes real property.

The key point is that COTS products are developed by a commercial entity for commercial purposes and for the general public. Most people would agree that these ideas approximate what they mean by COTS products. The salient characteristics are

- it exists a priori from a commercial vendor
- it is available to the general public
- it can be bought (or leased or licensed)

## 2.2 Defining NDI

A closely related term often heard in government circles is “nondevelopmental item” (NDI).

A *nondevelopmental item* is:<sup>2</sup>

- (1) any previously developed item used exclusively for government purposes by a federal, state, or local agency or government or by a foreign government that has a mutual defense agreement with the U.S.;
- (2) any item described in (1) above that requires only minor modification or modifications normally available in the commercial marketplace to meet requirements;
- (3) any item being produced that does not meet (1) or (2) above only because it is not yet in use.

The key point here is that NDI refers to something already developed by someone else. It might have been developed by a commercial interest, but typically it will have been developed for some other government, department, or agency. Hence, what is commonly called “government off-the-shelf” (GOTS) is a form of NDI. A large-scale example of an NDI would be a fighter aircraft developed by some other nation. A more meaningful example in the current context would be a radar developed for one aircraft that is available for use in another aircraft. The salient characteristics of a nondevelopmental item are

- it exists, although not necessarily off some vendor’s “shelf”
- it is available, although not necessarily to the general public
- it can be obtained for use, although perhaps not by purchase or lease

## 2.3 Comparing COTS and NDI

There are differences between these definitions of commercial item and nondevelopmental item and those that appeared in the previous FAR. In particular, commercial items used to be considered a subset of NDI, but now this position is reversed, since a restricted form of NDI qualifies as a commercial item (see item (6) under *commercial item*). In addition, support services such as installation and training are now also defined as commercial items.

---

<sup>2</sup> The following definition is also a summary, and the full text is in Part 2 of the FAR.

While these definitions may not be ideal,<sup>3</sup> they do characterize the features that are of interest when we speak of the potential benefits of using COTS products. To summarize, nondevelopmental and commercial (COTS) items are similar in that they both already exist, which is what makes them attractive. They are different in that COTS products usually appear in some sort of catalogue or price list, whereas it may be more difficult to discover the existence of NDI. The range of possibilities opened up by NDI is broader than what COTS products can offer, but by using NDI (as opposed to COTS products) the DoD likely loses some of the advantage of commercial leverage, in which a broad base of other users shares interest—and cost.

## 2.4 Problems with COTS

Unfortunately, those who have followed the COTS path have been learning the hard way that “just buying COTS” does not necessarily lead to all of the desired benefits. Problems and sources of risk are also introduced by the use of COTS products.

First, COTS products do not necessarily conform to any recognized interface standards. This leads to two other problems:

- *Lack of commonality with other products.* It is possible (in fact, likely) that using a closed COTS product commits the user to proprietary interfaces and solutions that are not common with any other product, component, or system. This will result in integration and interoperability difficulties.
- *Long-term maintenance issues.* If the sole objective of a system upgrade is to capture new technology more cheaply, then the use of COTS products may suffice. But many DoD systems have a 30- to 50-year lifetime or more,<sup>4</sup> while the average COTS component is upgraded every 6 to 12 months and new technology emerges about every 18 to 24 months. Thus money that is saved by using a COTS product with proprietary interfaces can quickly be lost in maintenance as products and their interfaces change with the marketplace. Even if the expected lifetime of a system is only 5 to 10 years, the fluctuations in COTS products and technology result in a state of constant change for any system employing them. Without interface standards, changes in the marketplace can impose unanticipated and unpredictable changes to systems dependent on closed commercial products.

---

<sup>3</sup> For example, how safe is a “minor modification,” and what if it looks like a vendor has a product, but in reality it is just vaporware?

<sup>4</sup> Source: Mr. Roy Willis, Principal Assistant Deputy Undersecretary of Defense for Logistics, briefing on June 12, 1996.

Second, the use of COTS products as opposed to custom-developed code implies a loss of control. Vendors' decisions now have much more effect on your plans and likelihood of success.

- *Vendors' schedules.* Whether the capabilities needed for a system will be available is subject to market forces and vendors' objectives in the market. The capability you need may not be highest priority for the vendor.
- *Vendors' license agreements.* License agreements can have a tremendous impact on a system's architecture and other key features. The ability to accommodate your needs will depend on your ability to negotiate successfully with the vendor.
- *Product discontinuation.* Vendors can decide to discontinue a product; vendors can go out of business; mergers can lead to abandonment of a former line of business. All of these marketplace dynamics can result in discontinuation of a product on which your system depends.

### 3 What Is an Open System?

An "open system" generally calls to mind a system that is flexible and adaptive, and "open" to the use of many products from different sources. To many people, the phrase "open system" carries with it an image of easy "plug-and-play" between components and products that were not necessarily originally designed to work together. It also holds out the promise of being able to take immediate advantage of the fast-moving commercial marketplace, because it should be easy to plug in new technology, either in place of an old component or as a new extension of the system. The idea of open systems is particularly attractive in situations where the rate of technology change or component costs are relatively high.

Many initiatives have been taken, both in the DoD and in individual services, agencies, and companies, to promote an open systems approach. For example, in November 1994 then-Undersecretary of Defense (Acquisition and Technology) Paul Kaminski directed "that 'open systems' specifications and standards be used for acquisition of weapon systems electronics to the greatest extent practical." In addition, DODD 5000.1 and DOD 5000.2-R have been rewritten to reflect the increased importance of open systems to the DoD, requiring that an open systems approach be considered for all components within DoD system development, not just weapon systems electronics.

Just as with "COTS," many different definitions of "open system" have been offered by various authors. The DoD Open Systems Joint Task Force (OS-JTF)<sup>5</sup> uses the following definition:

A system that implements sufficient open specifications for interfaces, services, and supporting formats to enable properly engineered components to be utilized across a wide range of systems with minimal changes, to interoperate with other components on local and remote systems, and to interact with users in a style that facilitates portability.

---

<sup>5</sup> The OS-JTF was chartered in November 1994 to facilitate the implementation of open systems in the DoD.

This definition captures the idea that there must be some rules that all “open” components and products follow; the rules are provided by the “open specifications for interfaces, services, and supporting formats.”

A more detailed definition that further illuminates the important qualities of these “open specifications” can be found in [OS:PP],<sup>6</sup> where an open system is defined as

a collection of interacting software, hardware, and human components, designed to satisfy stated needs,

with the interface specification of components<sup>7</sup>

- fully defined
- available to the public
- maintained according to group consensus, and

in which the implementations of components are conformant to the specification.

This second definition also describes the criteria for the interface specifications (i.e., the three bulleted items). Not only must the interface specification be fully defined, but it must also be available to the public. This implies that cost and public access must not be prohibitive constraining factors. That is, the specification cannot be available only to a select group of people who have some special interest. Anyone must be able to obtain a copy of the specification (perhaps at the cost of duplication and distribution, perhaps even at the cost of a *small* license fee), and they must also be free to produce and sell implementations of that specification. Finally, it is important that the specification is of interest to a wide range of parties and that it is not under the exclusive control of any single party.

To this end, the definition includes the idea that maintenance of the specification is by group consensus.<sup>8</sup> In particular, there are actually two dimensions along which to consider consensus: One is formal acceptance in the sense of pedigree of a standard specification, the other is market acceptance in the sense of both vendor interest (number of products on the market using the standard) and user interest (people buying and using the products). These dimensions make it possible to understand how even popular proprietary systems, such as Microsoft Windows 95, can have a place in DoD open systems.

Both of these definitions emphasize the salient features of open systems: the interface specification. In addition, they both remind us that the specifications are not enough: a system is made up of component implementations, and in an open system they must properly implement the interface specifications (i.e., “conform” to them) and use those specifications as their means of interaction. The importance of this is that it is impossible to enjoy the benefits of standardization if implementations take liberties with the specification.

---

<sup>6</sup> [OS:PP] was developed under support and guidance from the Navy’s Next Generation Computer Resources Program during 1993-1994.

<sup>7</sup> “Interface specification” covers interfaces, services, support formats, e.g., for data, and so forth.

<sup>8</sup> Taken together, these criteria come close to requiring that the interface specification be a “standard.” But because there are so many varied ideas of what a standard is, this definition spells out the criteria, thus allowing the use of a variety of consensus-based specifications.



While the interface specifications are key, it should not be expected that absolutely every interface in a system adhere to a standard for a system to be called open. Openness is not an all-or-nothing situation. As suggested in the OS-JTF definition (by the phrase “sufficient open specifications”), openness is a matter of degree, and sufficiency is determined by the goals and circumstances of each individual system.

### 3.1 Interface Standards Support Open Systems

Most interface standards currently used in software systems are for application program interfaces (APIs), data formats, or protocols. Checking these against the open system definitions given above, we find they fit very well.

- For all of these kinds of interface standards, fully defined specifications exist; these specifications guard against variant implementations, which in turn undermine the desired compatibility.
- Interface standards are made widely available to the public to generate a thriving market for components that can be plugged together.
- Interface standards are maintained using many forms of group consensus; this precludes one vendor or group from making arbitrary changes to the interface standard that will limit competition and availability of alternative products.
- For many of these interface standards (e.g., IEEE 1003.1 - POSIX<sup>9</sup>) it is possible to tell whether or not a given implementation really matches the specification; this is called conformance, and for many standards conformance tests have been developed. If the implementations all match the specification/standard closely enough, then part of the incompatibility between components can be reduced if not eliminated, and it may be possible to “plug” them into a system and get them to “play” with the other components. On the other hand, if implementations only loosely implement the standard or if incompatible interpretations cannot be detected before trying to integrate a component into the system, then it is less likely that the necessary flexibility and adaptability can be realized.

However, interface specifications alone are generally insufficient to ensure full “plug-and-play” operation. In practice, the real interface between two components of a system consists of all the assumptions that each makes about the other. APIs, data formats, and protocols address a large number of these assumptions, but by no means all of them. It will take further investigations to determine the full set of interface knowledge that must be standardized to ever get close to an ideal “plug-and-play” system creation process.

### 3.2 Open Systems May Be Implemented Without COTS or NDI

There is nothing in either of our definitions of open system that *requires* implementation with commercial products. It is possible to create an open system, based on interface standards, in which no COTS products or NDI are used. This might be even necessary in a situation where, for example, no COTS product conforming to the interface standard also meets other system requirements, such as requirements for real-time performance or security. Although one would

---

<sup>9</sup> POSIX: Portable Operating System Interface

not gain the economic and schedule advantages of using an existing implementation, the interface standards could still provide the framework for future evolution of the system. Some product could eventually appear that met all the requirements and conformed to the interface standard. In the meantime, the system enjoys the clarity and stability of a well-defined specification.

### 3.3 Advantages of Open Systems

Some of the disadvantages that accompany the use of COTS, as discussed above, are lessened with an open systems approach.

- Interface standards provide a source of stability in the midst of the constant changes in the COTS marketplace. This translates into improved system supportability over its lifetime.
- Open systems can provide an overall framework for system evolution and long-term stability, even if there are no COTS products that can be used to implement the system. This derives from the fact that a system can expand and evolve more predictably as the standards evolve.
- Open systems also help protect programs from getting locked into proprietary solutions. A given COTS product may be great for the job, but if using it forces everything else about the system to be tailored to its architectural view and its proprietary interfaces, then the ability to migrate cost-effectively to other products and other technologies in the future has been compromised: There is probably no plug-and-play outside that one vendor's sphere of influence. This situation is particularly painful when the vendor stops supporting the product or goes out of business altogether, forcing significant component and system changes. The use of widely accepted standards provides an alternative to such "vendor-lock."
- Interoperability with other systems is enhanced, to the extent to which those other systems utilize the same interface standards for data formats, for communications, and for sharing services.

### 3.4 Problems with Open Systems

Paradoxically, given the desire to produce systems more quickly, the emphasis on standards can actually be somewhat of an inhibitor.

- Some standards efforts, in their desire to achieve maximum consensus, have very long cycle times (five or more years), which certainly do not fit well with product development and release cycles. This conflict is of concern and is being addressed by some standards groups,<sup>10</sup> but it has led some projects to adopt alternatives, such as consortia-produced standards (e.g., the Common Object Request Broker Architecture (CORBA)) and de facto industry standards (e.g., Windows 95). While these are often practical alternatives to "formal" standards, they do have attendant risks; for example, the de facto standards are often proprietary.
- Up-front costs for an open system may actually be higher than for a custom-developed system. This is because there are new activities associated with the selection and profiling of standards that must be incorporated.

---

<sup>10</sup> For example, IEEE has investigated and implemented various ways to speed up their approval cycles.

In addition, to the extent to which open systems are implemented using COTS products, they are subject to some of the same problems as COTS-based systems that do not use interface standards.

## 4 The Relationship of COTS and Open Systems

While COTS and open systems are not the same thing, they are complementary. COTS products hold the potential for cost-effective acquisition of components and advancing technology. However, they are not necessarily open: It is possible to use COTS products without creating an open system. Open systems provide stability and a framework for the effective use of COTS products and other NDI through the use of interface standards; well-chosen interface standards can outlast any particular product, vendor, or technology. But it is possible to create an open system without significant use of COTS products or NDI.

This complementary relationship between COTS products and open systems is illustrated in the following chart.

	<b>Open Systems</b>	<b>Non-Open Systems</b>
<b>COTS &amp; Nondevelopmental Items</b>	standards-based COTS and NDI	other COTS and NDI
<b>Developmental Items</b>	standards-based new development	everything else

There are important system goals that are shared by the use of COTS products and an open systems approach. Among these are

- improving the quality and performance of systems
- developing them more quickly
- sustaining them more cost-effectively

Both COTS and open systems are means to these ends. The greatest advantage, however, can be gained by using the two of them together.

COTS and open systems also share a realm of applicability. That is, it is expected that both can have a positive impact on either new system development or upgrades of legacy systems. Although there is generally more decision-making freedom in the case of a new development, an open systems approach can nevertheless shape an evolutionary path for a legacy system that will help turn it into a more flexible and maintainable system.

Another trait shared by COTS and open systems is that the depth of understanding and technical and management skills required on a project team using them are not necessarily diminished or decreased because of the use of COTS or open systems. Arguably, the skills and understanding needed *increase* for both because of

- the potential complexity of integration issues
- the need to seriously consider longer term system evolution as part of initial development
- the need to make informed decisions about which products and standards to use

## 5 Conclusion

In this paper we have explored the definitions of COTS and open systems. With each we examined the benefits they bring to the development, maintenance, and evolution of systems. This monograph has not provided a thorough examination of the pros and cons of the use of either COTS products or open systems, however. There are drawbacks to each, which must be traded off against the potential benefits. The purpose here has been only to discuss the similarities, differences, and relationship between these two popular concepts.

Key points to remember include

- COTS products and an open systems approach are not synonymous, but they are complementary.
- COTS products and an open systems approach both support important system goals, such as improving the quality and performance of systems, developing them more quickly, and sustaining them more cost-effectively.
- The greatest advantage can be gained from using these two approaches together.
- Be prepared for requirements for new skills and understanding (e.g., how to choose the right standard or product) with either of these approaches.

## 6 References

- [DOD1] DOD Directive 5000.1 *Defense Acquisition*, 15 March 1996.
- [DOD2] DOD Regulation 5000.2 *Mandatory Procedures for Major Defense Acquisition Programs (MDAPs) and Major Automated Information System (MAIS) Acquisition Programs*, 6 October 1997 (date of Change 2).
- [FARs] *Federal Acquisition Regulations*.
- [OS:PP] Meyers, B. Craig, and Oberndorf, Patricia A., *Open Systems: The Promises and the Pitfalls*, SEI course, 1996.

## **Acknowledgments**

The following people have provided helpful comments and suggestions for this monograph: Linda Brown and Thomas Bozek (OSD), David Carney (SEI), and Mick Hanratty and Arv Larson (Open Systems Joint Task Force).

## **Feedback**

Comments or suggestions about these monographs are welcome. We want this series to be responsive to the real needs of government personnel. To that end, comments concerning inclusion of other topics, the focus of the papers, or any other issues are of great value in continuing this series of monographs. Comments should be sent to:

Editor  
SEI Monographs on COTS  
Software Engineering Institute  
Carnegie Mellon University  
Pittsburgh, PA 15213  
cots@sei.cmu.edu