Introduction to the Defense Information Infrastructure (DII) Common Operating Environment (COE)

Pamela Engert and Julie Surer
The MITRE Corp.

The Defense Information Infrastructure (DII) Common Operating Environment (COE) provides a foundation for building interoperable command and control systems using reusable software components. The DII COE is comprised of many concepts. It is, in one sense, a set of reusable software components. In addition, the DII COE is also a system architecture that allows the components to be reused in command and control systems, as well as the standards and guidelines that define how the components can be constructed. This article briefly defines the DII COE architecture and describes the defined compliance criteria.

THE DII COE PROVIDES A foundation for building interoperable systems through the use of reusable software components (building blocks). The DII COE can be characterized as a number of things, depending upon one's point of view. It is an architecture, a collection of reusable software elements, a software infrastructure, and a set of guidelines and standards. More importantly, however, is that it provides a common platform (or foundation) for building interoperable systems. Therefore, one could think of the DII COE as one component of a system architecture, as it is an implementation of the Joint Technical Architecture (JTA). One could also think of the DII COE as an approach to software development — how to go about building interoperable systems on a common platform. Finally, it is important to realize that the DII COE is not a system, but a set of building blocks from which a system can be built. Global Command and Control System, Global Combat Support System, and service unique programs (like Air Force Theater Battle Management Corp. Systems) are building their systems on top of the DII COE foundation.

Definition of DII COE

DII COE is a software infrastructure, a collection of reusable software components, a set of Application Program Interfaces (APIs), and a series of specifications and standards for developing interoperable systems.

The DII COE taxonomy defines two layers of reusable software components: infrastructure services, which include the DII COE kernel services, and the underlying commercial-off-the-shelf (COTS) operating systems. Infrastructure services address the movement of data through the network and includes distributed computing and web services. The kernel provides low-level services, including a desktop environment, runtime tools, and basic system and security administration.

Common support applications provide services that address common command and control functionality, for example, mapping and message processing.

Standard APIs provide the interfaces between mission applications and reusable software components of the DII COE. Mission applications are developed on top of the DII COE and provide mission domain specific functionality.

DII COE Compliance

DII COE compliance measures the degree to which a software component, including mission applications, can plug-and-play (the degree of interoperability) in the COE. The goal of meeting DII COE compliance is to ensure seamless software component integration and system operation. A software component is assessed in four categories for COE compliance: runtime environment, style guide, architectural compatibility, and software quality. Although all four categories of compliance are considered in an assessment, the primary focus is on the runtime environment.

Runtime Environment

The runtime environment category assesses how well the proposed software segment or system (collection of segments) functions within the COE environment and the extent that the software reuses COE components. The evaluation determines if the proposed software segment can be added to the system without adversely affecting system interoperability. Segments are evaluated against the checklist in the DII COE integration and runtime specification.

Style Guide

This category assesses the user interface of a segment for consistency and conformance to the checklist in the user interface specification for the DII.

Architectural Compatibility

This category determines if proposed software is architecturally sound and compatible with the COE. Unlike the runtime environment and style guide categories, the architecture compatibility category has not been defined to date.

Software Quality

This category assesses software for portability and integration into the COE. The level of life cycle maintenance support associated with the proposed COE component is estimated. The assessment of software quality compliance level is achieved by using complexity and quality metrics collections, portability analysis, and COE API compliance analysis. COTS analysis tools are used for automated, nonintrusive compliance checking.

Runtime compliance is expressed in terms of eight levels of compliance defined in the DII COE integration and runtime specification. Level 5 compliance is considered "minimal DII" and indi-

cates that segmented applications can share the same DII COE kernel without interfering with one another, that the segments can be installed using standard tools, and that the segments conform to the DII user interface specification. Level 8, or "full DII," implies 100 percent compliance with all DII COE runtime and user interface criteria.

DII COE Compliance Mandates

Several DII COE mandates exist for Department of Defense systems. The Office of the Under Secretary of Defense has issued a directive that all command, control, communications, computers, and intelligence systems (C4I) be JTA compliant and mandates DII COE compliance Level 5 for legacy systems and Level 6 for new systems. The goal is to reach Level 7. The JTA mandates DII COE Level 5 compliance with a goal of Level 8 for all C4I systems.

Summary

The DII COE will be a key contributor in achieving the C4I vision of providing warriors with technically advanced, interoperable command and control systems. The DII COE provides a foundation for building interoperable systems through the use of reusable software components. ◆

About the Authors

Pamela Engert joined the MITRE Corp. in Bedford, Mass., in 1986 and is now a lead engineer. She supports the System Engineering Process Office and the Acquisition Development Office. She has a bachelor of science degree in mathematics and computer science and a master's degree in engineering management.

The MITRE Corp.
202 Burlington Road
Bedford Mass. 01730-1421
Phone: 781-271-3138
Fax: 781-271-2101
E-mail: pengert@mitre.org

Julie Surer is a principal engineer at the MITRE Corp. in Bedford, Mass., where she supports the ESC/DIE (Electronic System Center) Chief Architect's Office. Surer is the Air Force DII COE chief engineer. She is also the corporate representative responsible for DII COE technical activities in the Air Force. She is also the corporate representative to The Open Group, an international standards consortium. She has a bachelor's degree from the University of Florida in engineering science and a master's degree from the University of South Florida in electrical engineering.

The MITRE Corp.
202 Burlington Road
Bedford Mass. 01730-1421
Phone: 781-377-6809
Fax: 781-377-7779
E-mail: jsurer@mitre.org

DII COE Web Information Resources

For extensive, up-to-the-minute information on DII COE, visit DISA's DII COE site at
http://spider.dii.osfl.disa.mil/dii

DISA home page
http://www.disa.mil

For current information on DII COE in the Air Force, visit
http://www.esc-dii.hanscom.af.mil/Chief_Architect/Ca_home.htm.

For current information on DII COE mandates, visit
http://www.escdii.hanscom.af.mil/Chief_Architect/dii-coe/faq/faq.html

Note: The AF ESC/DII Web pages are restricted to clients with a .mil primary domain.

HQ AFCA DII COE home page
http://www.afca.scott.af.mil/