Architectural Guideline

For the
Common Training Instrumentation Architecture (CTIA)

Introduction

The intent of this guideline is to provide architectural guidance towards a component-based approach for development of the CTIA. IEEE 610.12 defines an architecture as a structure of components, their relationships and the principles and guidelines governing their design and evolution over time. Adherence to this definition delineates the most basic purpose of the CTIA.

Component-Based Development

Component-based development can be characterized as:

- The specification of an evolvable, malleable software architecture.
- The identification and selection of components that meet architecture and functional requirements.
- The negotiation of requirements versus off-the-shelf (OTS) functionality
- The integration of components into the architecture

Transformation of the Army has created urgent needs for modernization of live training capabilities across to Army. In order to meet these needs within the constrained budgets, STRICOM is pursuing smarter, more efficient means of developing the next generation Live Training Transformation (LTT) domain products. A component-based architecture-driven product line strategy is the approach for the development of the LTT domain. A component-based architecture can be thought of as a group of software elements (components) designed as generic, reusable parts, used to construct an application or system. Software components are built according to rules and guidelines as specified in an architecture specification. Component interfaces define the behaviors of components that are observable and useable by other components. Using interface specifications enables project teams to design software emphasizing the functionality provided by components and de-emphasizing the implementation detail of the component. The LTT project team envisions that many component applications will be accomplished though the use of OTS solutions. OTS solutions are typically not

built according to the rules of the architecture specification and consequently integration of the OTS package may be a challenging task.

Architecture Driving Principles

The architecture principles (driving requirements) for the LTT/CTIA have been initially defined but are expected to be refined as the program evolves. They are as follows:

Interoperability**:** Defined as the ability of a model or simulation to provide services to and accept services from another model or simulation and to use the services to enable them to operate effectively together.

Reusability**:** Defined as the ability to easily reuse conceptual and physical software constructs in order to satisfy the LTT functional capabilities.

Extensibilty**:** Defined as the ability of adding new functionality or enhancing existing functionality without restructuring the existing architecture.

Malleability**:** Defined as the ability for the architecture to adapt as technology changes and matures.

Commonly shared data**:** Applications should be integrated through commonly shared databases.

Modularity: The extent to which a system can be divided into a set of individual modules.

Security: The ability for the architecture to resist unauthorized attempts at usage and denial of service while still providing its services to legitimate users.

Commercial standards: Maximize commercial standards (e.g. interfaces) for player units and remote devices.

Business Case Objectives

Business case objectives are very much related to the architecture principles, after all architectures are business-specific collections of software components and other artifacts. Architectures are most effective when combined with a focused vision and a set of business objectives or goals. The LTT/CTIA vision is to utilize product line principles and practices to establish, execute and maintain a repository-based set of domain assets that support reusability across the LTT/CTIA domain. STRICOM's business objectives fit four categories: 1) reduce the time and cost of developing and maintaining software, 2) improve the quality of the systems we build, 3) avoid training system obsolescence and 4) maintain the flexibility to adopt new technologies with minimal effort. At this stage of the program candidate objectives have been identified based on the "things" that are the most important to the Government. Some of the business case objective candidates are:

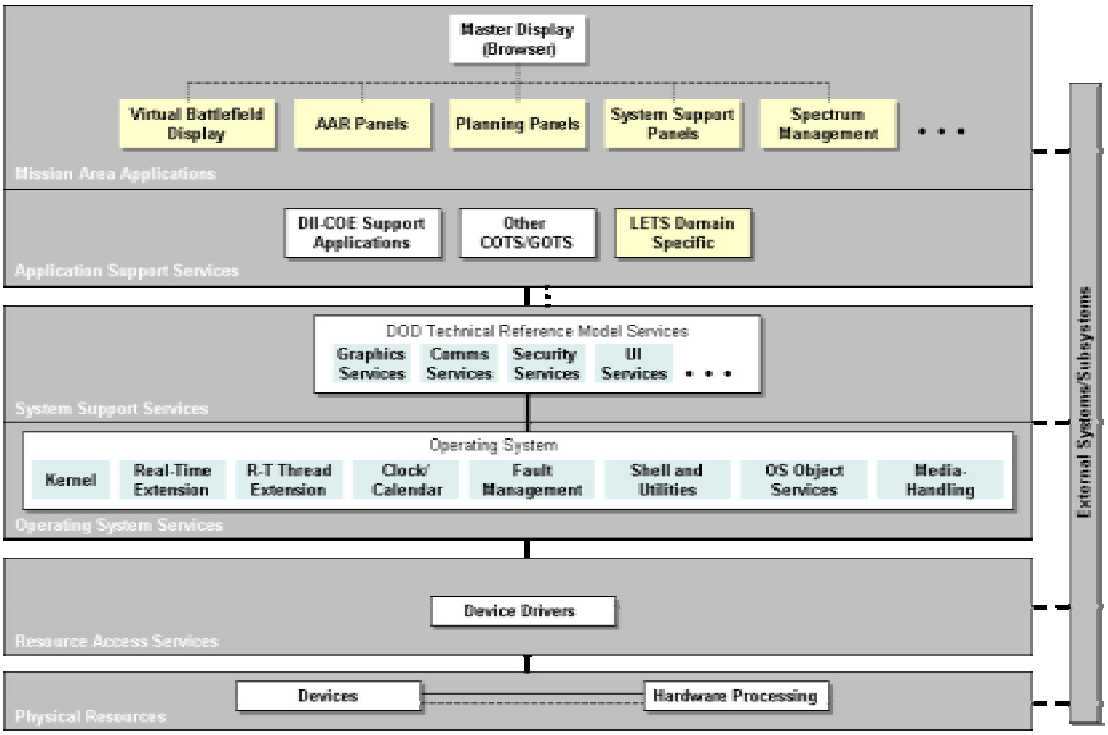
- Shorter development times (i.e., faster deployment)
- Improved reliability
- Improved quality
- Lower maintenance cost
- System flexibility
- Easier integration of new technology
- Greater opportunity for reuse
- Improved interoperability

Candidate areas of reuse or areas of importance to the LTT program are:

- Communications
- Planning
- Data visualization
- System support
- Data collection
- Spectrum management
- Tactical engagement simulation
- Data analysis
- Exercise control and monitor
- Data management
- Interoperability between constructive and virtual environments

The CTIA

The Government envisions the CTIA to be a component-based architecture. The candidate reuse areas mentioned above can be thought of as potential component boundaries. The Defense Information Infrastructure Common Operating Environment (DII COE) is one example of several component-based architectural frameworks and will be used in its modified form to illustrate a hypothetical interface view of the CTIA. The following diagram presents a hypothetical interface view of the CTIA.



Mission Area Applications:

- Mission-unique functionality
- Applications developed by Services/agencies
- Controlled and deployed by Services/agencies

Example: Think of the top level of the model containing a master display and all of the application addressed above. The master display could simply be a web browser along with some navigation and query tools provided by Application Support Services. The custom panel templates can be implemented by customizing a handful of Active Server Pages. These panels provide convenient access to the capabilities provided by the Application Support Services.

Application Support Services:

- Emphasizes interoperability via common view of data
- Functionality that is common within the domain
- Developed by Services/agencies
- Controlled and deployed by Services/agencies

Example: There are three boxes drawn in this layer: The first, DII-COE Support Applications, represent DII-COE compliant applications that may need to be web-enabled by CTIA developers. The second, other COTS/GOTS, are already web-enabled, and just need to be integrated into the user panels (e.g. Microsoft Office, Enterprise Management Tools, database tools such as natural language processors, etc). The third, LETS Domain Specific, are custom applications written to satisfy those requirements of the Functional Specification that are not adequately satisfied by COTS/GOTS (e.g. tools for automated generation of AAR products).

System Support Services:

Example: The database management system and the middleware reside here. The middleware may include the usual suspects, COM, CORBA, and EJB, as well as Simple Object Access Protocol (SOAP) and Internet servers to bridge the competing component models. These servers could be used to interface and integrate external systems such as ABCS.

Operating System Services:

Example: None of the above technologies dictates a particular platform. The operating system(s) adopted may be any of those sanctioned by DII-COE.

External Systems:

Example: External systems can be integrated and interfaced by the same technologies described above. Interfaces may be exposed to the web with a Simple Object Access Protocol

(SOAP) wrapper. They can then be made available to display interfaces with Active Server Pages/Java Server Pages (ASP/JSP) or made available to the database through XML middleware.

Points of Contact:

Mr. Robert Dixon – 407.384.3837, DSN: 970

Mr. Paul Watson – 407.384.5159, DSN: 970

Ms. Wanda Fuentes – 407.384.3914, DSN: 970