Applying Service Orientation to the U.S. Army's Common Training Instrumentation Architecture

Jeremy T. Lanman
U.S. Army PEO STRI
Orlando, Florida
jeremy.lanman@us.army.mil

Scott Clarke, Shawn Hillis, R. Darbin
General Dynamics C4 Systems
Orlando, Florida
scott.clarke@gdc4s.com
shawn.hillis@gdc4s.com
rowland.darbin@gdc4s.com

Dave Frank
AIT Engineering
Orlando, Florida
david.frank@aitengineering.com

ABSTRACT

The Common Training Instrumentation Architecture (CTIA) is one of the three architectures defined by the U.S. Army's Live Training Transformation (LT2) product line. It is used by LT2 products to define interoperability standards among live training applications to support force-on-force and force-on-target training. Using an introspective approach, honest dialog and user feedback, it was determined that CTIA must evolve to address technology obsolescence and meet the growing needs of the live training community. However, in order for the architecture to meet those needs, a Service Oriented Architecture (SOA) approach was identified as the preferred methodology. This paper documents the analysis process and methodology used by the Architecture team to apply service orientation to CTIA in order to address the long term goals of the LT2 product line. These goals include support for distributed training, mobile computing devices and cloud computing technologies.

The CTIA Architecture team utilized a series of workshops, SOA training and Human Centered Design (HCD) techniques to identify and prioritize the strategic business goals and objectives for the product line. As part of this effort, the team conducted an Open Group Service Integration Maturity Model (OSIMM) assessment to analyze and prioritize architecture attributes against the open services integration dimensions. Finally, the team selected and prioritized service oriented design principles which are being applied to the architecture in order to achieve those goals. The result of this process is a roadmap and high level design for the evolution of CTIA to a Service Oriented Architecture.

ABOUT THE AUTHORS

Jeremy T. Lanman, Ph.D. is the lead systems architect for the U.S. Army PEO STRI Common Training Instrumentation Architecture (CTIA) and Consolidated Product-line Management (CPM) construct supporting the Army's Live Training Transformation Family of Training Systems. His professional experience includes 10 years of DOD acquisition and systems engineering of military modeling and simulation, and training systems. His research interests include service-oriented computing, live, virtual, and constructive integration and analysis, and distributed computing. Dr. Lanman received his Bachelor of Science in Computer Science from Butler University, Master of Science in Software Engineering from Embry-Riddle Aeronautical University, and Ph.D. in Modeling and Simulation from the University of Central Florida.

Scott Clarke, chief architect at General Dynamics C4 Systems, has spent the last 10 years developing live training systems for the U.S. Army. He has been the primary technical lead in the development of the Army's Common Training Instrumentation Architecture (CTIA). Currently, he is the chief architect for the Consolidated Product Line Management (CPM) Live Training Transformation (LT2) Framework. Mr. Clarke is a former U.S. Army officer and has served in various leadership positions including tank company commander and platoon leader. He earned a Bachelor of Science in Computer Science from the Rochester Institute of Technology.

Shawn Hillis, lead engineer at General Dynamics C4 Systems, has spent 13 years developing live training systems for the U.S. Army. He has been a developer, team, and technical lead in the development of the Army's Common Training Instrumentation Architecture (CTIA) program and related Live Training Transformation (LT2) products such as the Homestation Instrumentation Training System (HITS) and the One Tactical Engagement Simulation

System (OneTESS). Mr. Hillis is currently the software tasking lead for the CTIA post-deployment sustainment and support (PDSS) efforts as well as the evolution of the CTIA architecture to a Service Oriented Architecture. He earned a Master of Science in Computer Engineering from the University of Central Florida.

Rowland Darbin, product line manager and software architect at General Dynamics C4 Systems, has spent nine years supporting Commercial off the Shelf products for virtual and constructive simulation and three years supporting live training. Mr. Darbin is currently the product line manager for the Consolidated Product Line Management (CPM) Product Line. He earned a Bachelor of Science in Computer Engineering from the University of Central Florida.

Dave Frank, software architect at AIT Engineering, has spent 10 years developing live training and simulation systems. He has been a developer, technical lead and architect on the U.S. Army Common Training Instrumentation Architecture (CTIA) as well as a member of the development and support team for the Combat Training Center Instrumentation System (CTC-IS) program. Mr. Frank is currently leading the design effort for evolving CTIA to a Service Oriented Architecture. He has a Bachelors of Science in Computer Science from Florida State University and a Master of Science in Industrial and Systems Engineering from the University of Florida.

Applying Service Orientation to the U.S. Army's Common Training Instrumentation Architecture

Jeremy T. Lanman
U.S. Army PEO STRI
Orlando, Florida
jeremy.lanman@us.army.mil

Scott Clarke, Shawn Hillis, R. Darbin
General Dynamics C4 Systems
Orlando, Florida
scott.clarke@gdc4s.com
shawn.hillis@gdc4s.com
rowland.darbin@gdc4s.com

Dave Frank
AIT Engineering
Orlando, Florida
david.frank@aitengineering.com

INTRODUCTION

The Common Training Instrumentation Architecture (CTIA) is the foundation architecture of the Live Training Transformation Family of Training Systems (LT2-FTS). From its inception, the goals of CTIA have been to reduce development costs, sustainment costs and fielding time of Army live training ranges. Goals have been met by a core focus on interoperability and fostering an environment that maximizes reuse (Dumanoir and Rivera, 2005). CTIA has been hugely successful and is a large driving factor in PEO-STRI's ability to obtain return-on-investment in excess of US$300 Million dollars for LT2 (Lanman and Kemper, 2011).

Army live training's focus has remained relatively constant; units conduct training which must be planned, monitored and evaluated. The technology used to perform these activities must evolve with the state-of-the-art in order to meet evolving user expectations. Users are increasingly expecting web interfaces and wireless mobile devices. These changes and the need to move toward different form factors and deployment strategies are reflected in the enhancement requests CTIA receives from product teams. In the course of executing these architecture change requests, it has become clear that CTIA is reaching the limitations of its middleware architecture. A main contributing factor for this limitation is that previous design strategies did not account for the types of technological evolutions that we are currently experiencing. To overcome these limitations CTIA must move to best-of-breed technologies and embrace design philosophies that maximize technology use. This new design must also maintain legacy support to protect the significant investment already made in the LT2 community.

The first task in any significant design effort is to ensure that the team has a clear understanding of the customer's end goals. Using Human Centered Design (HCD) techniques and a Goal Question Metric approach, the business vision of the customer was evaluated against potential technological solutions. Further analysis was performed to ensure that the problem domain was accurately captured and that user expectations will be met. The results of this analysis revealed that the SOA design philosophy (Erl, 2005) aligns very well with LT2 customer goals as well as LT2 product line development. An objective architecture was designed using services to provide the core capabilities needed in the live training domain. Orchestration of these services enables dynamic construction to support evolving LT2 use cases while remaining technologically agnostic to the greatest extent possible.

This paper describes the process that the CTIA Architecture team has undertaken to understand the future needs of the live training community and how SOA was applied to meet those needs. It addresses the unique considerations for performance, and defines the level of service granularity that is appropriate for Army live training.

BUSINESS VISION

The first step towards an architecture change for CTIA was to ensure that the design philosophy selected meets the business vision of the customer. Designing the system from the ground up to meet the primary objectives of the customer instead of technical concerns meant a greater alignment with customer needs and a higher probability of early success.

Stakeholders

To start an assessment of the business goals and objectives, it was first necessary to identify the various stakeholders who have an interest in the system. Each stakeholder was then evaluated for the amount of interest and influence they have in the specification of the system.

Table 1. Stakeholders Taxonomy

Stakeholder Category	Description
Capability Developer	Defines requirements, provides funding and overall direction
Material Developer	Designs, implements, and sustains architecture and components
Exercise Executor (Users of the system)	Interested in capabilities, performance, reliability, and usability; influence limited to ability to report problems and request enhancements
Integrator	Interested in aspects that affect integration and testing, influence
External System	Directly drives interoperability requirements
Training Unit	Not interested unless it impacts training
External Influences	No interest, influence only through changes in external software system dependencies

Table 1 above lists the category of stakeholders and specific members of each category. Figure 1 then graphs the interest and influence levels of each stakeholder category.

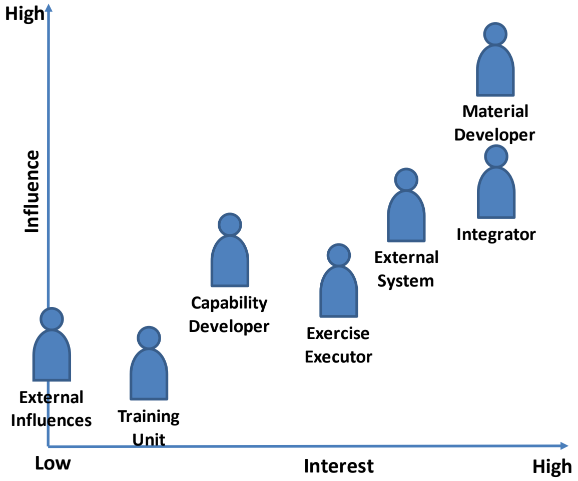


Figure 1. Stakeholder Interest and Influence

Business Goals and Objectives

The Architecture team, comprised of key stakeholders based on influence and interest, defined and prioritized the strategic business goals and objectives for the Live Training Transformation (LT2) architecture. CTIA provides the foundation for this architecture however business goals and objectives were extended to the LT2 community at large to ensure that the CTIA architecture aligns with community needs. These goals were then used to determine the priorities for the technology insertion effort.

The business goals were derived using the Goal-Question-Metric (GQM) paradigm (Basili, 2002). GQM is used to define measurement such that:

- resulting metrics are tailored to the LT2 organization and its goals
- resulting measurement data play a constructive and instructive role within the LT2 Product Line
- metrics and their interpretation reflect the values and the viewpoints of the different stakeholders affected

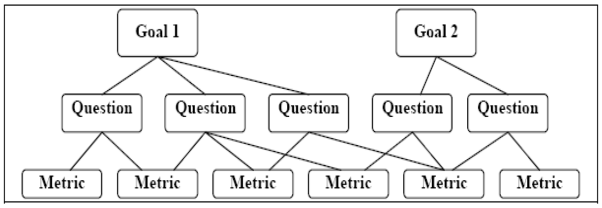


Figure 2. GQM Model

A GQM model is a hierarchical structure (Figure 2) starting with a goal (specifying purpose of measurement, object to be measured, issue to be measured, and viewpoint from which the measure is taken). The goal is refined into several questions that usually break down the issue into its major components. Each question is then refined into metrics, some of them objective such as the one in the example, some of them subjective. The same metric can be used in order to answer different questions under the same goal. Several GQM models can also have questions and metrics in common, making sure that, when the measure is actually taken, the different viewpoints are taken into account correctly (i.e., the metric might have different values when taken from different viewpoints). For each of the measurement areas that follow, the GQM process is documented and the details of recording and reporting the resulting metrics are described.

The Architecture team created high-level goals and used the GQM methodology to break down each goal into questions about the goal and finally drawing metrics that help to validate that the questions are answered and therefore, how well the goals are being met. The results of this analysis are shown in Table 2:

Table 2. Goal Prioritization

Goal	Description
Reduce Operational Costs and Complexity	Focused on reducing operational costs and complexity.
Align and support specific product development	Focused on using current live training development efforts to advance the product line by meeting core product needs and encouraging contributions back to the product line.
Enable Enhanced Soldier Training Effectiveness	Focused on increasing the training effectiveness of the LT2 products.
Reduce Development and Sustainment Costs	Focused on reducing the costs associating with building and sustaining LT2 products.
Increase Technology Agility	Focused on maximizing the ability of the architecture to incorporate new technology.
Leverage other Army Systems	Focused on providing ability for the LT2 product line to leverage systems and services that are developed by others (i.e. virtual, constructive, mission command or other live training systems).

After the ranking was complete, it became clear what the top priorities were. First, enable new training efficiencies and capabilities, giving trainers the ability to gather more data for more effective after action reviews. One of the key capabilities envisioned here is to support mobile devices. Second, increase support for distributed exercises, allowing for shared, centralized data centers and potentially even shared support staff and analysts. Third, lower product sustainment costs, decreasing the time and effort related to fixing issues after deployment. Then, improve alignment with current product development efforts, becoming more responsive to changing product requirements. Finally, improve support for individual product development, demonstrating value in the architecture and enabling and encouraging reuse across the product line.

Constraints

As the business goals and architectural objectives were analyzed, the following constraints were used to perform a tradeoff analysis of desired capabilities versus costs, impacts and requirements from the product line.

Protect Existing Investment (cost avoidance)
A significant investment has already been made in CTIA and LT2 components. The business goals and architectural objectives must retain at least some level of backwards compatibility to allow the product line to recoup a return on investment over approximately the next 5 years. Backwards compatibility may be achieved by implementing any combination of a number of different techniques, including compile-time (application programmer interface), run-time (network protocols) or design-time (data models) backwards compatibility. Although run-time backwards compatibility may provide the least impact to fielded systems, it might incur the greatest cost in implementing. In the past, the product line has strived to achieve at least some level of compile-time backwards compatibility since it provides a relatively equal balance of cost versus benefit.

Alignment with Army Standards
The U.S. Army's Chief Information Officer (CIO) G-6 has issued directives to implement a Common Operating Environment (COE) in order to support cloud computing and virtualization on the Army's Global Network Enterprise Construct (GNEC). As we evolve the LT2 product line to identify and achieve business goals for the live training community, we must also keep in mind the broader business goals and technology objectives for the Army.

Information Assurance
When the CTIA and LT2 architectures, change, components must consider the security and accreditation impacts in order to support the information assurance policies and processes. We must also consider that the Army will evolve to implement cloud computing and virtualization and that security and information assurance requirements are likely to evolve and introduce new requirements.

Product Schedules
As we identify courses of action and architectural changes, we must consider that impacts that negatively affect LT2 product schedules will not be well received. In order to ensure that the changes that are necessary for the architecture to evolve do not impact existing schedules, activities will need to focus on adding new capabilities instead of changing or removing existing capabilities. This constraint is also related to the "Protect Existing Investment (cost avoidance)" constraint, in that maintaining some level of backwards compatibility is one of the techniques for achieving this goal as well.

ARCHITECTURE VISION

Based on this analysis, it became clear that a Service Oriented Architecture (SOA) was likely a good approach to meeting the goals.

The objective architecture will be designed using SOA principles and web services with a shift from thick to thin clients (Lanman and Horvath, 2011). User applications will be able to be hosted in a web browser or run as “apps” on mobile devices and tablets. By making this switch, the architecture will become more flexible, scalable and simpler to operate. These changes also support the goal of hosting some or all of a range’s services at a Regional Training Center (RTC), which could support multiple ranges from a common data center.

The objective architecture will not require any of the legacy CTIA services to operate. However, wherever possible and necessary, backwards compatibility will be maintained with legacy software components through the existing CTIA Framework interfaces and a small set of services to support legacy operations. Additionally, the existing data-model and requirements will be reused, but improved based on lessons learned. The objective architecture will no longer have private, hidden interfaces and current services, such as gateways with public interfaces that allow other components to query data or send commands, will become first class services.

In addition to providing the set of core services, CTIA will also provide the service hosting software infrastructure. This underlying support software will host both CTIA services and product services. This includes a common message bus, application hosting, application monitoring, web-based user interface services and virtualization.

OSIMM ANALYSIS

With the identification of SOA design principles (Erl, 2008) as a major element of meeting the objectives of the architecture, the Architecture team needed to assess the maturity of the current CTIA architecture and processes against the principles as well as identify the desired future state. The Open Group Service Integration Maturity Model (OSIMM) was selected to complete this assessment. In addition to providing a way to measure the maturity of an organization and its information technology systems and business applications, it provides guidance on how to achieve the specific levels of service maturity necessary to realize related business benefits. The model defines seven dimensions and seven levels of maturity within each of those dimensions. The dimensions are: Business View, Governance and Organization, Methods, Applications, Architecture, Information and Infrastructure and Management. A system’s architecture may be at varying maturity levels across the dimensions (The Open Group, 2009).

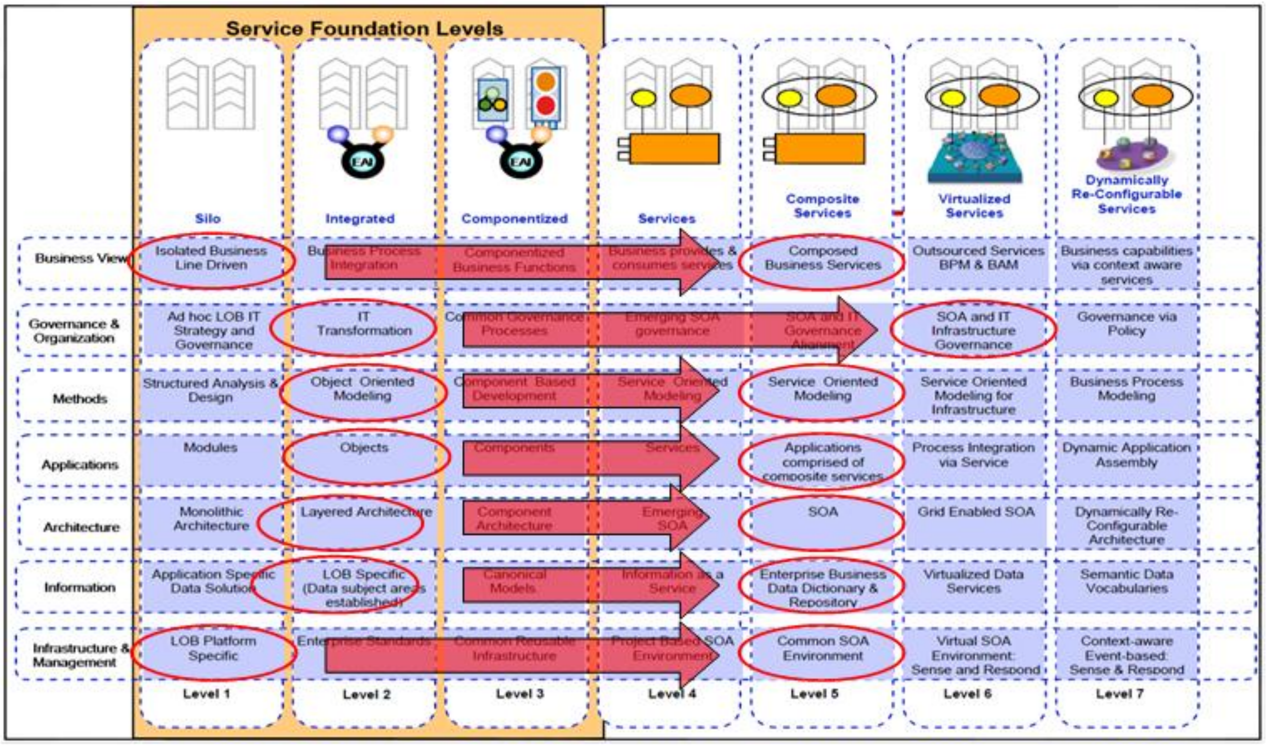
Current OSIMM Maturity

The analysis revealed that CTIA was within the Silo and Integrated levels of the seven dimensions; the two lowest levels. Silo is categorized by little cooperation between organizations within an enterprise and a high difficulty of integration between systems. Integrated improves on both of these factors. However, common standards for data and common business processes still haven’t been achieved. Integration still requires complex conversions of data, protocols and operations. Therefore it is still complex and costly to change or add new business processes.

These low ratings reflect several issues with not only the architecture but also the complexities of procurement and product requirements. The architecture and LT2 were found to be lacking in several key areas: formal infrastructure guidelines and recommendations, quality and performance metrics collection at the product level, a modern SOA implementation, and a capability to perform service monitoring.

Objective OSIMM Maturity

Much of the improvement comes from formalizing processes, formalizing goals, governance, moving to a modern SOA, and providing common infrastructure components. Figure 3 illustrates the objective OSIMM maturity levels for each dimension. It should be noted that the highest levels were not selected in any of the dimensions. As the levels increase, it’s believed that the benefits start to shrink when applied to the live training product line while the cost continues to increase.



	Silo	Integrated	Componentized	Services	Composite Services	Virtualized Services	Dynamically Re-Configurable Services
Business View	Isolated Business Line Driven	Business Process Integration	Componentized Business Functions	Business provides & consumes services	Composed Business Services	Outsourced Services BPM & BAM	Business capabilities via context aware services
Governance & Organization	Ad hoc LOB IT Strategy and Governance	IT Transformation	Common Governance Processes	Emerging SOA governance	SOA and IT Governance Alignment	SOA and IT Infrastructure Governance	Governance via Policy
Methods	Structured Analysis & Design	Object Oriented Modeling	Component Based Development	Service Oriented Modeling	Service Oriented Modeling	Service Oriented Modeling for Infrastructure	Business Process Modeling
Applications	Modules	Objects	Components	Services	Applications comprised of composite services	Process Integration via Service	Dynamic Application Assembly
Architecture	Monolithic Architecture	Layered Architecture	Component Architecture	Emerging SOA	SOA	Grid Enabled SOA	Dynamically Re-Configurable Architecture
Information	Application Specific Data Solution	LOB Specific (Data subject areas established)	Canonical Models	Information as a Service	Enterprise Business Data Dictionary & Repository	Virtualized Data Services	Semantic Data Vocabularies
Infrastructure & Management	LOB Platform Specific	Enterprise Standards	Common Reusable Infrastructure	Project Based SOA Environment	Common SOA Environment	Virtual SOA Environment: Sense and Respond	Context-aware Event-based: Sense & Respond
	Level 1	Level 2	Level 3	Level 4	Level 5	Level 6	Level 7

Figure 3. Objective OSIMM Maturity Roadmap

OBJECTIVE ARCHITECTURE

With a clear understanding of how to apply the SOA design principles (Erl, 2008) to meet the business vision. The next step is to apply these design principles to a technology that addresses the technical demands of the system.

Conceptual View

In the current state, each installation of CTIA has dedicated infrastructure ranging from server racks full of equipment to installation on a laptop. In the future, the Army would like to embrace cloud computing through the development of a regionalized and distributed training capability that provides the hardware and software at central locations. This relieves the ranges and units being trained from having to operate and maintain their own service infrastructure. While this may not be entirely practical for the large Combat Training Centers (CTC), many of the smaller training ranges, including homestation training, could leverage this model. Additionally, there may be a blend of local and central resources where the majority of the infrastructure is hosted at central locations with range assets supplementing it where performance, security, or other restrictions require it.

Going forward CTIA must support the mobile computing world. It must enable trainers to use mobile devices to capture training observations and evidence just like one might use an app to post a picture to a social networking site.

Figure 4 illustrates the conceptual view of this capability.

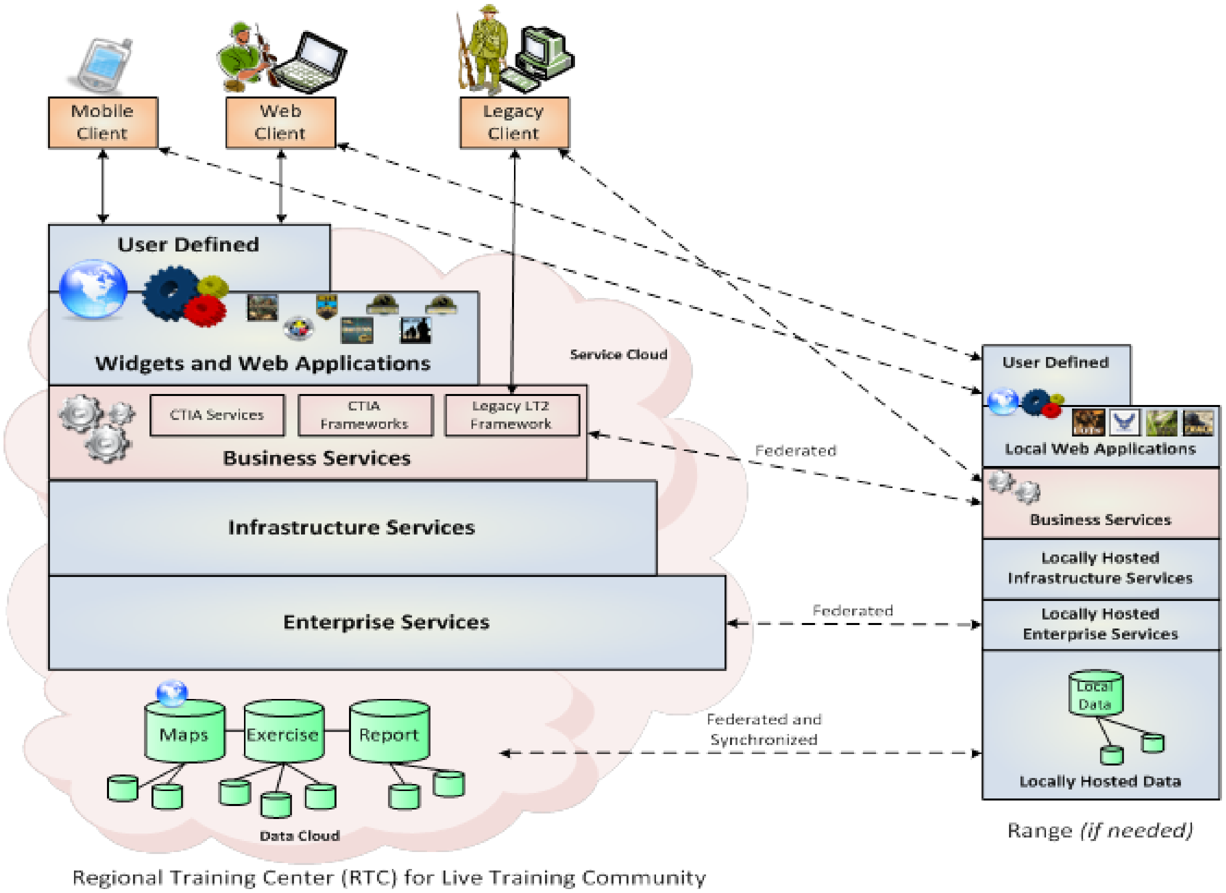


Figure 4. Architecture Conceptual View

While CTIA is moving towards support for Training as a Service (TaaS)—distributed and web-based training—the Department of the Army and DOD is also moving in this direction and is providing new guidance on cloud computing, SOA, and mobile strategy development. The Joint Command and Control (JC2) Objective Architecture is defining a reference architecture for the C2 community and a cloud computing environment that would host their services. At the same time, the CIO/G-6 Common Operating Environment (COE) Architecture guidance is doing the same for the Army Enterprise Network. The SOA-based objective CTIA is consistent with the JC2 Objective Architecture software view in that it embraces proper SOA concepts and architectural tenets. The mapping of layers is straight forward and provides a necessary logical connection between the objective CTIA and JC2 Objective Architecture so that the CTIA architectural concept can easily be described.

Logical View

One of the biggest initial challenges with applying SOA principles to an existing architecture is changing the way one thinks about the problem. CTIA is already composed of segregated services; a natural inclination is to wrap these with web interfaces. To avoid this pitfall and help ensure a clean bottom up approach was taken, the Architecture team applied Service Oriented Analysis techniques (Erl, 2008) to identify the core functions of the system. First, the basic training system business processes were modeled with simple flow charts. This resulted in a loose set of functional capabilities that must be met. Those capabilities then were assigned to contexts, which in turn were iteratively refined until a set of service candidates were identified around those contexts. This iterative process takes into account applying the SOA design principles, performance, the deployment environment, and other factors to create agnostic services that are reusable and composable.

As shown in Figure 5, services are logically grouped into three layers. The utility layer includes non-business related services that support higher level services such as logging. Entity services model the real world business entities and provide the set of operations on those entities. Task services perform business processes that can be composed of multiple other services.

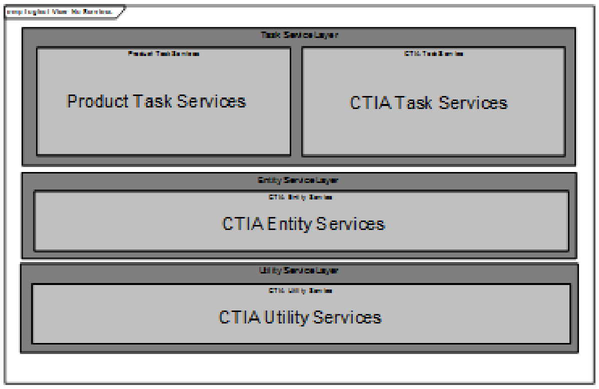


Figure 5. SOA Services Layers

The key break through for those involved in this process came while defining the entity services. Initially, the Architecture team fell into the trap of just wrapping SOA around the old architecture design. In the legacy architecture, player unit instrumentation, targets, and cameras, for example, were modeled by the same service that modeled platforms and people. Additionally, gateways supported conversion from these devices to CTIA and software components. However, platforms and people are very different from real world devices. Instrumentation, targets, and cameras are real world devices that are controlled, have communications intricacies, have status, etc. The team naturally started along the same path as the legacy architecture, until service oriented analysis was really taken to heart. Business entities then started falling out as their own services, with operations specific to their context. This provided clean, well defined interfaces to a set of reusable services for not only modeling real world devices but also interfacing with them.

Governance

LT2 is managed as a product line that maintains all LT2 core assets. First generation CTIA is viewed as a single core asset in the LT2 product line. It is released, updated, and adopted by product teams as a single dependency despite consuming approximately half of the average product's code base.

The principles of SOA governance for managing individual services align very well with the philosophy of managing a product line's core assets (Lanman and Proctor, 2009). Current component agreements are analogous to service agreements. Change management for services is conducted using the same Core Asset Change Proposal (CACP) process ensuring that all product teams have insight into and influence over their development and evolution. Existing automated testing managed by components can be applied in the same way to services. Even the design-time service inventory for SOA architectures is present in the LT2 product line. LT2 software components are inventoried on the LT2 Portal with version history component agreements and all of their associated requirements.

Within the LT2 product line individual services can now be treated as single point core assets that can be released and managed independently. This means that product teams can consume them as they do other core assets. Releasing individual services reduces the overhead associated with integration of an entire architecture thereby encouraging product teams to adopt early and often.

TRANSITION ARCHITECTURES

To support project, business, and product line goals, the services and capabilities of the *objective architecture* were allocated into five *transition architectures*. Each transition architecture was based on a specific use case, where each subsequent use case increases in scale, capability, and complexity from the one preceding it. Considerations for transition architectures were also constrained by project funding. The objective SOA services and capabilities were then allocated to the transition architectures. Services allocated to each transition architecture will enable progressive levels of product team adoption. Product teams will be able to orchestrate the architecture services to meet their intended training use case, and develop user level application user interfaces. Finally, each transition architecture will support integration with first generation CTIA to the extent of the services provided.

Transition Architecture 1

Transition architecture 1 provides the Service Oriented Infrastructure (SOI), basic entity creation and tracking for the purpose of exercising the architecture, and the core services that all future transition architectures will build from. This will provide a demonstrable capability for the first year, including a prototype 2-dimensional map visualization application and participant definition tool, orchestrated by task, entity, and utility service compositions.

Transition Architecture 2

Transition architecture 2 expands on transition architecture 1 to provide basic unit instrumentation and tracking. This will provide enough capability for product teams to adopt the architecture and compose the provided services to implement systems for land navigation with database persistence.

Transition Architecture 3

Transition architecture 3 adds services to support force-on-target engagements. Services will be available for product teams to implement instrumented ranges with fixed targets and support mobile devices.

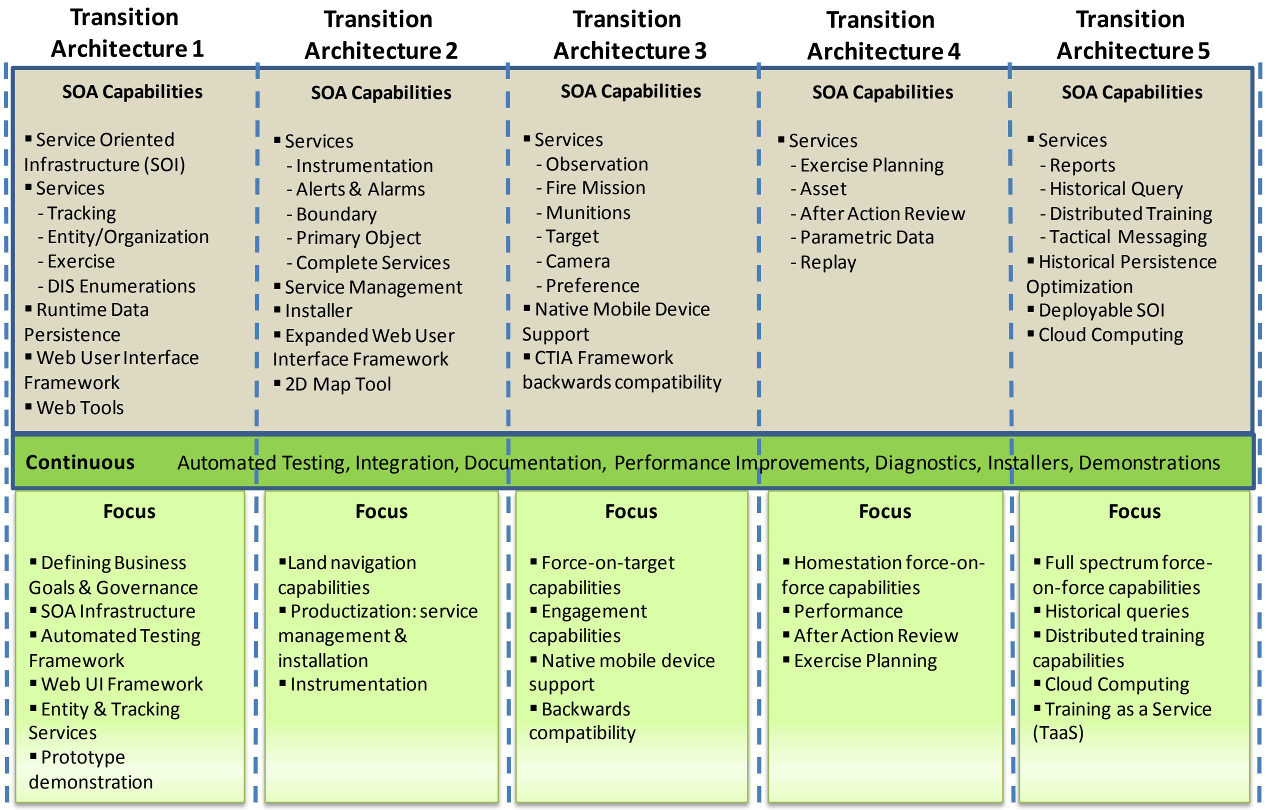
Transition Architecture 4

Transition architecture 4 provides services to support basic force-on-force instrumentation for brigade level homestation training with constructive data feeds and battle damage assessment. Services will include asset tracking and exercise replay.

Transition Architecture 5

Transition architecture 5 is the final instantiation of the objective architecture. Cloud-based with a deployable SOI supporting the full live training domain. Exercises will support up to battalion level force-on-force exercises integrating with mission command systems and include full wrap-around live, virtual, and constructive interoperability capability.

Figure 6 below illustrates the roadmap of the Transition Architectures and the SOA capabilities associated with each.



Transition Architecture 1	Transition Architecture 2	Transition Architecture 3	Transition Architecture 4	Transition Architecture 5
SOA Capabilities ▪ Service Oriented Infrastructure (SOI) ▪ Services - Tracking - Entity/Organization - Exercise - DIS Enumerations ▪ Runtime Data Persistence ▪ Web User Interface Framework ▪ Web Tools	**SOA Capabilities** ▪ Services - Instrumentation - Alerts & Alarms - Boundary - Primary Object - Complete Services ▪ Service Management ▪ Installer ▪ Expanded Web User Interface Framework ▪ 2D Map Tool	**SOA Capabilities** ▪ Services - Observation - Fire Mission - Munitions - Target - Camera - Preference ▪ Native Mobile Device Support ▪ CTIA Framework backwards compatibility	**SOA Capabilities** ▪ Services - Exercise Planning - Asset - After Action Review - Parametric Data - Replay	**SOA Capabilities** ▪ Services - Reports - Historical Query - Distributed Training - Tactical Messaging ▪ Historical Persistence Optimization ▪ Deployable SOI ▪ Cloud Computing
Continuous Automated Testing, Integration, Documentation, Performance Improvements, Diagnostics, Installers, Demonstrations				
Focus ▪ Defining Business Goals & Governance ▪ SOA Infrastructure ▪ Automated Testing Framework ▪ Web UI Framework ▪ Entity & Tracking Services ▪ Prototype demonstration	**Focus** ▪ Land navigation capabilities ▪ Productization: service management & installation ▪ Instrumentation	**Focus** ▪ Force-on-target capabilities ▪ Engagement capabilities ▪ Native mobile device support ▪ Backwards compatibility	**Focus** ▪ Homestation force-on-force capabilities ▪ Performance ▪ After Action Review ▪ Exercise Planning	**Focus** ▪ Full spectrum force-on-force capabilities ▪ Historical queries ▪ Distributed training capabilities ▪ Cloud Computing ▪ Training as a Service (TaaS)

Figure 6. SOA Transition Architectures Roadmap

CONCLUSION

A strong focus on business goals and objectives and a thorough understanding of the technical needs of the community is critical to success. Technology obsolescence and evolving user expectations toward mobile computing and web-based user interfaces along with the migration of LT2 product teams toward more distributed computing platforms, exposed the limitations of the first generation CTIA. Realizing these limitations the Architecture team along with the stakeholders evaluated the applicability of migrating towards a services-based architecture.

The team was able to emphasize that SOA is a design strategy (Erl, 2005) not an implementation. Design principles of SOA do not mandate web interfaces to services or communication over slow, high bandwidth web-based protocols. By using SOA design principles (Erl, 2008) and establishing a repository of well defined atomic services instead of plug-in modules in a fixed framework, the new LT2 Architecture will leverage legacy systems to protect the significant investment already made by LT2 while at the same time make the necessary step up to current state-of-the-art technology. SOA will enable product teams to reduce operational cost and complexity though greater reuse and finer grained selection of services to meet the user's specific needs. Aligning and supporting specific product development will be accomplished by developing services in a manner which provides live training domain level capabilities that can be composed into user interfaces to meet tailored needs. Service composition eliminates the overhead associated with thick, monolithic software applications containing copious features that are only used by some use cases; the result is more productive users and enhanced solder training effectiveness.

The CTIA technology insertion effort has resulted in a clearer understanding of the business vision and a better alignment of technology to meet the business goals and objectives of the Live Training community. Breaking the previous design philosophy and embracing SOA design principles will enable the next generation of product teams to leverage state-of-the-art technologies to meet user needs and realize the Army's vision for a Common Operating Environment, regionalized and distributed training capability, and web-based mobile devices.

ACKNOWLEDGEMENTS

The authors wish to thank Col. Mike Flanagan, Project Manager Training Devices (PM TRADE), Mr. Brian Kemper, PEO STRI Deputy Director of Engineering, and Mr. Jim Grosse, APM TRADE Chief Engineer, for their vision, contributions, and continued support in pursuing the methodologies and technologies described in this paper.

REFERENCES

Basili, V.R., et al. (2002). The Goal Question Metric Paradigm. In: Marchiniak J. J (ed.): *Encyclopedia of Software Engineering*. New York, pp. 578-583.

Dumanoir, P., Rivera, J. (2005). *Live Training Transformation (LT2)-A Strategy for Future Army and Joint Live Training*. 2005 Interservice/ Industry Training, Simulation, and Education Conference (I/ITSEC), Orlando, FL.

Erl, T. (2008). *SOA Principles of Service Design*, Boston, MA: Prentice Hall.

Erl, T. (2005). *Service-Oriented Architecture: Concepts, Technology, and Design*, Upper Saddle River, NJ: Prentice Hall.

Lanman, J.T.., Kemper, B.E., et al. (2011). *Employing the Second Generation Software Product-line for Live Training Transformation*. 2011 Interservice/Industry Training, Simulation, and Education Conference (I/ITSEC), Orlando, FL.

Lanman, J.T., Horvath, S.D., and Linos, P.K. (2011). *Next Generation of Distributed Training utilizing SOA, Cloud Computing, and Virtualization*. 2011 Interservice/Industry Training, Simulation, and Education Conference (I/ITSEC), Orlando, FL.

Lanman, J.T., Proctor, M.D. (2009). Governance of Data Initialization for Service Oriented Architecture-based Military Simulation and Command & Control Federations. *Journal of Defense Modeling and Simulation: Application, Methodology, Technology*, Vol. 6, No. 1, 5-16.

The Open Group, (2009). *The Open Group Service Integration Maturity Model Technical Standard (Version 2)*. Retrieved August 2011, from https://collaboration.opengroup.org/projects/osimm/