FOUNDATION INITIATIVE 2010: THE FOUNDATION FOR RANGE INTEROPERABILITY

George J. Rumford
U.S. Army White Sands Missile Range
george.rumford@osd.mil

Minh Vuong
U.S. Army Simulation, Training, and Instrumentation Command
minh_vuong@stricom.army.mil

Stephen T. Bachinsky
Science Applications International Corporation
stephen.t.bachinsky@saic.com

Dr. Edward T. Powell
Science Applications International Corporation
edward.powell@saic.com

ABSTRACT

Foundation Initiative 2010 (FI 2010) is a joint interoperability initiative of the Director, Operational Test and Evaluation. The vision of FI 2010 is to enable interoperability among ranges, facilities, and simulations in a timely and cost-efficient manner and to foster reuse of range assets and future range system developments. To achieve this vision, FI 2010 is developing and validating a common architecture with a common range object model, a core set of tools, inter-range communication capabilities, interfaces to existing range assets, interfaces to weapon systems, and recommended procedures for conducting synthetic test events and training exercises. During FY 01, the project is developing the second Test and Training ENabling Architecture (TENA) Middleware Prototype as a basis for range communication. FI 2010 will advance a simulation-based acquisition or a 'distributed engineering plant' methodology to streamline weapon system acquisition. Benefits from the FI 2010 products include cost effective replacement of customized data links, enhanced exchange of mission data, organic TENA-compliant capabilities at test sites to be leveraged for future test events, and instrumentation system reuse. Through FI 2010, future inter-range operations, instrumentation development, and range capability sustainment will cost less and incur less risk.

KEY WORDS

Interoperability, Reuse, Architecture, Instrumentation, Ranges, Test and Evaluation, TENA, SBA, JDEP

INTRODUCTION

Over the coming decades the demands on the U.S. Military will be transformed by the evolution of America's security and economic interests coupled with the continuing revolution in information technologies. Joint Vision 2020 (JV 2020), an evolutionary enhancement to the original Joint Vision 2010, provides a roadmap for, over the coming decades, "creation of a force that is dominant across the full spectrum of military operations — persuasive in peace, decisive in war, pre-eminent in any form of conflict." [1]

As the demands on the military are being transformed by information technologies so are its acquisition processes. Simulation Based Acquisition (SBA) has the goal of substantially reducing the time, resources, and risk associated with acquisition, while producing higher quality products through adoption of a "model-simulate-fix-test-iterate approach" to acquisition.

These transformations coalesce directly within the Test and Evaluation (T&E) community. Test and training ranges must support implementation of the JV 2020 vision through joint testing, must serve as an enabler for SBA, must foster integration of testing and training, and must do so in a way that minimizes future costs of range operations. To accomplish these goals, ranges must become more integrated and interoperable, allowing range assets to be procured and used efficiently, to be reused effectively and to be combined to create the scale and scope of capabilities required to meet the challenges of implementing JV 2020. The Foundation Initiative 2010 (FI 2010) project is the range community's mechanism for creating the infrastructure needed to meet these challenges.

PROJECT DESCRIPTION

The FI 2010 solution includes a suite of products that will enable affordable interoperability between test ranges and facilities as a means of sharing resources and data, thus enhancing the capabilities of the integrated test ranges and facilities; and provide a common architecture that will enable improvements and modernization efforts made by one range to be incorporated by all cooperating ranges. The FI 2010 project is developing and validating a common architecture with a common object model, a core set of tools, inter-range communication capabilities, interfaces to existing range assets, and interfaces to weapon systems, along with recommended procedures for conducting synthetic / multi-range test events or training exercises.

FI 2010 Common Architecture. The FI 2010 common architecture, referred to as the Test and Training Enabling Architecture (TENA) (see Figure 1, below), while compatible with the Department of Defense (DoD) High Level Architecture (HLA) for Modeling and Simulation, addresses test requirements beyond those supported by HLA and drastically improves the ability of ranges to interact with simulations. The FI 2010 project is coordinating with the Range Commanders Council (RCC) so that TENA objects, standards, and protocols are reviewed and adopted by the range community as a whole. The TENA architecture enhances software interoperability and reuse throughout the Range Community by giving guidance on how to design range software applications such that they can easily interact with other TENA applications to support a range event. The architecture also specifies a common TENA Object Model, akin to a common set of interface definitions, which enables this interoperability through a community-wide understanding of range-related information. The primary benefit of the TENA architecture is the ability it gives exercise planners to rapidly compose TENA-compliant range capabilities into a geographically distributed "Logical Range" (sometimes also called a virtual range) for a specific test mission or training exercise.

TENA applications (also called "TENA Resources"), tools, repositories, and gateways use the TENA Middleware as their means of communication. A copy of the TENA Middleware software is linked into every TENA application and is the mechanism for all execution-time communication between TENA applications.

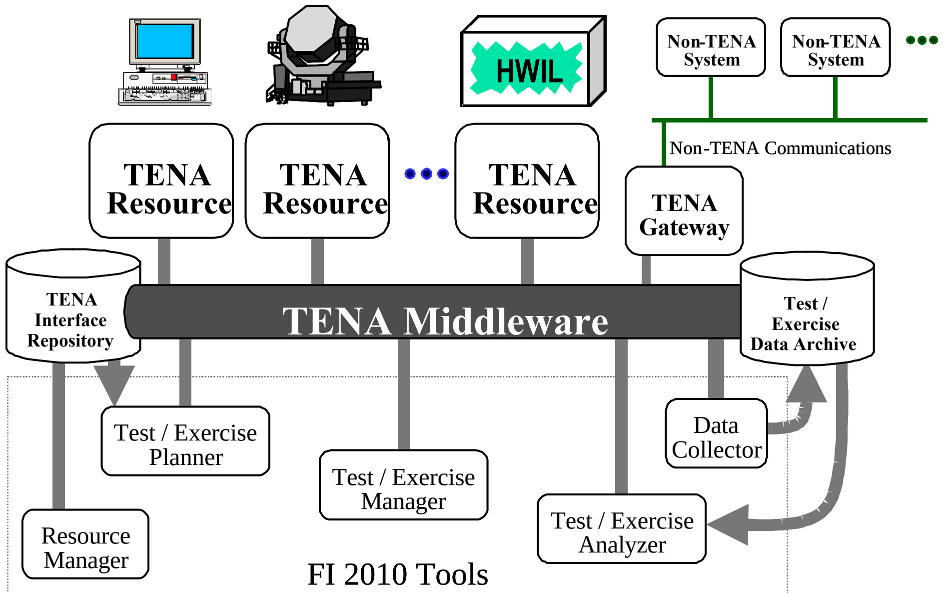

Figure 1: The TENA Architectural Construct

The FI 2010 Tools are a suite of software applications to assist range engineers in planning, configuring, controlling, and analyzing geographically-distributed, simulation-enhanced range missions. These tools will enhance the productivity of the range engineer so that planning and reconfiguring large-scale synthetic exercises can be accomplished much faster and with higher reliability.

TENA inter-range communication capabilities will function over both organic DoD wide-area networks and commercial communication services, using the necessary end equipment and encryption devices required for transferring large quantities of data between geographically dispersed locations in a TENA-compliant exercise.

The FI 2010 project is developing the software infrastructure necessary for existing range resources to become TENA-compliant. The common infrastructure will cost-effectively enable current range tools to be adapted to this common architecture, rather than require that replacement systems be developed. Allowing weapon systems under test to be stimulated with a simulation, the FI 2010 project is also defining TENA interfaces for tactical systems, such as C4I systems. Finally, the FI 2010 project is documenting recommended procedures for test missions involving simulations or across multiple ranges as well as methods for making new instrumentation systems compliant with the common architecture.

Development Strategy. To ensure that the TENA architecture satisfies the DoD ranges' needs, it is being developed with significant involvement from the major DoD ranges. Development Test Cells (DTCs) have been established at representative ranges to provide early insight to TENA performance so that the ranges can, in turn, give essential, detailed feedback on operational issues. Furthermore, system engineers from across the range community participate in the Common Test and Training Range Architecture (CTTRA) workshops to give recommendations to the TENA managers, architects, and developers. TENA is being defined and developed in three stages, each resulting in the creation of a significant software prototype. During FYs 99 and 00, the project developed and tested the first TENA Middleware prototype that both updated and validated a subset of the TENA Technical Reference Architecture. This software prototype was tested at six test centers across the three Services. During FY 01, the FI 2010 project is developing the second TENA Middleware prototype (codenamed IKE 2) with expanded functionality. Using a rapid prototyping approach, the development spirals of IKE 2 will allow test engineers at DoD ranges to quickly evaluate and refine the architecture, software design, and common object model. A third middleware prototype is planned for the FY 03 timeframe with the goals of optimizing performance, broadening platform support, expanding the common object model, increasing the complexity of logical ranges being tested, and enhancing support for the integration of legacy systems. The FI 2010 project is also currently reviewing several commercial-off-the-shelf (COTS) software applications to serve as FI 2010 tools. The project is on track and scheduled for completion by the fourth quarter of 2004, with the ultimate goal to have TENA adopted as an RCC standard.

IKE 2 TECHNICAL OBJECTIVES

The essential problem addressed by the creation of the second TENA Middleware prototype (IKE 2) is interoperability between and among range information processing systems leading to tighter integration, more software reuse, increased connectivity, and a better ability to create larger, more realistic, and functional logical ranges. The middleware will facilitate reuse through composable or modular solutions to range problems, accomplished primarily by providing a standards-based object-oriented Application Program Interface (API) for range software development.

The IKE 2 middleware will support full integration of dynamic range data through a combination of object-based and message-based data distribution mechanisms that are applicable across run-time, management, planning, and analysis applications. The approach will support high-performance, run-time data transfer via a real-time data distribution system derived from the Common Object Reference Broker Architecture (CORBA). The IKE 2 approach facilitates legacy integration through inclusion of "Gateway Applications" that bridge TENA and legacy protocols.

IKE 2 TECHNICAL APPROACH

The IKE 2 design draws heavily on object technology, employing object-oriented frameworks to provide flexibility, extensibility, and modularity to allow TENA developers to tailor the infrastructure to meet their particular needs.

System Design Philosophy. The IKE 2 middleware is designed to be as flexible as possible for use by the range software developer, with support for multiple programming languages and configurations, and with some IKE 2 modules capable of being tailored to meet a specific range event's unique requirements. The TENA vision calls for the IKE 2 middleware be designed to embrace change, promote flexibility, and manage complexity so that the range community can work together to build interoperable and reusable systems in as efficient and cost-effective a solution as possible. The goal is for the IKE 2 design to be robust enough for the most demanding applications and general purpose enough to meet the requirements of future range event configurations.

The use of object-oriented frameworks provides the technical capability to create a middleware that is both generic (applicable to any event) and tailorable (customizable to meet the special needs of any particular event) within a single software architecture and using the same interfaces. An object-oriented framework is a set of classes, designed to work together, that embodies an abstract design for solutions to a family of related problems. In the case of IKE 2, the family of related problems is the interoperability, reuse, and object distribution problems addressed by TENA. Developers specialize framework classes to solve a specific problem. Frameworks add structure to these technical solutions, and it is this structure that enhances flexibility and adaptability.

The basic approach in designing IKE 2 is to create a design that is general enough to meet the needs of the range community, interoperable enough to present a consistent API to all range systems, and flexible enough to adapt to the distinct needs of a given test mission.

CRITICAL ARCHITECTURAL ELEMENTS

The IKE 2 prototype design is based on a number of key architectural elements, assumptions, and technologies. This section discusses these elements.

Object Model Interaction Mechanisms. There are a number of object communication modes that need to be supported by TENA (and the IKE 2 prototype) and incorporated into the TENA object model. The two primary modes are:

- **Stateful Distributed Objects.** This communication mechanism represents a distributed object paradigm (such as the one used in CORBA) augmented to include dissemination and local caching of object state information. The distributed object aspect of this mechanism allows an object's methods to be accessed independently from where the actual server object resides. The client application manipulates a proxy object that "stands in" for the servant object. When the client application invokes a method on its proxy, the proxy invokes the method on the servant object (wherever it is) and pulls any return information from the servant and delivers it to the client. If the client application merely wishes to access the object's state information, accessor methods immediately return the values of the appropriate state variables that have been cached locally inside the proxy. The combination of CORBA-like location transparency with efficient state dissemination and caching is referred to as a "Stateful Distributed Object" paradigm.
- **Event Channel Messaging and Data Streams.** This communication mode is similar to more low-level message-based communication in that single messages and data streams can

be sent between applications based on their needs. Support for data streams is intended to provide TENA applications with native capability to efficiently send and receive audio, video, and telemetry without incurring undue encapsulation and transmission overhead.

Pre-compiled Object Model. An object model that is pre-compiled and linked into an application yields many benefits, including increased efficiency, the ability to support strong type checking, and the ability to efficiently handle complex object relationships. Since run-time typing of the object model can lead to difficult-to-find defects in the application — given that both type and value information are supplied by the logical range integrator and are not checked for consistency by the compiler — compile-time definition and typing is chosen to provide increased reliability as well as increased performance.

A mechanism is needed to enable a logical range integrator to create a compilable object model without impacting IKE's generality. The IKE 2 solution to this problem is to use auto-code generation based on defining the object model in a variant of the CORBA Interface Definition Language (IDL) called the TENA Definition Language (TDL). A TDL compiler will compile the object model for a given exercise or configuration into a programming language code such as C++ or Java. The compiler would auto-code-generate all the software necessary for the object model to be seamlessly linked into the application.

Inter-Architecture Integration Mechanism. TENA must have a mechanism to incorporate legacy range systems so as to leverage the existing DoD test and training infrastructure. Moreover, TENA must be able to interface to weapon systems to properly inject (or stimulate) and extract (or collect) data from the weapon systems under test. Modular, specialized gateway applications allow logical range integrators to build logical ranges with existing range resources and, more importantly, incorporate weapon systems into a logical range construct. There is nothing inherent in the gateway application concept to preclude the use of multiple gateways, or even federated gateways, in a given exercise or event.

Performance. The optimization of critical performance parameters is critical for certain test events. Minimal latency and jitter, combined with maximal throughput, are the goals of IKE 2. There are a number of things that can be done to create an IKE 2 prototype that will minimize latency caused by delivering data to, and receiving data from, the network. Incorporating the object model at compile time will yield significant gains, since the data that needs to be sent over the wire can be packaged very efficiently using pre-compiled code rather than code that needs to interpret each attribute before packaging. In addition, a great deal of efficiency is gained using intelligent memory management that minimizes data copying and context switching.

Security. Security is important in TENA since only certain authenticated users and applications will have the right to interact with certain objects. The CORBA Security Service, defined in the CORBA Services specification and implemented within the TAO Object Request Broker (ORB) (see below) will be evaluated as the distributed object security mechanism used by TENA. Since physical security for the computers and networks will be maintained based on DISA standards (e.g., separate networks, bulk and/or packet/cell encryption devices), the CORBA Security Service will provide additional object-based security within a trusted enclave. Thus a layered approach, starting with physical security and network security then adding object security on top, will ensure that the final systems and configurations meet the security needs of any given user community.

IKE 2 DESIGN

A high-level view of the IKE 2 design is shown in Figure 2, below. Each box represents a package that defines a segment of internal IKE functionality. A brief description of each major package follows.

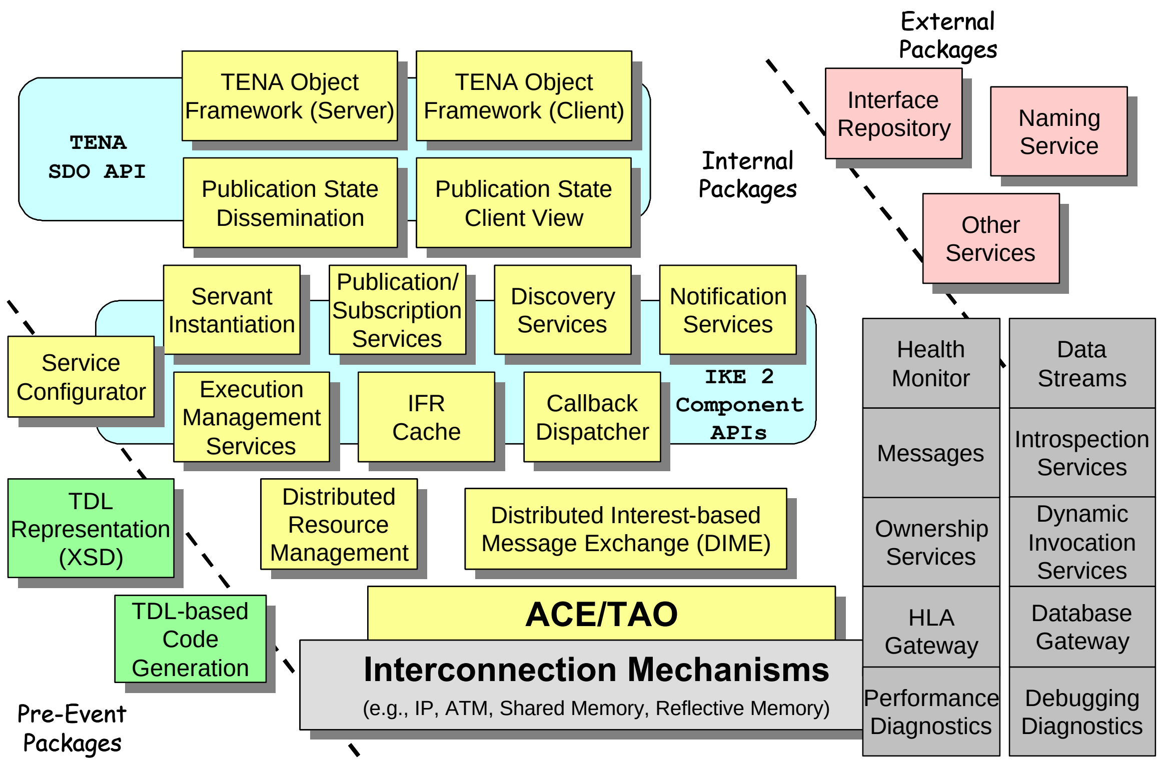

Figure 1 — Layered conceptual view of the IKE 2 internal software packages.

Broadly speaking, there are four categories of packages in the design. The first category consists of those packages required for pre-exercise activities, primarily defining the TENA Object Model. The XML-based TDL representation gives users access to many COTS XML editors to use in designing and documenting the TENA objects that will be used in a given event. The TDL-based code generator generates the C++ and Java classes (both header files and implementations) that implement the stateful distributed objects (SDOs), messages, and data streams used in the exercise, including all the supporting and ancillary classes that make IKE 2 work with the SDOs.

The second category of packages consists of those external software items that exist at run-time but are not linked with the user's application. The interface repository provides run-time access to the TDL object definitions to any application that wishes to participate in a TENA logical range but declines to compile the TENA object model directly into their application. For these "introspective" applications, the interface repository provides all the information they need to subscribe to SDOs, interpret SDO state when it is delivered, and dynamically invoke methods on SDOs without knowing their structure beforehand. The Naming Service provides a mechanism for SDOs to be given well-defined names in a logical range, allowing applications to find a particular SDO simply by knowing its assigned name or ID number.

The third category of packages are those in the center of Figure 2, the ones that make up the bulk of the IKE 2 software that is linked into every TENA application. IKE 2 is based on a foundation provided by the Adaptive Communication Environment (ACE) and The ACE ORB (TAO). ACE is a freely available, open-source, object-oriented framework that implements many core design patterns for concurrent communications. ACE provides a rich set of reusable C++ wrapper facades and framework components that perform common communication software tasks across a range of computer platforms. The communication software tasks provided by ACE include event demultiplexing and event handler dispatching, signal handling, service initialization, inter-process communication, shared memory management, message routing, dynamic (re)configuration of distributed services, concurrent execution and synchronization. TAO is a freely available, open-source, and standards-compliant real-time implementation of CORBA (including many CORBA Services) that provides efficient, predictable, and scalable quality of service (QoS) end-to-end.

Basic low-level publish and subscribe functionality used by all communication mechanisms is provided by the Distributed Interest-based Message Exchange (DIME) framework. DIME provides for the declaration of a subscriber's interest and the consequent delivery to subscribers of only that information they wish to receive.

Important additional functionality is provided in the packages that make up the IKE 2 Component APIs. These packages allow users to instantiate SDOs, forward publication and subscription information to DIME, discover existing SDOs that match outstanding interests and instantiate them, and notify the application (if requested) of changes in the state of SDOs that have already been discovered. A Callback Dispatcher package manages the facility for calling back the application in response to new events, based on an application-defined threading model. The Interface Repository (IFR) cache gives the middleware access to class definition information so the middleware can instantiate objects for which compiled-in factories do not exist. Execution Management Services allow applications to create, join, synchronize with, leave, and destroy logical range executions. Finally, a Service Configurator package manages the middleware's instantiation of services so that those that are not needed for a given logical range execution may be omitted with consequent decrease in middleware overhead.

The final collection of packages in the core IKE 2 software, comprising the TENA SDO API, consist of those classes needed to support the management of the SDOs themselves. TENA Object Frameworks (one for server objects and another for client objects) define the base classes from which all TENA objects and their proxies derive. Other classes are required for the management of a TENA object's publication state information, which can be created, updated, and read by the software application serving that object, as well as read by client applications.

The final category of packages being created for IKE 2 are those in the bottom right-hand corner of Figure 2, and represent functionality that will be designed, developed, and integrated in subsequent development spirals. The Health Monitor package provides the middleware and TENA applications insight into the status of all applications comprising the current logical range. This type of monitoring can provide crucial information to support fault tolerance in application and logical range design. The Message and Data Stream packages implement efficient transmission and reception capability for these important communication modes. Introspection Services and Dynamic Invocation Services provide simplified access to the Interface Repository (which in the current design is the CORBA Interface Repository), allowing applications to fully participate in a logical range without compiling in the TENA Object Model. Diagnostics provide the IKE 2 developers and the application developers the ability to measure IKE 2 performance, while also providing valuable information for debugging purposes. Finally, gateways are planned to provide an integration mechanism with HLA federations and Structured Query Language (SQL) databases. The "Database Gateway" represents a unique TENA-based logging and replay mechanism using standard interfaces to a wide range of commercial databases.

CONCLUSION

Future warfighting concepts, relying heavily on network-centric warfare capabilities, cannot be adequately tested with today's test resources. A new, interoperable foundation must be established. The nation cannot afford to rebuild the entire test and training infrastructure, so a strategy must be instituted to ensure that needed legacy capabilities are adapted, and future investments are designed, to be interoperable in support of these future warfighting scenarios. The FI 2010 project, with its IKE 2 prototype, is a clear step toward developing this strategy by investigating, developing, and coordinating the standards and procedures needed to achieve interoperability. Range asset owners and acquisition program managers are the key to successful implementation. The FI 2010 team is heavily involved in coordinating their activities among all potential stakeholders to ensure success.

REFERENCE

[1] Joint Vision 2020, U.S. Government Printing Office, Washington, DC, June 2000