# Evolving Enterprise Infrastructure for Model & Simulation-Based Testing of Net-Centric Systems

Steven Bridges[a]
Email: steve.bridges@disa.mil

Bernard P. Zeigler, Ph.D. [b]
Email: zeigler@ece.arizona.edu

James Nutaro, Ph.D. .[c]
Email: nutarojj@ornl.gov

Dane Hall,
Tom Callaway, and Dale Fulton[a]
Email: {dane.hall,tom.callaway,dale.fulton}ctr.@ disa.mil

a)Joint Interoperability Test Command, P.O. Box 12798, Huachuca, AZ 85670-2798,Telephone: 520-538-5482

b) Arizona Center for Integrative Modeling and Simulation,Electrical and Computer Engineering, University of Arizona,Tucson, AZ 85721, Telephone: 520-621-2108

c) Oak Ridge National Laboratory, P.O. Box 2008, Oak Ridge, TN 37831,
Telephone: 865.574.4160

## Introduction

This paper attempts to provide perspectives on how a test organization can organize and plan for enterprise-wide adoption of advances in emerging technologies and techniques, whether developed in-house or acquired from external sources. We discuss an overarching strategic plan for integrating existing test technologies, identifying enterprise-wide technology gaps, and coordinating the development and acquisition of new test capabilities. Given Department of Defense (DoD) mandates for transition to net-centric operation, a test organization must acquire the ability to perform large-scale and fast-paced developmental and operational testing of Global Information Grid/Service Oriented Architecture (GIG/SOA)-based development projects. For example, the Joint Interoperability Test Command has the responsibility to test for GIG/SOA compliance for such projects as Net-Centric Enterprise Services and Net-Enabled Command Capability. We enumerate capabilities that greatly enhance the test organization's ability to support the impending testing demands from such GIG/SOA-based projects. Although most test organizations already have these capabilities to some extent, we believe they could benefit from the proposed approach to greatly accelerate their readiness to meet impending net-centric testing challenges.

After enumerating the recommended testing capabilities, we recommend a plan with short-, medium-, and long-term horizon components to acquire or improve said capabilities, and offer a layered architecture that provides a framework for such capability acquisition. We also recommend that test organizations incentivize their contractors to exploit the Extensibility, Composability, and Reusability of technical attributes of SOA to support the development of the layered architecture. We also conclude that the design of the test organization instrumentation and automation on top of the GIG/SOA infrastructure should be based on a model-driven software approach, systems-engineering modeling, and simulation principles and frameworks.

## Net-Centric Test Capabilities

Several specific capabilities that a test organization must address to effectively conduct developmental and operational tests of net-centric systems are described below [BUC04] [CAR05]:

**Composability:** The capability to seamlessly compose the elements of the desired test environment by selecting and configuring live (e.g., human players, military systems) and/or virtual (digital representations of live components) versions of all test environment components. Test organizations can take advantage of the SOA and component styles that offer technical advantages for the composition of test instrumentation services and applications. Contractors should be incentivized to exploit the SOA constructs to build plug-and-play capabilities while meeting current and future needs.

**Reusability and Persistence:** The test infrastructure persists over time and includes organized repositories to support the reuse of such elements as simulation models/digital representations, test development and implementation processes, and test experimentation components and tools (intelligent test agents, for example). This includes the capability to automatically store, catalogue, and retrieve all information produced by any node on the network in a comprehensive, standard repository. A critical advantage of such repositories for the test organization is that they also help to avoid duplication of efforts by the test organization’s multiple contractors.

**Extensibility**: The test infrastructure can be efficiently extended through the use of common architecture, interfaces, processes, and tools. Extensibility, Composability, and Reusability are mutually supportive attributes of model-driven software design methodology informed by engineering modeling and simulation fundamentals. The test organization must incentivize contractors to adopt such methodologies to achieve Extensibility, Composability, and Reusability attributes in its developments.

**Instrumented Trustworthy Measurement:** The ability to instrument test environments in a manner that is principally non-intrusive and highly embedded, which provides real-time measures at the system and system-of-system levels. Measurement is consistent and repeatable across experimental replications, providing reliable and trustworthy data. Specifically, instrumented trustworthy measurement includes the:

- Capability to reproduce the test environment and play back segments of the test event in a manner that facilitates assessing the effects of modifying the experimental conditions with plug-and-play replaceable test components.
- Capability to measure, compare, and evaluate experimentally-specified architectural and parametric configurations of the system under test.
- Capability to collect and segregate operational data (e.g. tactical and strategic data exchanged between systems under test) from test support data (e.g. instrumentation, simulation, analysis, and test control data).
- Capability to seamlessly switch between real-time and after-test analysis of collected data.
- Capability to perform the testing of Net Ready Key Performance Parameters (NR-KPP) and compliance to the Net-Centric Reference Model for upcoming GIG/SOA and other net-centric developments.

**Visibility and Controllability:** As net-centric systems under test become increasingly complex, the ability to visualize complex interactions and exert control over such interactions becomes increasingly vital for the test organization's ability to provide credible test results.

**Real-time Interactivity:** This includes visibility into events and processes through a display/representation of the test environment that is tailorable and provides accurate situational awareness of the test infrastructure and the tests that are underway. Currently, many test test environments focus on relatively simple interactions and do not allow for highly complex many-on-many scenarios in which test environment components (networks, systems, and forces) react within a dynamic, closed-loop environment.

**Features of Advanced Test Organizations:** the test organization should strive to be on the cutting edge of test organization capabilities, including

- **Agility:** Ability to automatically and adaptively monitor and manage selective functioning of the test infrastructures, test scenarios, networks, and systems and services under test.
- **Automation:** Ability to continually enhance the degree of automation of all the processes involved in defining, implementing, managing, reusing, and executing test events. This includes automated self-organizing recognition, initialization, and control of plug-and-play test environment components.
- **Scalability and Applicability to Full Life Cycle:** Ability to scale the test infrastructure in terms of size, fidelity, and numbers of participants to accommodate the domains of systems engineering, development, development testing, operational testing, interoperability certification testing, and net readiness and information assurance testing.

**GIG/SOA Integrated Robust Computer and Communication Infrastructure**: Ability to provide high-performance computational support wherever needed in the configuration and execution of the test environment and the analysis of test data (in real time and after test). As the SoS and collaborations brought in by customers for testing become increasingly complex, the test organization will require increasingly powerful computing resources to manage all aspects of testing. The test organization will also require the ability to provide reliable, cost-effective, flexible, and GIG-enabled communication to all nodes.

*Most of these requirements are not achievable with current manually based data collection and testing. Instrumentation and automation based on model-driven and systems-engineering modeling and simulation principles and frameworks are needed to meet these requirements.*

## Proposed Acquisition Strategy

Acquiring all the assets needed for the above capabilities would significantly upgrade the test organization's capability for net-centric testing, but they will vary in degree of maturity. Some may be ready for implementation or purchase in the near term, and others are may require significant investment in research and development. To help manage the acquisition of such assets, we propose an acquisition strategy having three levels corresponding to long-, medium-, and short-term planning horizons: 1) Overall plan for test infrastructure evolution, 2) Test infrastructure development to address test technology shortfalls, and 3) Planning for individual test venues and events (Figure 1). The underlying objective of the proposed strategy is to foster re-use of existing assets so as to maximize the cost-effectiveness of acquisition. **The goal should be to set up a process for re-use so that new capabilities are needed only when existing ones cannot be reasonably applied to the new situation.**

| |
|---|
| **Long term – Overall evolution of net-centric T&E infrastructure.** |
| **Medium term – Test infrastructure development plan to address test technology shortfalls.** |
| **Short term – Planning for individual test venues and events.** |

**Figure 1. Net-Centric Testing Planning Levels**

We proceed to describe each of the planning levels.

***Long-Term Planning***

With respect to long-term planning, the objective is to look out past the horizon of imminent test events and current infrastructure improvement projects to identify emerging technologies and emerging system objectives and to lay out the broad approach to development of the test and evaluation infrastructure. As Figure 2 illustrates, we suggest a planning approach to test individual customer projects and test events as part of the longer life cycle of the test infrastructure evolution. Key activities in the long-term strategic plan are:

- As new systems are defined and developed by a customer that will be subject to the test organization certification, the test organization must derive a coherent family of test objectives from the stated or to-be-developed, system under test requirements and behavior specifications.

- Synchronize test events, venues, and infrastructure evolution with the customer system development schedule.
- Determine the high-level characteristics of the test development methodology and of the infrastructure to be used to meet the perceived complexity, volume, variety, and velocity of test challenges—with the objectives of furthering re-use of test resources and fostering cumulative knowledge management. This includes, among other things, establishing requirements for infrastructure development tools, such as formalizing and designing test models.

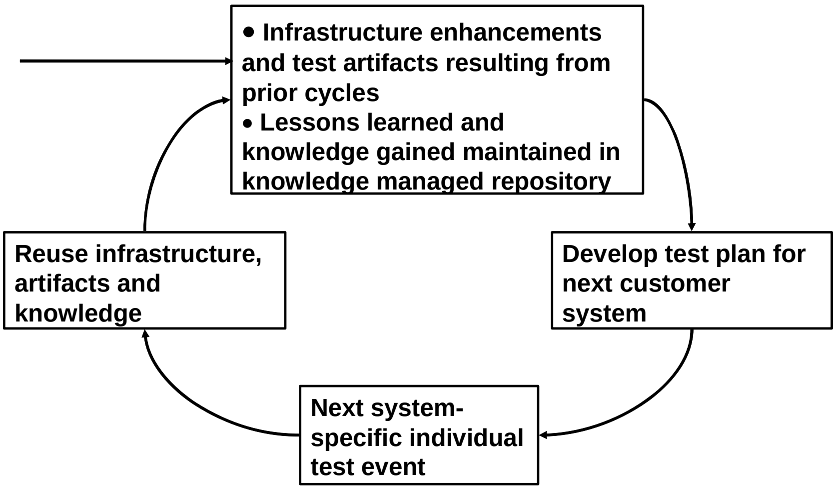


**Figure 2. Long-Term Cycle of Test Activities**

This long-term planning process passes technical shortfalls and their temporal attributes (e.g., needed immediately, needs can be foreseen for tests scheduled in the near future, or is not critical now) on to medium-term planning.

***Medium- and Short-Term Planning***

The planning for individual test venues and events consists of a cycle of activities that work within the structure established by the high level planning. As Figure 3 illustrates, this cycle consists of the following basic elements:

**Establish objectives:** The test objectives must provide an overview of the high-level, system specific test objectives and identify basic technical and operational evaluations that are needed to support future decision events. The objectives must:

- Be tied to the system acquisition strategy.
- Establish the basis for a test and evaluation schedule in terms of test capabilities that will be available after each iteration of the test and evaluation process—this should

include both anticipated costs and timelines. It is vital that the test organization and the customer agree to an integrated budget and timeline for each test objective.

- Be coordinated with the customer's strategy for system development and demonstration.
- Identify major strategic risks to achieving the identified test capabilities and lay out the activities necessary to mitigate the risks.
- Identify challenges, such as from complexity and need for testing that cannot be accomplished manually in sufficient volume, which must be overcome to effectively assess SoS and systems to contribute to their improvement. Update plans to meet these challenges.

**Identify relevant test environment requirements:** Once the test objectives are set, identify and evaluate specific test-support capabilities with respect to how they contribute to satisfying the test objectives. At this stage, a test environment description is constructed, which is tailored to the test objectives; relevant capabilities of the system under test are identified, and testable metrics are developed for those capabilities.

**Reuse/build scenarios and mission threads to exercise given system under test requirements:** The list of requirements for the system under test is linked to the underlying operational concepts and capabilities. With this list in hand, it is vital to develop specific mission threads that exercise these capabilities in a way that is relevant to the test objectives and anticipated operational environment.

**Identify atomic functional units, decompose such functions into atomic behaviors, and implement test behaviors:** The preceding three activities set the stage for technical development of the test environment. The technical development phase includes 1) Identifying the atomic functional units of the system under test that comprise the identified capabilities, 2) Decomposing these functional units into atomic testable behaviors, and 3) Combining these test behaviors as test models that can be compared to, and operated against, the system under test in the test environment. At this point, specific system under test components and/or subsystems are identified as being relevant to specific system capabilities in the context of identified mission threads and the test machinery needed to stimulate and observe these components is ready to be put into place.

**Build and/or reuse test bed software and hardware for executing test models; design and execute test events**: Test events are planned to apply specific test bed items to the system under test. The test plan includes a test environment configuration for the test events, identifies the source of test data (e.g., live data, recorded system traces, or simulations), and sets specific pass/fail criteria for the event. Acquire, build, and/or improve infrastructure development tools, such as tools for formalizing and designing test models.

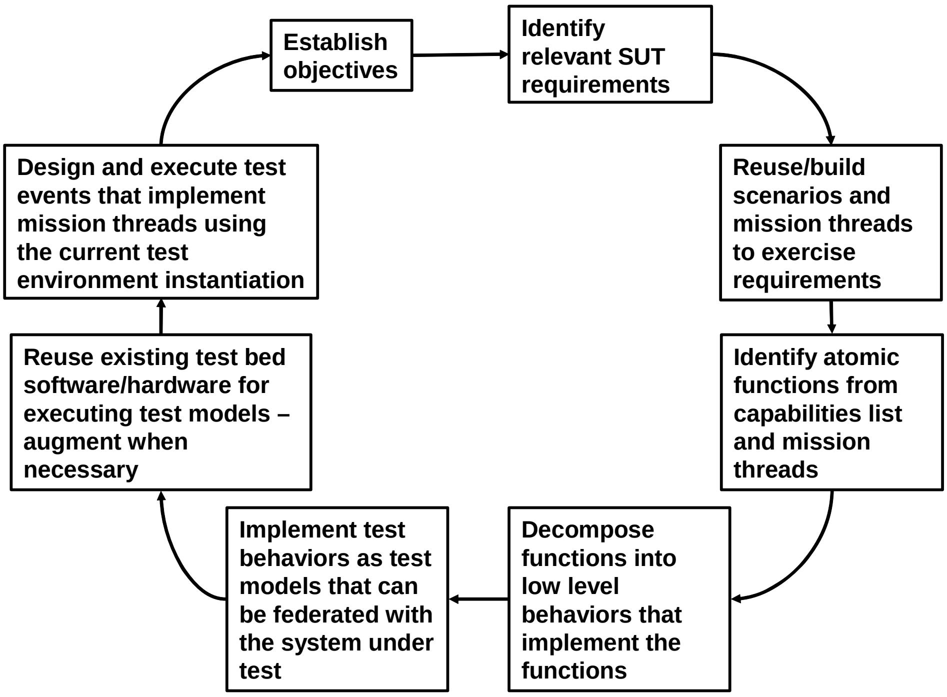


**Figure 3. System-Specific and Individual Event Planning Cycle**

This cycle of test activities defines an iterative process that allows for the evolution of each test phase as the system under test moves through its life cycle. Throughout the cycle of test activities, there must be an emphasis on the reuse of proven, reliable, and efficient infrastructure elements and artifacts that were acquired as a result of earlier test projects. Efforts first capitalize on reusing existing software and hardware for executing test models. Of course, the requirements of each new project may exceed the capabilities of the current infrastructure and artifacts, in which case we seize opportunities to enhance the infrastructure. Thus, each specific system under test feeds back lessons learned and contributes to long-term capabilities and knowledge. This feedback loop is illustrated earlier in Figure 2.

## Proposed Layered Architecture

To support the acquisition of net-centric testing capability with the time horizons just discussed, we offer a layered architecture that provides a framework for such capability acquisition. We propose that the test organization develop an overall architecture for net-centric instrumentation as illustrated in Figure 4. The architecture is based on that presented in [SAR01] and refers to background in literature on modeling and simulation, see for example, [ZEI05][ ZEI00 ] [ZEI07] [TRAO], Systems of Systems [SAG07] [WAY92][WAY76][MOR04], model-driven software development [DIM07][DIM06] [OMG] [SIAP][Wad02] [Weg02], and integrated simulation-based development and testing [MAK07] [MIT06a] [Mit06b] [MIT07g][MIT07i].

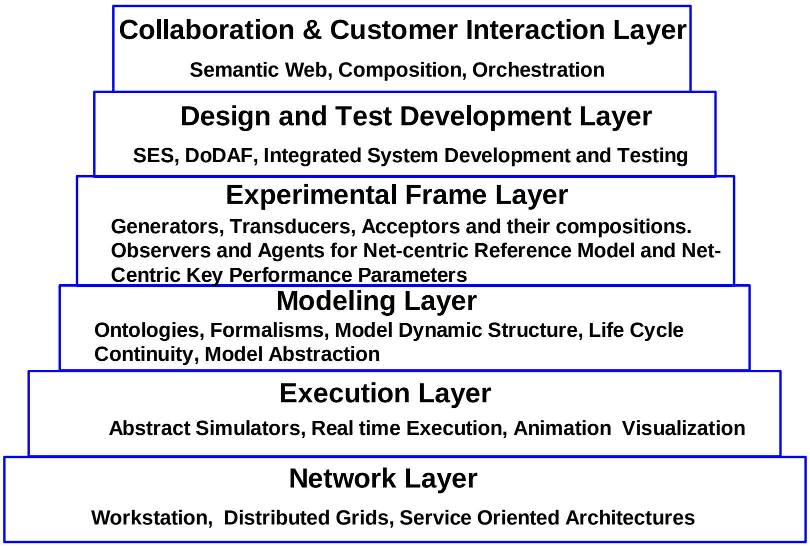


**Figure 4. Architecture for Net-centric Test Instrumentation**

- *Network Layer* contains the actual computers (including workstations and high performance systems) and the connecting networks (both local area network and wide are network, their hardware and software).

- *Execution Layer* is the software that executes the models in simulation time and/or real time to generate their behavior. Included in this layer are the protocols that provide the basis for distributed simulation (such as those that are standardized in the High Level Architecture). Also, included are database management systems and software for controlling simulation executions and for displaying test results and animated visuals of the behaviors generated.

- *Modeling Layer* supports the development of simulation models and other digital representations for net-centric testing in formalisms that are independent of execution layer implementations. At this layer, the test organization would compose services and applications. Also in this layer is support for the quality control of model acquisition, especially the key processes of verification and validation of models, simulators, and test tools.

- *Experimental Frame Layer* employs the artifacts and services of the Modeling Layer to develop test components, such as generators, acceptors, and transducers and their compositions, to provide test instrumentation services. Included are the observers and agents that run in the execution layer, and that interface with the systems and services under test to connect them to the experimental frame components. Also, included are means to capture relevant measures of performance and effectiveness and instrument them as experimental frame compositions employing modeling layer and execution layer services. These measures are critical to the testing of NR-KPPs that the test organization must be able to accomplish.

- *Design and Test Development Layer* supports the ingestion and analysis of model-based system specification documents, such as in the Department of Defense (DoD) Architecture Framework, where the design is based on specifying desired behaviors through models and implementing these behaviors through interconnection of system components. In the Modeling Layer, results of this analysis of system behavior requirements will be used with automated generation of test models that when deployed in the Execution Layer as automated test cases that will interact with systems and services under test. The *Design and Test Development Layer* also includes maintenance and configuration support for large families of alternative test architectures, whether in the form of spaces set up by parameters or more powerful means of specifying alternative model structures such as provided by the System Entity Structure (SES) methodology. Artificial intelligence and simulated natural intelligence (evolutionary programming) may be brought in to help deal with combinatorial explosions occasioned by analysis for test development.

- *Collaboration and Customer Interaction Layer* enables people and/or intelligent agents to manage and control the infrastructure capabilities supplied by underlying layers. This includes interactions with the customer in which test results are conveyed and explained if needed.

Note that these layers describe functionalities that can be partially supplied by proven and reliable legacy tools in the test organization's inventory from earlier developments. However, **the primary objective of such architecture is to facilitate carrying out the multi-horizon planning approach** discussed earlier. As customer projects arrive, their testing requirements can be referenced to the elements within the layered architecture—the detailed test assets at the various levels are called out. Missing assets can be the cues to start an acquisition process to fill the gap. Figure 6 illustrates the application of the layered architecture to sensor simulation infrastructure acquisition.

Artifacts, such as models and test and evaluation are results of processes (systems) that must not only have hardware and software support but must be done by competent people using competent methods in an environment that fosters each process. Indeed, to be effective, there must be collaboration among layers and continuity of people, methods, software and hardware, good input and materials, and a supportive environment (e.g., from management and external networks). This collaboration is illustrated in Figure 5, employing the basic categories of *People, Policy and Methods, Hardware and Software, Input Data and Materials* and *Environment* expressing the areas DoD often refers to as DOTMLPF—doctrine, organization, training, materiel, leadership, personnel, facilities. To better communicate the main collaboration path, connections for exception handling and additional feedback have not been included in Figure 5. We recognize that a real-world portrayal of the collaboration would include numerous iterations, feedback, and exception handling.

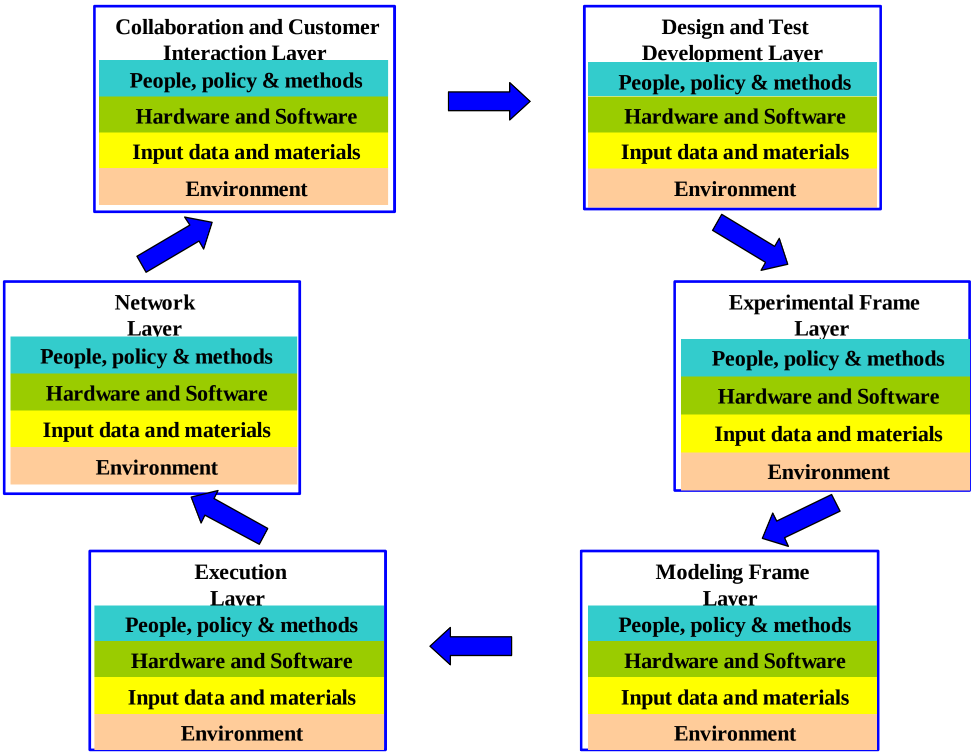


**Figure 5. The Layered Architecture Viewed from the DOTMLPF Perspective**

Table 1 suggests how some of the identified layers can be further elaborated in terms of representative needs that must be met in the basic categories that are most pertinent to each layer.

| Layer | People, Policy and Methods | Hardware and Software | Input Data and Materials | Environment |
|---|---|---|---|---|
| Experimental Frame Layer | Experimental Frame Developers 1) are qualified, 2) have methodologies that are appropriate and effective, 3) have shared awareness of development plans, design decisions, and progress, and 4) have good access to model developers and to test development personnel who are prepared to clarify requirements and standards governing the SUT. | 1) Access to relevant models and software to gather required measures (MOEs, MOPs), generate required stimuli and loads, and control. 2) Model development tools and software integrated design environments are adequate. 3) Access to JITC network and to test workstations. | 3) V&Ved experimental frame artifacts and test components from the Modeling Layer 2) V&Ved data for DT, & V, V & T. 3) Good requirements and/or standards. 4) V&Ved means to capture relevant measures. | 1) Development, testing, and V&V are managed to plan. 2) Proper SW CM environment and practice. |

| Layer | People, Policy and Methods | Hardware and Software | Input Data and Materials | Environment |
|---|---|---|---|---|
| Design and Test Development Layer | Designers/Test Developers 1) are qualified, 2) have methodologies that are appropriate and effective, 3) have shared awareness with the JITC team, and 4) have good access to personnel who are prepared to clarify requirements and standards governing the SUT. | Adequate tools to capture and characterize SUT behaviors and interfaces. | 1) Adequate system specification documents and DoDAF documents.<br>2) Behavior requirements and/or standards are sufficiently well-specified. This applies particularly to GIG/SOA based developments (e.g. NCES, NECC). | 1) Unplanned requirement additions are avoided.<br>2) Proper CM environment and practice. |

**Table 1. Illustrating the Layered Architecture in Relation to DOTMLPF**

We note that the table makes clear that besides the acquisition and application of test infrastructure elements, JITC must plan for acquiring the right personnel and instituting the right organization. Specifically, JITC must develop a culture that will facilitate the interactions among personnel that are critical for the enterprise to be effective.

## Mapping Shortfalls to Architectural Layers

The proposed layered architecture will provide a framework for focusing the planning and acquisition of the test infrastructure capability. With the Xs in the cells of Table 2 we offer a mapping to the shortfall areas that we think are best addressed in each layer. The test organization should employ this architecture as the basis for its net-centric instrumentation plan.

**Table 3. Illustrating the Mapping of Shortfalls in Architectural Layers**

| Layer | Composability | Reusability and Persistence | Extensibility | Instrumented Trustworthy Measurement | Visibility and Controllability |
|---|---|---|---|---|---|
| Network Layer | | x | | | x |
| Execution Layer | | | | | x |
| Modeling Layer | x | x | x | | |
| Experimental Frame Layer | x | x | x | x | |
| Design and Test Development Layer | x | x | x | | |
| Collaboration and Customer Interaction Layer | | | | | x |

| Layer | Real-Time Interactivity | Agility | Automation | Scalability and Applicability to Full Life Cycle | GIG/SOA Integrated Robust Computer and Communication Infrastructure |
|---|---|---|---|---|---|
| Network Layer | | | x | x | x |
| Execution Layer | x | | x | x | x |
| Modeling Layer | | | x | | x |
| Experimental Frame Layer | x | | x | | x |
| Design and Test Development Layer | | x | x | x | x |
| Collaboration and Customer Interaction Layer | | x | x | x | x |

## Strategies for Net-Centric Instrumentation Planning

**With the Layered Architecture as basis, the test organization can develop specific strategies that take into account long-, medium-, and short-term considerations for orderly acquisition of effective and reusable infrastructure.** One alternative is to continue to rely on legacy tools while employing the architecture to plan for new tool acquisitions as the opportunities present themselves. Another alternative is to invest immediately in high priority tool developments that are compliant to such an architecture and that implement non-existent capabilities such as planning or automated testing and may not replace legacy tools in the near term.

### Illustrative Application to Sensor Simulation Infrastructure Acquisition

Figure 6 sketches how the planning cycle of Figure 2 might apply to the acquisition of sensor simulation for net-centric testing.

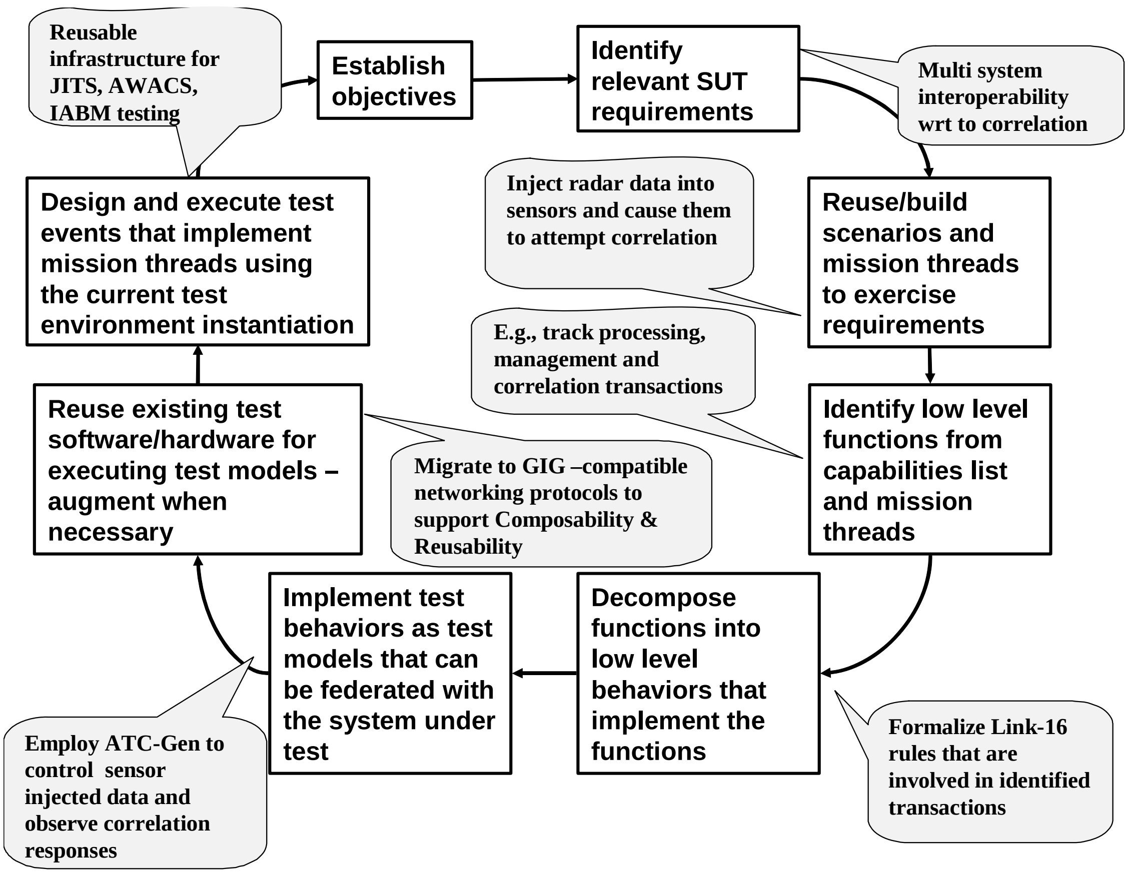


**Figure 6. Illustrating Event Planning Cycle for Sensor Simulation Acquisition**

The perspectives offered by multi-horizon planning and layered test infrastructure architecture are intended to facilitate developing and evaluating acquisition strategies. By themselves, they do not decide the choices to make.

## Summary and Recommendations

A test organization needs an instrumentation development and maintenance system that can be considered an open subsystem of an open system—the test organization, test evaluation, and certification system, which produces results shown on the left side of Figure 7.

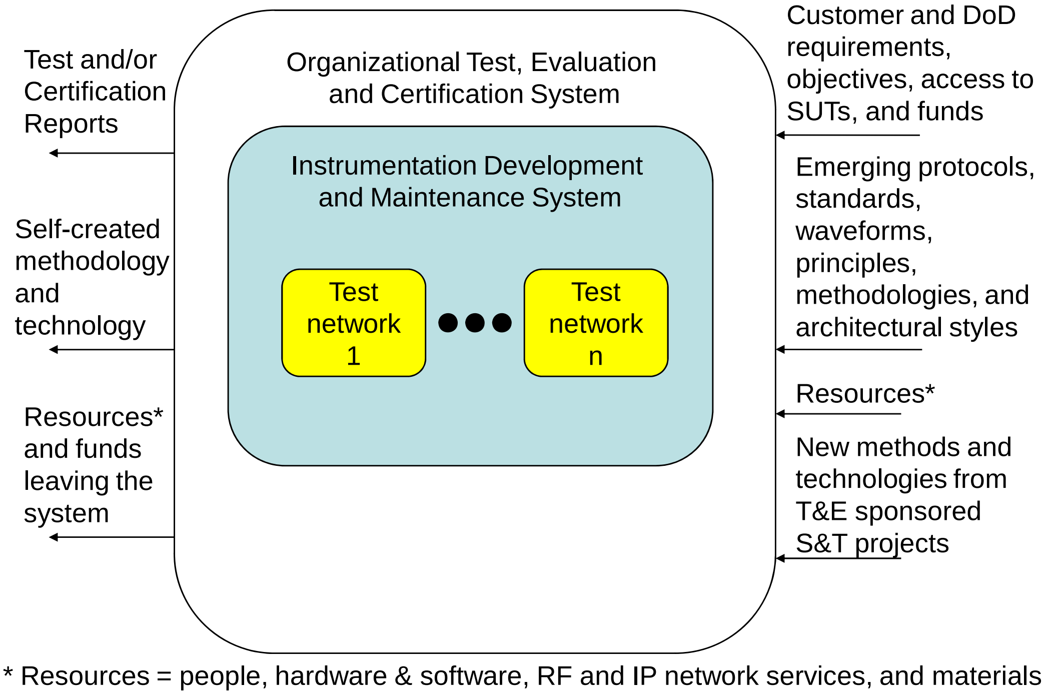


**Figure 7. Instrumentation Development and Maintenance Subsystem of the test organization Test and Evaluation and Certification System**

Shown on the left are the resources and funds leaving the system, and on the right are the funds and resources coming in. Additionally, entering at the right is a seemingly high volume of a broad variety of not always clear or fixed system-under-design requirements, protocols, waveforms, standards, and mandated architectural styles (e.g., net-centric reference model and SOA). As shown at bottom right, the test organization must encourage scientific research and technology development projects of the government, academia, and industry to develop methods and technologies needed to fill test capability gaps.

The specific inclusion of infrastructure development as an integral part of the top-down approach fosters significant reuse of test resources and cumulative knowledge management of the products of testing. We recommend that in addition to basic test development, each iteration of the individual test event/venue planning cycle should *also target a small, well-defined, and* *incremental enhancement of the test environment functionality that we implement as components of the overall test infrastructure*. Iterations should refine and/or enhance test objectives and develop and/or modify the test bed technology as needed; and test events should realize these test objectives using the available test bed capabilities. In addition to supporting the planned test objectives, every iteration should, to the extent possible, include a test event that specifically demonstrates the new test environment functionality.

Testing in this paradigm is objectives-driven rather than event driven (i.e., test events must be traceable back to established test objectives). In most cases, major shortfalls of test technology should be identified early, either during the refinement/expansion of test objectives, or in the early phases of test event planning. Interim technology solutions to reduce shortfalls that are identified late in test event planning or even later during test event execution which should be considered tentative pending review in the next iteration of the test bed development. These interim solutions should be the exception and not the rule.

We recognize that infrastructure development requires competent people using competent methods in an environment that fosters the development of each process and artifact. In this regard, we recommend including in the test organization team a test-infrastructure development component that supports testing for each customer project and its test events. The responsibilities of this infrastructure team would be to

- Identify existing, reusable testing tools and requirements that are common across test activities for use and for potential adaptation or conversion to a reusable component.
- Build and maintain reusable technical components of a common test infrastructure
- Promote test asset reuse where appropriate
- Advise test event planning and execution when the events rely on pieces of the common test infrastructure
- Retain and disseminate lessons learned from a test event

In addition to the net-centric test infrastructure components involved in specific customer projects, the test organization should stand up a global test infrastructure development team to operate within the larger framework of its enterprise level plans for coordinating instrumentation, automation, and architecture support across all the test organization portfolios. This team would:

- Coordinate efforts for customer-specific developments with the test organization's enterprise level net-centric test infrastructure development and identify overlapping concerns and/or testing tools. Customer-specific testing requirements can be referenced to the elements within the layered architecture, calling out detailed test assets at the various levels. Missing assets can be the cues to start acquisitions.
- Provide *proactive* technical solutions to identified customer-specific test requirements. These solutions will be incorporated into test events that will be planned in detail later on in the test and evaluation process.
- Seek out and recommend best practices and cultural innovations that will facilitate effective coordination of the personnel working at the various architectural layers as customer projects arrive.
- Participate actively in teams responsible for test planning and developing test tools for specific events. Successful reuse requires positive involvement at all levels of the

organization.  Consequently, persons responsible for long-term infrastructure development must be constructively and actively engaged with the elements of the organization that they support.

# References


[BUC04] J.B. Buchheister, “Net-centric Test & Evaluation”, Command and Control Research and Technology Symposium: The Power of Information Age Concepts and Technologies, 2004, available at: www.dodccrp.org/events/2004/CCRTS_San_Diego/CD/papers/208.pdf, last accessed October 2005

[CAR05] Carstairs, D.J., “Wanted: A New Test Approach for Military Net-Centric Operations”, Guest Editorial, ITEA Journal, Volume 26, Number 3, October 2005

[DIM07] M.J. DiMario SoSE Discussion Panel Introduction, From Systems Engineering to System of Systems Engineering, 2007 IEEE International Conference on System of Systems Engineering (SoSE). April 16th -18th, 2007, San Antonio, Texas

[DIM06] M.J. DiMario System of Systems Interoperability Types and Characteristics in Joint Command and Control, Proceedings of the 2006 IEEE/SMC International Conference on System of Systems Engineering, Los Angeles, CA, USA - April 2006

[MOR04] J. Morganwalp and A. P. Sage, "Enterprise Architecture Measures of Effectiveness," *International Journal of Technology, Policy and Management*, vol. 4, pp. 81-94, 2004.

[MAK07] E Mak, S Mittal, MH Hwang, “Automating Link-16 Testing using DEVS and XML”, Journal of Defense Modeling and Simulation, to appear

[MIT06a] Mittal, S. "Extending DoDAF to Allow DEVS-Based Modeling and Simulation", Special issue on DoDAF, Journal of Defense Modeling and Simulation JDMS, Vol 3, No. 2.

[Mit06b] Mittal.S., Mak, E. Nutaro, J.J., “DEVS-Based Dynamic Modeling & Simulation Reconfiguration using Enhanced DoDAF Design Process”, special issue on DoDAF, Journal of Defense Modeling and Simulation, Dec 2006

[MIT07g] Mittal, S, DEVS Unified Process for Integrated Development and Testing of Service Oriented Architectures, Ph. D. Dissertation, University of Arizona

[MIT07i] Mittal S, Ferat Sahin, and Mo Jamshidi, Modeling and Simulation for Systems of Systems Engineering to appear in Systems of Systems -- Innovations for the 21st Century (to be published by Wiley)

[OMG] Object Modeling Group (OMG) www.omg.org

[SIAP] Jacobs, Robert W. “Model-Driven Development of Command and Control Capabilities For Joint and Coalition Warfare,” Command and Control Research and Technology Symposium, June 2004.

[SAG07] Andrew Sage: From Engineering a System to Engineering an Integrated System Family, From Systems Engineering to System of Systems Engineering, 2007 IEEE International Conference on System of Systems Engineering (SoSE). April 16th -18th, 2007, San Antonio, Texas

[SAR01] H. Sarjoughian, B. Zeigler, and S. Hall, *A Layered Modeling and Simulation Architecture for Agent-Based System Development*, Proceedings of the IEEE 89 (2); 201-213, 2001

[TRAO] Traore M, Muxy A, Capturing the Dual Relationship between Simulation Models and Their Context, SIMPRA (Simulation Practice and Theory), Elsevier, 2004

[Wad02] L.W Wagenhals, S. Haider, A.H.Levis, “Synthesizing Executable Models of Object Oriented Architectures”, Workshop on Formal Methods Applied to Defense Systems, Adelaide, Australia, June 2002

[Weg02] Wegmann, A., “Strengthening MDA by Drawing from the Living Systems Theory”, Workshop in Software Model Engineering, 2002

[WAY76] Wayne A Wymore, A Mathematical Theory of Systems Engineering: The Elements, Krieger, Huntington, NY, 1967.

[WAY92] Wayne A Wymore, Engineering Modeling and Design, (with Bill Chapman and A. Terry Bahill), CRC Press Inc., 1992.

[ZEI00] Zeigler, B. P., T. G. Kim, and H. Praehofer. (2000). Theory of Modeling and Simulation. New York, NY, Academic Press.

[ZEI05] Zeigler, B.P., Fulton, D., Hammonds P., and Nutaro, J. Framework for M&S–Based System Development and Testing In a Net-Centric Environment, ITEA Journal of Test and Evaluation, Vol. 26, No. 3, 21-34, 20

[ZEI07] Zeigler, B.P., Bernard P. Zeigler, Bernard P. Zeigler, and P. Hammonds, “Modeling & Simulation-Based Data Engineering: Introducing Pragmatics into Ontologies for Net-Centric Information Exchange”, 2007, New York, NY: Academic Press.

**Biographies:**

## Steven Bridges

**Bernard P. Zeigler** is Professor of Electrical and Computer Engineering at the University of Arizona, Tucson and Director of the Arizona Center for Integrative Modeling and Simulation. He is internationally known for his 1976 foundational text *Theory of Modeling and Simulation*, recently revised for a second edition (Academic Press, 2000), He has published numerous books and research publications on the Discrete Event System Specification (DEVS) formalism. In 1995, he was named Fellow of the IEEE in recognition of his contributions to the theory of discrete event simulation. In 2000 he received the McLeod Founder's Award by the Society for Computer Simulation, its highest recognition, for his contributions to discrete event simulation. He received the JITC Golden Eagle Award for research and development of the Automated Test Case Generator, 2005 and the Award for Best M&S Development in the Cross-functional Area, 2004/2005, by the National Training Simulation Association, May 2, 2006.

**James Nutaro**

James Nutaro is with Modeling and Simulation group at the Oak Ridge National Laboratory in Tennessee. He obtained his Ph. D from dissertation, the University of Arizona entitled Parallel Discrete Event Simulation with Applications to Continuous Systems, and has published papers describing his research in th earea. He was a post-doctoral researcher at the Arizona Center for Integrative Modeling and Simulation. He has several years of employment experience developing and applying distributed simulation software in a defense systems context.

**Dane Hall**

**Tom Callaway**

**Dale Fulton**