A Case Study of Verification, Validation, and Accreditation for Advanced Distributed Simulation

ERNEST H. PAGE, BRADFORD S. CANOVA
The MITRE Corporation
and
JOHN A. TUFAROLO
BDM International

The techniques and methodologies for verification and validation of software-based systems have arguably realized their greatest utility within the context of simulation. Advanced Distributed Simulation (ADS), a major initiative within the defense modeling and simulation community, presents a variety of challenges to the classical approaches. A case study of the development process and concomitant verification and validation activities for the Joint Training Confederation (JTC) is presented. The JTC is one of the largest current ADS efforts, and the primary application of the Aggregate Level Simulation Protocol. A dichotomy between classical verification and validation approaches and the requirements of a prototypical ADS environment is illustrated. Mechanisms and research directions to resolve these differences are briefly discussed.

Categories and Subject Descriptors: I.6.4 [**Simulation and Modeling**]: Model Validation and Analysis; I.6.5 [**Simulation and Modeling**]: Model Development—*modeling methodologies;* I.6.8 [**Simulation and Modeling**]: Types of Simulation—*discrete event, distributed, gaming*

General Terms: Design, Management, Verification

Additional Key Words and Phrases: Advanced distributed simulation, aggregate level simulation protocol, IDEF modeling, life cycle, validation and accreditation, verification, wargame

1. INTRODUCTION

Verification, validation and accreditation (VV&A) is the collective term used within the United States Department of Defense (DoD) community to

---

This work was supported under contract DAAB07-97-C-E601 while J. A. Tufarolo was a member of the technical staff of the MITRE Corporation.

Authors' addresses: E. H. Page and B. S. Canova, The MITRE Corporation, 1820 Dolley Madison Boulevard, McLean, VA 22102; email: {epage,bcanova}@mitre.org; J. A. Tufarolo, BDM International, 1501 BDM Way, McLean, VA 22102; email: jtufarol@bdm.com.

describe the methods, techniques, and processes through which models and simulations (M&S) are evaluated and formally approved for use. The DoD investment in M&S is substantial, and the Executive Council for Modeling and Simulation (EXCIMS) and the Defense Modeling and Simulation Office (DMSO) have been established to manage M&S expenditures. But what exactly does "M&S" imply in this context? The classical taxonomy of discrete event, continuous and Monte Carlo simulation (see Nance [1993]) only partially covers the issue. Within the DoD arena simulation is regarded as encompassing everything other than war itself.[1] Accordingly, three types of "simulation" are prominently referenced: (1) live, (2) virtual, and (3) constructive. Precise characterizations of these terms are difficult to provide since the boundary between them is often blurred by common elements. For purposes of this presentation however, the definitions of Sikora and Coose [1995] suffice: a *live* simulation involves real people operating real systems in realistic operational conditions, a *virtual* simulation involves real people operating in simulated systems, while *constructive* simulation refers to the more commonly recognized computer simulations (simulated people operating in simulated systems).

A major goal of the EXCIMS and DMSO is an environment that provides the "seamless integration" of live, virtual and constructive simulations. Efforts toward this objective are often referred to as Advanced Distributed Simulation (ADS). ADS is envisioned to support the gamut of DoD activities, including: analysis, training, test, and evaluation, as well as acquisition. A significant portion of the current DoD M&S budget falls under the auspices of ADS.

The relationship between ADS and VV&A warrants examination. ADS presents a variety of challenges to existing methodologies and techniques for simulation model verification and validation. Many of the constituents in an ADS environment may be wholly unlike the "traditional" discrete event, continuous and Monte Carlo simulations upon which many of the classical VV&A techniques are based (see Balci [1994b] for a comprehensive survey). ADS environments, particularly those that support training, often include human-in-the-loop aspects. Additionally, training simulations—ADS or otherwise—rarely contain a formulation of the kind of output process that is typical in, say, the discrete event simulation (DES) world. The DES model validation paradigm of comparing system outputs to model outputs for statistically relevant differences is not generally utilized in interactive settings. A variety of measures of performance (MOPs) and measures of effectiveness (MOEs) may be used for validation of interactive models, but these can be quite difficult to construct and even more difficult to evaluate in operational settings (see Hopkinson and Sepúlveda [1995]; Knepell and Arangno [1993]). Many ADS efforts contain animation of model behavior. Animation *can* significantly aid in model validation, but model animation is a mixed blessing. Captivating, real-time, high-resolu-

---

[1] The U.S. Army Simulation Training and Instrumentation Command logo incorporates the phrase "All but war is simulation."

tion images are persuasive, but can provide a superficial picture that belies the underlying truth with respect to the time and state relationships in the model. And more often than not, seeing is believing. In the absence of an objective appeal to statistical methods, the probability of committing Type II error (see Balci [1990]) is enhanced. Other complicating factors are perhaps more programmatic than technical, e.g. operating within a vast array of political, cultural, and organizational constraints. Factors such as these should not be discounted but are nonetheless rarely addressed in the simulation literature.

This article describes the verification, validation and accreditation process used within one of the largest current ADS efforts, the Aggregate Level Simulation Protocol (ALSP) Joint Training Confederation (JTC). The JTC is a collection of constructive training simulations that supports joint training at the command and battle staff levels during several major exercises each year. The primary objective of the article is to document "lessons learned," both in terms of failures as well as successes, for the benefit of future, similar systems. The remainder of this article is organized as follows. Section 2 provides the context for the case study, outlining the terminology used in the article and briefly reviewing both ALSP and the JTC. Section 3 presents a development process model for the JTC and highlights the VV&A activities within it. An evaluation of the process is given in Section 4. Initiatives that mark the future for joint training are briefly discussed in Section 5, and conclusions appear in Section 6.

2. BACKGROUND

The scope of this article is limited to a case study. The reader is assumed to have a working familiarity with the fundamental principles in several areas, including: discrete event simulation, distributed simulation, interactive simulation, wargaming and VV&A. Some context for the case study is warranted however, and a few relevant items are reviewed briefly below.

2.1 Terminology

A clear indication that a discipline is in the throes of infancy is the presence of a "terminology debate." While the field of simulation has emerged from this stage in many areas (see, for example, Nance [1981]), with respect to verification and validation (V&V), the promise of adolescence seems further off. Perhaps this is due to the applicability of these concepts in areas other than simulation: from software engineering, to systems engineering, to expert systems. Nonetheless, definitions for V&V abound and these definitions often contain subtle differences, if not absolute contradictions.

We leave for others the task of resolving the terminology debate for verification and validation. For purposes of this presentation, the terminology described by Balci [1994b]—which represents the consensus view within the discrete event simulation (DES) community and is consistent with Law and Kelton [1991], Pace [1993], Sargent [1992] and Sikora and Williams [1994]—is adopted.

Validation involves substantiating that the model, within its domain of applicability, behaves with satisfactory accuracy consistent with the objectives governing its use. It relates to the comparison of model behavior with system behavior. (Did we build the *right* model?)

Verification involves substantiating that the model is transformed from one form into another, as intended, with sufficient accuracy. (Are we building the model *right?*)

The officially sanctioned DoD definitions are as follows [U.S. Department of Defense 1996b]:

Validation. The process of determining the degree to which a model is an accurate representation of the real-world from the perspective of the intended use of the model.

Verification. The process of determining that a model implementation accurately represents the developer's conceptual description and specifications.

Accreditation. This is the official certification that a model is acceptable for use within the context of a specific objective.

The differences between the definitions presented by Balci and those sanctioned by the DoD are not relevant to this presentation. Generally speaking, however, we regard Balci's definition of verification as preferable since it conveys the notion of non-loss transformations among *multiple,* evolving model representations.

2.2 VV&A First Principle

The fundamental principle underlying VV&A is that validity is subject to diminishing returns. Further, *absolute* model validity is impossible to achieve—except, perhaps, in the most trivial of circumstances. As Law and Kelton [1991, pp. 306–312] observe:

> A simulation model of a complex system can only be an *approximation* to the actual system, regardless of how much effort is put into developing the model. There is no such thing as an absolutely valid [simulation] model. The more time (and hence money) is spent on model development, the more valid the model should be in general. However, the most valid model is not necessarily the most cost-effective one. For example, increasing the validity of a model beyond a certain level may be quite expensive, since extensive data collection may be required . . . Furthermore, we question whether hypothesis tests, as compared with constructing confidence intervals for differences, are even the appropriate statistical approach. Since the model is only an approximation to the actual system, a null hypothesis that the system and the model are the "same" is clearly false. We believe that it is more useful to ask whether or not the differences between the system and the model are significant enough to affect any conclusions derived from the model.

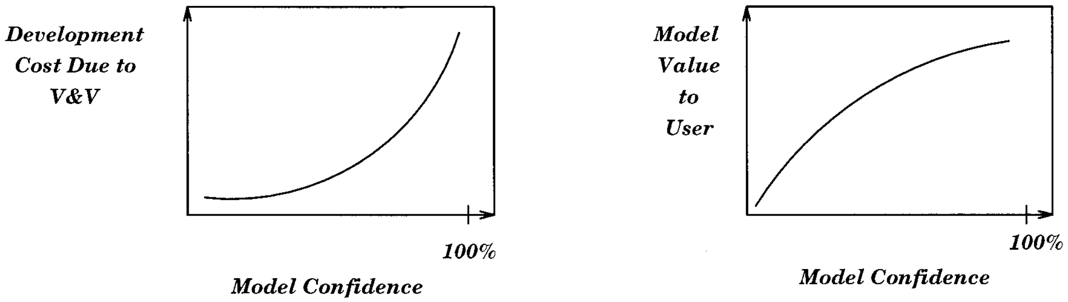

Fig. 1. The cost-benefit relationship for simulation model validation.

A similar argument appears in Pegden et al. [1990, p. 154]:

> As already stated, a simulation model is our theory describing the structure and interrelationships of a system. The theory (model) can be useful, useless, or outright dangerous, depending on whether it is sound, inadequate, or wrong. Correctness (validity) can only be judged in relationship to the real system. Since all models contain both simplifications and abstractions of the referent, real-world system, no model can ever be absolutely correct, i.e., it can never have a one-to-one correspondence with its real-world counterpart.

And in Banks et al. [1996, p. 407]:

> Validation is not an either/or proposition—no model is ever totally representative of the system under study.

This cost-benefit relationship is depicted in Figure 1 (which is a slight variation of the illustration of Sargent [1992]). We echo this principle here because it has (or should have) a significant impact on the construction of budgets for VV&A. And despite the simplicity of this principle, many in the business of both applying and paying for VV&A fail to recognize it.

2.3 The Aggregate Level Simulation Protocol

The Aggregate Level Simulation Protocol (ALSP) project was initiated in 1990 through the Advanced Research Projects Agency to examine the feasibility of extending the distributed environment utilized by SIMNET to existing, so-called "aggregate" combat simulations. SIMNET was an effort to connect tank simulators—potentially over a wide-area network—to form a common *virtual* training environment [Alluisi 1991]. Such simulators are referred to as "entity level" since the elements they portray correspond to fundamental military entities, e.g., a tank, an airplane, and so forth. Aggregate-level simulations are so named because the elements they portray are generally *collections* of fundamental military entities like tank battalions and fighter squadrons. Entity-level simulations are used to train on the small scale, i.e., training the individual soldier, whereas aggregate-level simulations provide a training environment at larger scales, i.e., training command and battle staffs.[2]

The primary objective of the ALSP investigation was to reduce the number of point-to-point, ad hoc interfaces between existing (*legacy*) aggregate-level Service simulations in an effort to provide a cost-effective environment to support *joint* (i.e., multi-Service) training.[3] Could these simulations be connected such that the environment was stable? Could the capabilities for integration be easily incorporated into the legacy models? Would the models still be able to satisfy existing (and continuing) Service training needs?

The protocol and the infrastructure software that supports it are described in Weatherly et al. [1993, 1996] and Wilson and Weatherly [1994]. A few of the basic aspects of the protocol are described briefly below.

2.3.1 *ALSP Fundamental Design Principle.* To design a mechanism that permits *existing* simulations to interact, two strategies are possible: (1) define an infrastructure capable of translating between the representations of all constituent simulations, or (2) define a common representational scheme and require all intersimulation activity to be within the context of the common representation. The former approach has the advantage that very few perturbations to the existing simulations are required; interaction is facilitated entirely through the interconnection network. However, this solution does not scale well. The ALSP design adopts the latter strategy. ALSP prescribes that a *translator* be constructed to facilitate mappings between the representational scheme of the collection of simulations as a whole, and the representational scheme of a particular simulation. The infrastructure itself is independent of the simulations it interconnects.

2.3.2 *ALSP Conceptual Framework.* A *conceptual framework* is an organizing structure of concepts that facilitates simulation model development [Balci 1990]. Also referred to as *simulation strategy* and *world view,* common discrete event simulation conceptual frameworks include: event scheduling, activity scanning and process interaction. Following the terminology of Nance [1981], the ALSP conceptual framework is fundamentally *object-based:* a *model* is comprised of *objects;* an object is characterized by *attributes* to which values are assigned.[4] A *confederation* is a collection of simulations supporting a single, common model. The object basis permits

---

[2] Here the term *scale* is used to connote echelon rather than “size” (in number of participants) of the training exercise. Entity-level exercises such as those supported by the Distributed Interactive Simulation (DIS) protocols are envisioned to support thousands of trainees.

[3] An environment for joint training could be provided in two ways: (1) by developing a single joint training simulation, or (2) by integrating existing service-specific training simulations. Both of these tacks are currently being taken within the DoD. The Joint Theater Level Training Simulation (JTLS) is an effort in the former category (see Whittman [1995]); ALSP is an effort in the latter.

[4] Within ALSP, object *classes* are organized hierarchically in much the same manner as object-oriented programming languages, however the inheritance mechanism is less powerful.

any of the common DES conceptual frameworks to provide the organizing framework for a confederation. Further, a conceptual orientation of each simulation, or *actor,* in a confederation is not mandated. The only requirement is that the internal world view of any actor must enable some well-defined mapping to the object-based conceptual framework supported by the confederation as a whole.

2.3.3 *Time and State in ALSP.* The construction of a translator represents one of the three ways in which a simulation must be fundamentally altered to participate in an ALSP-supported confederation. The remaining modifications involve [Weatherly et al. 1993, p. 1068]:

—Recognition that not all objects that a simulation perceives are *owned* by the simulation.

—Modification of the internal time flow mechanism to work cooperatively with the other simulations within the confederation.

In the familiar, "traditional" use of simulation, objects come into (and perhaps go out of) existence with the passage of simulation time between the instants defining model initialization and model termination. The disposition of these objects is solely within the purview of the simulation. When acting within a confederation, the simulation-object relationship is more complicated. In fact, ownership in ALSP is defined at the *attribute level* rather than the object level. Ownership of an attribute implies that the owning simulation is responsible for both calculating and reporting value changes for the attribute. In the parlance of ALSP, these reports are called *updates.* Value changes for all owned attributes that occur during an instant are reported in the context of a single update.

Ownership in ALSP is dynamic—attribute ownership may be transferred among simulations. While attribute-level ownership provides the maximum modeling flexibility, ownership at the object level *is* a useful notion, and by convention, a simulation is said to *own an object* if the simulation owns the "identifying" attribute, or *handle,* associated with that object. Objects not owned by a simulation but within the area of perception for the simulation are known as *ghosts.* Note, however, that since ownership is defined at the attribute level, a simulation may own one or more attributes of a ghost and conversely, might not own one or more attributes of an owned object.

When an object is created, the creating simulation reports this fact to the confederation to enable the creation of ghosts (as applicable) within other simulations. Likewise, when an object is deleted, the deleting simulation (which must own the object in order to delete it) must report this fact to enable ghost deletion. Whenever an action is taken between two public objects, the simulation owning the initiator reports the action to the confederation. In the parlance of ALSP, this is known as an *interaction.*[5]

---

[5] Typically, an interaction occurs between an owned object and a ghost. The protocol permits (for generality) both objects involved in an interaction to be owned by the same simulation.

ALSP provides both fixed-time increment and next-event time flow mechanisms (TFMs) for discrete event simulation (see Kiviat [1967] and Nance [1971]). Although the fundamental design principle espouses a common, global model representation, ALSP makes no such provision in terms of TFM implementation. Rather than appealing to a centralized event list, time flow is regulated using a variant of the Chandy-Misra-Bryant (CMB) parallel discrete event simulation (PDES) protocol [Bryant 1977; Chandy and Misra 1979; 1981].[6] A null-message scheme is used for deadlock avoidance, and the standard CMB requirement for lookahead applies. Note that in a fixed-time environment, such as the Joint Training Confederation (see Section 2.4), an actor, a, with timestep, δ, has lookahead, $l = \delta$ if no message received by a at simulation time t can cause an output message to be generated by a at t.

2.3.4 *A Brief Word on Parallel Discrete Event Simulation.* The use of a PDES protocol in the ALSP context is somewhat misleading. While efficiency of the TFM *is* a consideration, *speedup* of computation is not—for ALSP confederations there is no serial analogue to the distributed computation upon which to define speedup. The same is true for many ADS efforts: distribution of the computation is simply an artifact of the desire to utilize legacy models, or realize the cost savings that result from allowing users to operate from their "home stations." For training purposes, simulations are often constrained to run no faster than real (wallclock) time. Further, they typically must provide a user (operator) with a temporally-consistent, evolving picture of the *entire* battlefield. Therefore the statement "you must have good lookahead to achieve good performance for conservative protocols," while generally regarded as true for traditional PDES settings, is not necessarily true in an ADS context. In an interactive environment, copious lookahead necessitates an output buffering mechanism akin to GVT.[7] The overhead induced here can outweigh the benefits of large lookahead given the no-faster-than-wallclock constraint.

2.4 The Joint Training Confederation

The primary application of ALSP is the Joint Training Confederation (JTC). In its current configuration the JTC consists of eight primary models:[8]

—*The Corps Battle Simulation (CBS).* A U.S. Army model originally named the Joint Exercise Support System, CBS is used in the Battle Command Training Program (BCTP) to train corps, division, and brigade staffs

---

[6] Each actor in an ALSP confederation is roughly equivalent to a logical process (LP) in the traditional PDES paradigm.

[7] Global Virtual Time. From Time Warp (see Jefferson [1983], Jefferson and Sowizral [1982; 1983]): the smallest Local Virtual Time (LVT) for any LP in a simulation.

[8] Five additional models are under consideration for the 1997 JTC: two additional sustainment models, the Logistics Anchor Desk (LAD), and the Analysis of Mobility Platform (AMP); the Joint Command and Control Attack Simulation (JCAS); the Joint Operational Visualization Environment (JOVE); and a DIS application, the Air Force Semi-Automated Forces (AFSAF).

[Mertens 1993]. CBS portrays ground-based objects primarily at the battalion level, but mechanisms are provided to incorporate specialized units at lower levels (e.g., platoon and below). Air units (both fixed-wing and rotary-wing) are represented as either individual aircraft or missions. These units conduct a variety of air mission types, including close air support, battlefield air interdiction and airlift. Written mostly in SIMSCRIPT II.5, CBS is a discrete event simulation. Attrition calculations are based on Lanchester equations (see Taylor [1983]) supplemented by the Combat Outcome Based on Rules for Attrition (COBRA) expert system.

—*The Research, Evaluation, and Systems Analysis model (RESA).* A U.S. Navy model originally named the Naval Warfare Interactive Simulation System, RESA provides a platform for the analysis of naval command, control, and communications systems [Sonalysts Inc. 1995c]. RESA provides resolution of naval surface and subsurface forces at the individual ship and submarine level. Aircraft representation is at both the individual unit and aggregate mission levels. Written primarily in Rational FORTRAN, RESA is an interactive, time-stepped (paced by wallclock), discrete event simulation.

—*The Reengineered Air Warfare Simulation (AWSIM/R).* A U.S. Air Force model originally derived from RESA and redesigned in Ada, AWSIM/R is used to train senior commanders and their battle staffs in the execution of wartime general defense plans that emphasize joint and combined operations. AWSIM/R is a theater-level model, with a scope that covers all aspects of conventional theater-level air combat. AWSIM/R supports all existing conventional air and surface-to-air weapons, and represents air bases and radar sites. Mechanisms for day and night operations and weather effects are also provided.

—*The Marine Air Ground Task Force Tactical Warfare Simulation (MTWS).* MTWS provides a model of the littoral warfare combat environment that enables interactive, multisided, force-on-force activity for all combat and combat support units of the Marine Air Ground Task Force in a joint, combined, or stand-alone tactical combat scenario [Blais 1994; 1995]. MTWS provides a full range of command and control capabilities to the Tactical Exercise Support team, including force initialization, planning and scheduling of amphibious operations, air operations and operations ashore, integration and analysis of intelligence data, and analysis of comparative combat powers. Written mostly in Ada, the normal mode of operation for MTWS is real time synchronous. However, an event synchronization mode is provided which prevents time advance until all scheduled processing for a given (real) time is completed [Blais 1995, p. 1282]. Using this mechanism, MTWS may be classified as a discrete event simulation.

—*The Tactical Simulation Model (TACSIM).* A U.S. Army model, TACSIM provides an interactive simulation environment to support intelligence training from the brigade level through echelons above the corps level.

Table I. Training Exercises Supported by the Joint Training Confederation

Exercise	*Date*	*Location*
III Corps Exercise	December 1993	Germany
Atlantic Resolve	December 1994	Germany
Central Fortress	June 1992	Germany
Prairie Warrior (PW)	May 1994, 1995, 1996	US
REFORGER	September 1992	Germany
	May 1993	Germany
RSOI	April 1996	US, Korea
Ulchi Focus Lens (UFL)	August 1992, 1993	Korea, Germany
	August 1994, 1995, 1996	Korea, Germany, US
Unified Endeavor (UE)	May 1995	US
	October 1995	US
	April 1996	US
	December 1996	US
Yama Sakura	January 1995, 1996, 1997	Japan, US

TACSIM simulates the tasking, collecting, and reporting functions of selected US reconnaissance assets. A discrete event simulation written in FORTRAN, TACSIM produces sensor product reports in standard formats. These reports replicate those delivered to the intelligence community in wartime.

—*The Joint Electronic Combat-Electronic Warfare Simulation (JECEWSI).* A Joint Command and Control Warfare Center model, JECEWSI is an exercise driver for command post exercises designed to focus on the electronic combat environment in support of tactical air and air defense operations. Originally written in SIMSCRIPT II.5, and later ported to MODSIM III, this Monte Carlo simulation model portrays the effects of electronic warfare systems on battlefield outcomes by providing both stand-off and self-protect jamming against radars and communications. JECEWSI models an integrated air defense system (IADS) and provides the capability to degrade the IADS.

—*The Combat Service Support Training Simulation System (CSSTSS).* CSSTSS is a U.S. Army exercise driver used for collective training of logistics commanders and staff personnel in command, control, and coordination of logistics operations. Military personnel are modeled at the individual name, social security number, grade, and military occupation specialty level. Supplies are represented at the national stock number and DoD identification code level. Movement is tracked at the Transportation Control Number level, while medical patients are tracked by wound type, availability of proper medical personnel, operating rooms, blood, and evacuation assets. CSSTSS is a real-time transaction system written in COBOL.

—*The Portable Space Model (PSM).* A U.S. Space Command (SPACECOM) model, PSM is a discrete event simulation written in C that models satellite detection and early warning for tactical ballistic missiles.

Table I lists several of the large-scale, joint training exercises supported by the JTC. Typically, the training audience numbers between 500 and 1500,

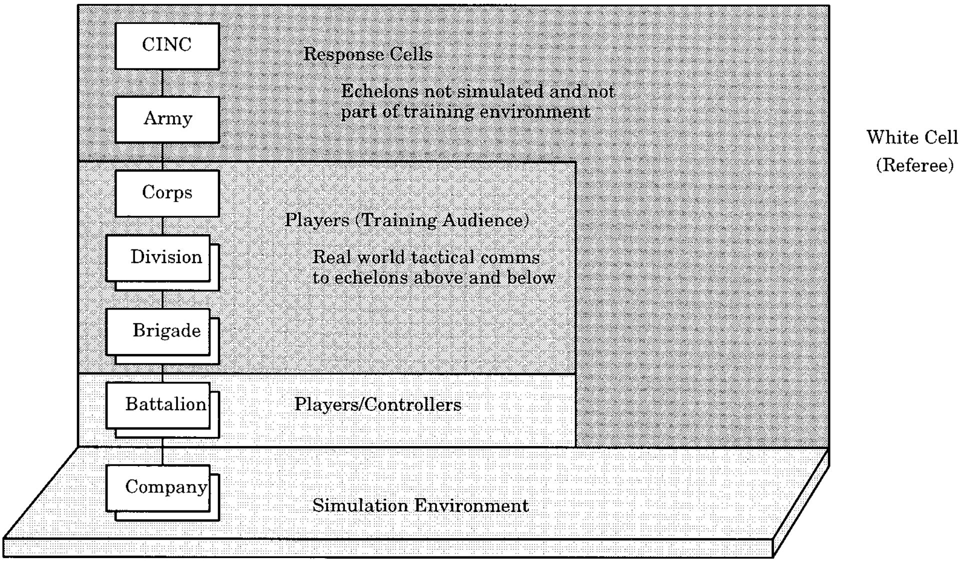

Fig. 2. Architecture for a Command Post Exercise (CPX).

and the support and ancillary personnel involved number in the several thousands. These exercises are generally *Command Post Exercises* (*CPXs*).[9] A CPX focuses training on commanders, battle staffs, Command Post and Headquarter staffs (using the Army vernacular). In a CPX the atmosphere is kept as realistic as possible: the exercise runs in real time; the lighting, housing, and communications are as they would be in combat; and the staff operate according to established doctrine using real-world mechanisms and techniques [Mertens 1993].

Figure 2 (adapted from Zabek [1994]) depicts a typical configuration for a large-scale computer-supported CPX. The term *players* refers to the training audience—in this example Corps, Division, and Brigade commanders, subordinate commanders and the affiliated battle staffs. The players are ensconced in wartime Command Posts and communicate with higher and lower echelons using real-world mechanisms and techniques—so-called Command, Control, Communications, Computers and Intelligence (C^4I) systems. In current practice, direct interfaces between C^4I systems and the fielded training simulations are rare. Orders generated by the training audience are intercepted and translated into formats recognized by the training simulations by *controllers* and other personnel staffing *response cells.* Since the impacts of human interaction and widely varying experience and skill levels may result in battlefield situations that run counter to the training objectives, referees located in a *white cell* may intervene and influence the direction of the exercise and state of the simulation(s). The

---

[9] Other types of military exercises include Field Training Exercises (FTXs) and Staff Exercises (STAFFEXs). Refer to National Simulation Center [1994] for a comprehensive taxonomy.

[A0] Develop ALSP Joint Training Confederation
- [A1] Develop Concepts
  - [A11] Create Concepts
  - [A12] Review Concepts
  - [A13] Approve Concepts
- [A2] Implement Concepts
  - [A21] Refine Concepts
  - [A22] Develop OPSPEC
  - [A23] Develop ICDs
  - [A24] Develop AIDs
  - [A25] Modify Software
  - [A26] Develop Test Plans, Enumerations
    - [A261] Develop Component Tests
    - [A262] Develop Functional Tests
    - [A263] Develop Operational Tests
    - [A264] Develop Load Tests
    - [A265] Develop Enumerations
    - [A265] Develop Technical Tests
- [A3] Test Concepts
  - [A31] Develop Test Tools
  - [A32] Conduct Component Testing
  - [A33] Conduct Functional Interface Integration
  - [A34] Conduct All-Actor Integration (AAI)
  - [A35] Conduct Confederation Test (CT)
- [A4] Deploy Confederation
  - [A41] Create Accreditation Report
  - [A42] Accredit Confederation
  - [A43] Post-CT Corrections

Fig. 3. Activities comprising the JTC development process.

controllers, response cells and white cell are not visible to the training audience.

3. A DEVELOPMENT PROCESS MODEL FOR THE JOINT TRAINING CONFEDERATION

The JTC development process is described using the Integration Definition for Function Modeling (IDEF0) approach [National Institute of Standards and Technology 1993]. In an IDEF0 model *activities* (alternatively, *functions*) are designated by boxes. Annotated, directed arcs are used to convey data or objects related to these activities. Arcs entering from the left are *inputs*—items that can be altered by the activity. Arcs entering from the top imply a *control* or *constraint*—items influencing the activity. Arcs exiting from the right define an *output*—something produced or changed by the activity. Finally, arcs entering the bottom depict a *mechanism* for conducting the activity. The collection of these arcs is referred to as an "ICOM" (Inputs, Controls, Outputs, and Mechanisms). IDEF0 permits a hierarchical decomposition of activities.

The activities comprising the JTC development process model are presented in Figure 3. The VV&A aspects of the development process are highlighted in the subsequent narrative. The interested reader should refer to: Fischer [1994]; Miller and Zabek [1996]; Weatherly et al. [1993]; Weatherly et al. [1996]; and Wilson and Weatherly [1994] for other treatments of both the protocol and the JTC.

3.1 Definitions

The following terms are used in the model.

- *Accredited Confederation Software.* The simulations and ALSP Infrastructure Software.
- *ALSP Master Plan.* Plan developed by the ALSP Executive Agent describing the overall scheme for achieving the established JTC requirements. The ALSP Master Plan is updated on a yearly basis.
- *ALSP Review Panel.* The "voting" members of the JTC community. This body approves requirements, development plans to meet requirements, and also provides formal accreditation of the JTC.
- *Executive Agent.* The Simulation, Training and Instrumentation Command (STRICOM)—has primary responsibility for JTC management.
- *Systems Engineer.* The MITRE Corporation—supports the ALSP Executive Agent, builds and maintains AIS, and administers all systems engineering and day-to-day support functions within the JTC development cycle. The Systems Engineer assumes the primary responsibility of V&V agent for the JTC.
- *Simulation Proponents, Developers.* The organizations owning and controlling the development of the simulations comprising the JTC.
- *User Community.* The training audience and their representatives.
- *ALSP Interface Working Group (IWG).* This group includes representatives from all relevant program offices and development organizations that participate within the JTC (i.e., user community, Executive Agent, Systems Engineer, ALSP Review Panel, simulation proponents and developers). The IWG serves as a consensus-driven organization that facilitates and directs JTC development. Refer to Fischer [1994] for a detailed account of the JTC management structure.
- *CINC/Service Requirements.* The set of training objectives and other requirements formally defined and agreed upon by all Commanders-in-Chief (CINCs) and Services. These requirements may be prioritized for implementation.
- *Exercise Site Requirements.* A set of requirements determined by those sites employing the JTC to support computer-aided exercises. These requirements may be distinct from CINC/Service requirements.
- *Existing Simulations, AIS, Test Tools.* JTC constituent simulations, the ALSP Infrastructure Software (AIS) and test-support software.
- *ICDs, AIDs, Enumerations, OPSPEC, Tech Spec.* These documents collectively provide the system specification for the JTC.
  - *Interface Control Document (ICD):* provides the specification proper—defining message syntax, contents and context to enable a *functional interface*[10] between simulations (see, for example, Sonalysts Inc. [1995a]).

---

[10] The principal organizing framework for the JTC specification, e.g., air-to-ground combat, ship-to-ship combat, sustainment, and so forth.

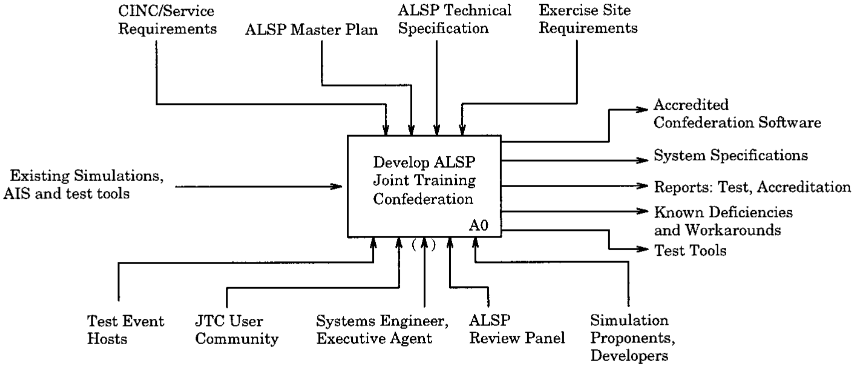

Fig. 4. ALSP JTC development process: Inputs, controls, and mechanisms.

  - —*Actor Implementation Document (AID):* defines the portions of one or more ICD(s) a particular simulation (actor) has implemented (see, for example Sonalysts Inc. [1995b]).
  - —*Enumerations:* the set of valid enumerated attribute values.
  - —*Operational Specification (OPSPEC):* a compendium document describing the JTC interfaces, i.e., objects, attributes, interactions and ownership.
  - —*Technical Specification (Tech Spec):* the technical reference manual for the ALSP protocols.
- —*Reports: Test, Accreditation.* Test reports are generated for each test event. The level of formality for these reports varies according to type of test and current needs. The Accreditation Report describes the confederation, the results of the Confederation Test, and provides the official sanctioning of the JTC (by the ALSP Review Panel) to the user community.
- —*Known Deficiencies.* The list of deficiencies and operational workarounds for the JTC.

3.2 ALSP JTC Development Process

The JTC development process is defined at the highest level as illustrated in Figure 4. Inputs into this process include the constituent simulations, ALSP Infrastructure Software, and test-support software. Outputs are an accredited confederation (new versions of software) and associated documentation (system specifications, test plans, and so forth) including a list of all known JTC deficiencies. Controls include the established CINC/Service requirements, the ALSP Master Plan and Technical Specification, and specific exercise site requirements. The means for executing this activity are identified as: the ALSP Review Panel, Executive Agent, Systems Engineer, user community, simulation proponents and developers, and the host sites for the various test activities. Note that the Interface Working Group is not explicitly referenced as a mechanism in Figure 4, but is denoted through its components. Note also that the Executive Agent and Systems Engineer are *tunneled* mechanisms. This indicates their pervasiveness throughout all activities in the process.

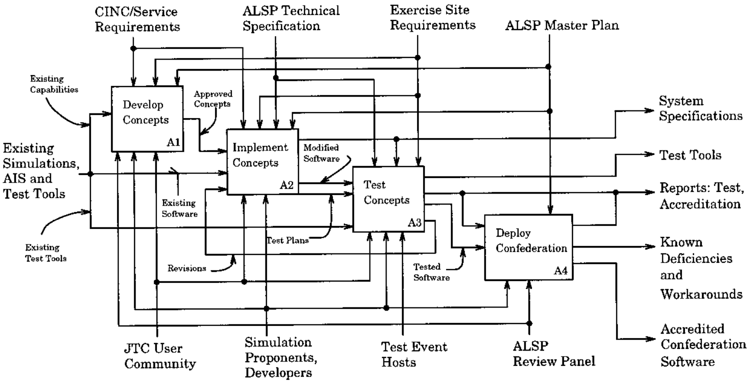

Fig. 5. ALSP JTC development process: Fundamental activities.

The process shown in Figure 4 is decomposed into four fundamental activities as depicted in Figure 5: (1) concept development, (2) concept implementation, (3) concept test, and (4) confederation deployment. The JTC development process follows a spiral methodology (see Boehm [1986]) where one revolution occurs each year. The intrarevolution activities depicted in Figure 5 are essentially sequential—culminating, hopefully, in an accredited confederation.

3.2.1 *Decomposition of Concept Development (A1).* The concept development activity consists of: concept creation, review, and approval as illustrated in Figure 6. Broadly stated, a *concept* is a desired capability for the JTC. A *Concept Paper* serves as the formal proposal for concept implementation. Typically a Concept Paper includes: (1) an indication of the relevant JTC requirements; (2) a proposed design; (3) an estimation of the impact on the existing JTC functionality, i.e., a risk assessment; and (4) an estimation of the level of development effort required for each JTC constituent.

These stages of the development process provide the earliest opportunity for V&V within the JTC. Throughout creation and development, concepts are evaluated with respect to the prioritized JTC requirements, existing simulation capabilities and available funding, as well as the ALSP Master Plan. Concept evaluation is formalized through written decision papers prepared by the Systems Engineer. Approval of concepts occurs through endorsement of the ALSP Review Panel.

3.2.2 *Decomposition of Concept Implementation (A2).* Concept implementation follows the path illustrated in Figure 7, transforming an *approved* concept into modified simulation software. The products resulting from this process include a high-level design (represented in the Operational Specification Document), interface design (the Interface Control Document), detailed design (the Actor Implementation Document) and finally, modified simulation software. Test planning and enumerations development occur throughout this process.

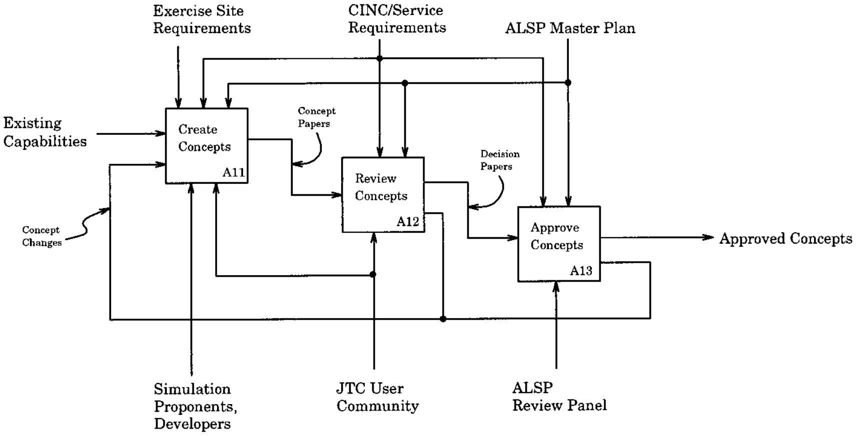

Fig. 6. Concept development activities.

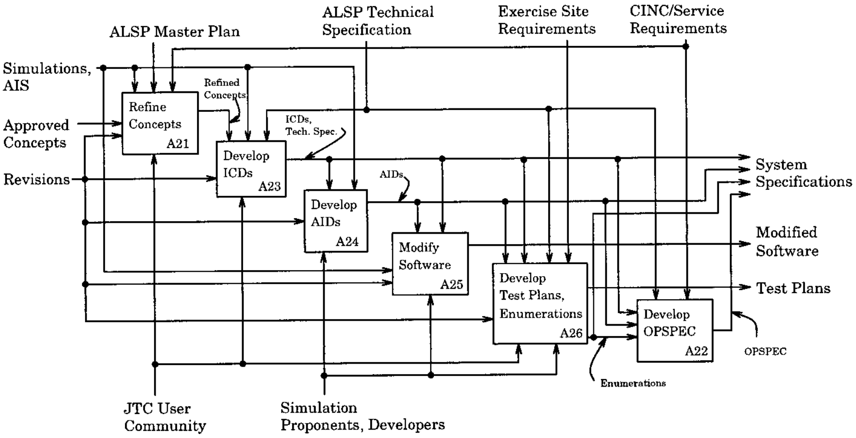

Fig. 7. Concept implementation activities.

V&V activities within the concept implementation phase are pervasive but not rigidly structured. Refinement of concepts balances development with the JTC requirements, simulation capabilities and development funds, and the ALSP Master Plan. As a concept progresses through design, the details of interfaces are captured in the system specification documents. These documents are created and reviewed by committee. Review methods range from formal structured walkthroughs to informal briefs. The level of formality is generally commensurate with the priority and/or novelty of the concept as well as the estimated risk associated with integration of the concept with existing JTC capabilities.

The documents (OPSPEC, ICDs, AIDs) are used by developers to guide changes in simulation software and to develop test plans. Note that implementation *within the simulations* as well as V&V of the simulations themselves is (generally) outside the scope of the JTC development process described here. In accordance with the management structure originally

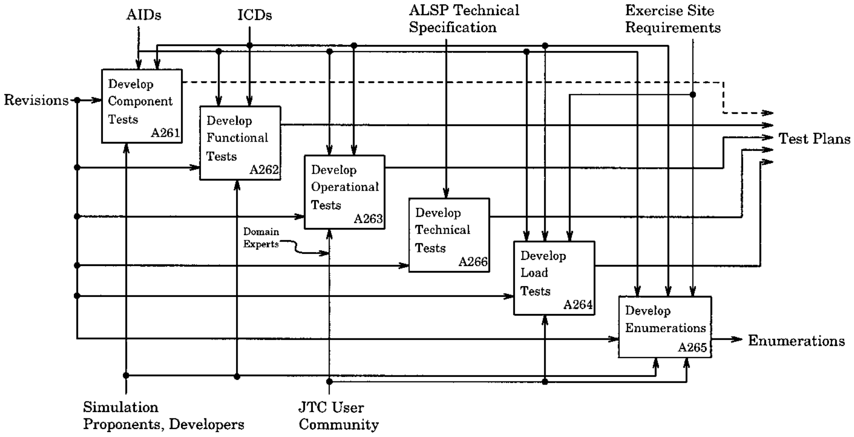

Fig. 8. Test planning and enumeration development activities.

installed for the JTC (see Fischer [1994]) and pursuant to U.S. Department of Defense [1996b], each Service has the responsibility of defining its own V&V process and naming its own V&V agent(s) for any simulation that participates within a "federation of models and simulations." The Systems Engineer assumes responsibility of the JTC as a whole, but has no dominion over the processes or agents appointed by the individual Services. The JTC is a democratic, cooperative organization. Participation within the joint training arena is ostensibly voluntary. Therefore with respect to the JTC development process and the V&V of the system as a whole, individual simulation modification must very often be treated as a black-box activity. Methods and guidance for code-level testing and regression testing can be suggested, but no mechanism for enforcement exists. Similarly, the products resulting from V&V of the constituent simulations can be requested, but not demanded. Where these products are made available, they are utilized within the process described here.

Decomposition of test plan development (A25). The test planning and enumerations development activities are depicted in Figure 8. Test plans are developed in five areas: (1) component test, (2) technical test, (3) functional test, (4) operational test, and (5) load test. Currently, component tests for individual simulations are not developed nor managed by the JTC development process. However, the JTC Systems Engineer may assist simulation developers when conducting in-house component tests.[11] The remaining test plans are formally constructed, evaluated, and executed within JTC development process.

—*Technical testing* deals with matters of conformance and compliance with the ALSP protocols. The ability of an actor to connect to the ALSP Infrastructure Software, indicates its operating parameters, advance

---

[11] Current initiatives are underway to formalize component tests at the interface level during the 1997 development cycle (see Tufarolo and Page [1996].

simulation time, and perform checkpointing are examples of items covered under technical testing.

- —*Functional testing* is designed to verify that the JTC specification has been correctly implemented.
- —*Operational testing* is performed to evaluate JTC behavior against the objectives of the training community.
- —*Load testing* is performed to evaluate the performance of the JTC. To satisfy training objectives, the JTC must be capable of keeping pace with real-time over the period of a training exercise. Specifically if an exercise begins at 1200 on a given date and ends at 1100 three weeks later (according to the wallclock and calendar), the "JTC time" at end-Ex should be *very near* 1100 (on start-Ex plus three weeks). Note, however, that this does not imply that the JTC needs to meet "hard real-time" performance measures. The training audience does not interact *directly* with the JTC simulations during an exercise; interaction is through actual "go-to-war" systems.[12] Deviations ranging from several minutes to several hours from wallclock time can be tolerated (in specific situations and for limited duration) as long as output to these systems can be otherwise mediated.

Technical and functional test plans support both the All-Actor Integration and Confederation Test (see Section 3.2.3). In addition to functional and technical test plans, operational test and load test plans are developed to guide testing at the Confederation Test.

Developing enumerations is a critical item for the JTC. Enumerations are lists of valid attribute values agreed upon by the ALSP Interface Working Group for use within the JTC. The enumerations document captures these approved values, and is used as a reference for developers and exercise managers.[13]

3.2.3 *Decomposition of Concept Testing (A3).* Concept testing is driven by the test plans developed during concept implementation, and is composed of four activities shown in Figure 9: (1) component testing, (2) Functional Interface Integration (FII), (3) All-Actor Integration (AAI), and (4) Confederation Test (CT).

As indicated in Section 3.2.2, each Service defines the V&V processes to be used by, and designates the V&V agent(s) for, simulations that participate in confederations such as the JTC. Therefore, the structure and execution of (standalone) component tests is outside the dominion of the JTC V&V agent. Interface-level component tests *are* subject to oversight within the JTC development process and the Systems Engineer assists simulation developers when conducting these in-house interface-level component tests.

---

[12] As indicated in Section 2.4, where interoperability between the JTC simulations and extant go-to-war systems is lacking, a layer of intervening personnel is utilized.

[13] Beginning in 1996, the enumerations document is subsumed by the OPSPEC.

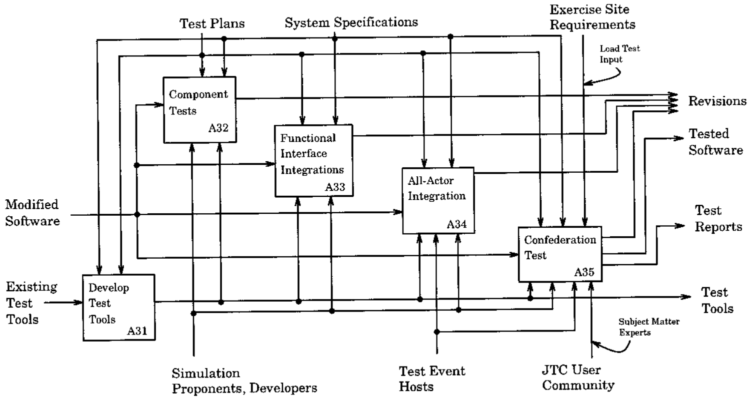

Fig. 9. Test activities.

Functional Interface Integration tests are conducted *as needed* between subsets of the JTC simulations. These integration tests are analogous to those used in modular program development and facilitate Confederation composition. Typically these subsets are formed according to functional interface—hence the moniker for the test. For example, if substantive changes occur to the air-to-ground combat interface between JTC development cycles, the air model(s) and the ground model(s) typically will participate in a Functional Interface Integration prior to the all-actor test events (AAI and CT).

The All-Actor Integration and Confederation Test are the primary test events for the JTC. These events are typically two weeks in duration and require that all simulations be convened in a single location for testing.[14] The All-Actor Integration serves as the "developers test" and focuses on evaluation of JTC functionality with respect to the specification (functional testing). The Confederation Test serves as the "user's acceptance test" and focuses on operational realism (operational testing and load testing). During the Confederation Test the behavior of the JTC is evaluated with respect to the relevant training objectives.

Structure and execution of the Confederation Test. The structure of the CT merits some discussion. The Executive Agent acts as CT Test Director.

---

[14] This point may raise an eyebrow. Centralized testing seems contrary to *distributed* simulation. If the operating environment is distributed, why should not the testing environment be likewise distributed? In fact, distributed testing *does* occur within the JTC, (FIIs are often distributed). However, centralized testing has several useful consequences. Obviously, the coordination of testing activities is simpler when all interested parties are colocated. A less obvious benefit, but a highly valuable one, is the camaraderie and sense of community that only results from close quarters interaction—a similar argument may be made for technical conferences and workshops. It is also worth noting that the primary benefit of distribution in the JTC is the distribution of training audience. In the majority of JTC exercises, "technical control" of the participating simulations is generally centralized within the simulation center hosting the exercise. Accordingly, the AAI and the CT provide the opportunity to rehearse this deployment.

The Systems Engineer assumes the overall responsibilities for test coordination and serves as director of both the Technical Test and the Load Test. The position of Functional/Operational Test Director is held by a Subject Matter Expert (SME) that represents the user community. Functional/operational testing is separated by functional area. A separate "test cell" is allocated to each functional area and the test plan for that area is executed under the direction of a Test Cell Coordinator.[15]

Two confederations are used to support testing, a *production* confederation and a *test* confederation. Each confederation is assigned a coordinator—usually a representative of the user community—who supervises and controls the technical aspects of the confederation activities, e.g., joining, resigning, scheduling and effecting confederation-wide saves, and so forth. Execution of the CT test plans occurs on the production confederation. When a test fails, a Problem Report (PR) is filed by the Test Cell Coordinator and submitted to a problem tracking system which is generally under the purview of the Systems Engineer. Problem tracking personnel determine the source of the failure and adjudicate the disposition of the PR. If software changes must be made to permit test passage, the software is modified and the test evaluated on the test confederation. Upon successful passage of the test (and any necessary regression testing as determined by the Test Confederation Coordinator) the modified software is "rolled" into production and the test is reexecuted there. A test is only considered passed when it passes in the production confederation.[16]

Validation techniques applied within the Confederation Test. Of the 13 subjective validation techniques identified by Balci [1990], *event validation, face validation, sensitivity analysis,* and *submodel testing* are the primary techniques used to evaluate the JTC. *Schellenberger's criteria* also apply to the JTC approach although the formal delineation of model assumptions is missing in the current practice.[17] In some sense, the previous year's exercises serve as *field tests* for the Confederation under test. Year-to-year changes in model functionality are rarely "drastic" and are usually localized within the overall structure of the simulation code. As a result, some level of general confidence in a model accrues over time. But demonstrated successes in actual exercises not only provide a level of confidence in the model itself, but perhaps just as importantly, they engender a level of confidence in the model developers and model operators. Generally speaking, the probability of a successful exercise is *higher* given a bad model that has competent, experienced model operators, and skilled programmers who are intimately familiar with the code, as opposed to a good model with

---

[15] For the 1996 CT, eleven test cells were utilized: air-air, air-ground, ground-air, TEL/TBM, ship/ground, cruise missile ship/air, airlift/airdrop, electronic warfare, intelligence, and sustainment.

[16] The twin confederation paradigm used during the CT is also often used within JTC exercises.

[17] See Balci [1994b] for comprehensive survey of the verification and validation techniques for discrete event simulation.

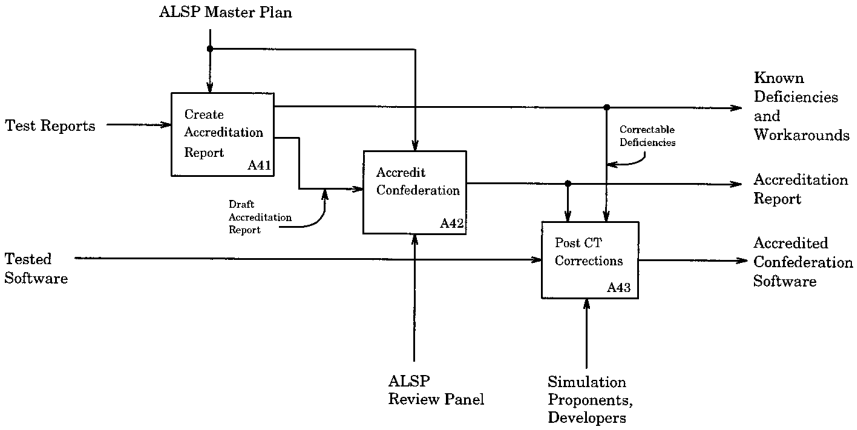

Fig. 10. Confederation deployment activities.

inexperienced operators and newbie programmers. The reasoning here is simple: no model is completely impervious to “attack” from bad or unexpected input. And at exercises, bad input happens. Simulations crash. In these situations the success of the exercise hinges on the ability of the operator(s) to recover from the crash, or the ability of the programming team to quickly develop and install the necessary patch. Some other issues affecting exercise success are discussed in Section 4.

3.2.4 *Decomposition of Confederation Deployment (A4).* Confederation deployment is the culmination of (one revolution in the spiral of) the JTC development process and is composed of three activities: (1) create Accreditation Report, (2) accredit JTC, and (3) post-CT corrections. Figure 10 illustrates these activities and their relationship.

The Accreditation Report describes the tests performed at the CT and the results obtained. The ALSP Review Panel provides the official accreditation of the JTC and authorizes the release of the Accreditation Report.

Accreditation may be dependent upon post-CT corrections. These required corrections are outlined in the Accreditation Report and are evaluated at the exercise site prior to the first use of the JTC.

4. EVALUATION

An evaluation of the VV&A process within the Joint Training Confederation must begin with a recognition that no singular “VV&A Agent” is defined for the JTC (see Fischer [1994]). Such an agent would require the omnipotence to establish and oversee the VV&A within the individual simulations as well as the VV&A of the confederation as a whole. In the current JTC management structure, each Service (or model proponent) has the responsibility of V&V for its simulation. The ALSP Systems Engineer assumes responsibility of V&V of the system as a whole. The Review Panel assumes the role of accrediting authority. This arrangement is consistent with U.S. Department of Defense [1996b]. But as Balci [1994a] indicates, a collection of valid subcomponents does not necessarily result in a valid system when these components are integrated. Therefore validation of the whole requires validation beyond that of the parts. However the limited purview of the V&V agent for a confederation of models as implied by the DoD Instruction means that most of the testing undertaken to support the system-level validation activity is necessarily functional or *black-box.* Is this a "show stopper?" Not necessarily. However, this arrangement may result in increased costs for system evaluation—assuming multiple black-box tests are needed to establish system characteristics that, alternatively, might be accomplished by simple code inspection.

4.1 The Vanilla Approach to Systems Testing and its Application within the JTC

Borrowing the term coined by Harel [1992], we describe the *vanilla* approach to systems testing as being comprised of four stages: (1) component test, (2) integration test, (3) system test, and (4) acceptance test (assuming the existence of a customer for the system). The strength of this framework is that, for the majority of software systems, it provides the recipe for cost-effective development. Of course, there are a variety of ways to tailor various aspects of this approach, pair-wise integration versus big-bang integration, for example (see Beizer [1984]). But, generally, the most cost-effective approach to system testing will reflect this framework, and recognition of this fact influenced the formulation of the JTC development process. But the addition of the requirement to accredit the system complicates the direct application of this traditional paradigm. That is, ideally, we would like to (annually) develop, evaluate and accredit the JTC and then field it to support the exercises that occur during the year. But a variety of factors stand in the way of this. A few of these are reviewed below.

The test system and fielded system differ in scale. The systems and personnel assembled to support and participate in a JTC exercise are extensive, typically involving thousands of people and hundreds of workstations over a period that can exceed a month. In contrast, the 1996 ALSP JTC Confederation Test involved roughly 100 people for two weeks, and approximately 40 simulation workstations. Both practicality and cost limit the size and scale of JTC testing. A test cannot cost as much as an exercise. In fact, testing costs arguably must be many orders of magnitude less than an exercise. As a result, the system that is used to support training is much larger in scale than the system tested.

The test system and fielded system differ by platforms. Exercises are usually held at military simulation centers.[18] Each center differs with regard to hardware, software, and network infrastructure. Therefore, test

---

[18] For 1996, RSOI, Yama Sakura, and Ulchi Focus Lens were hosted by the Korean Battle Simulation Center, Seoul, Korea; Prairie Warrior was hosted by the National Simulation Center, Ft. Leavenworth, KS; and the United Endeavor exercises were hosted at the Joint Training and Analysis Simulation Center, Suffolk, VA.

results are only meaningful (for certain measures, e.g., performance) with respect to a particular infrastructure (the one used for the test) and, of course, subject to the scale problem noted above.

The test system and fielded system differ by composition. The ALSP JTC may be viewed as a layered architecture. At the hub are a set of simulations that exchange information and coordinate the advance of simulation time over ALSP. Many of these simulations are interactive, i.e., they have user input terminals and battlefield situational displays. These devices are generally referred to as "controller workstations" and provide a layer around the simulation hub. An additional layer of systems (referred to in the JTC community as peripheral software systems or PSS) are routinely attached to the JTC when employed at an exercise. These systems are influenced by actions and changes made within the JTC, and facilitate information exchange with the training audience. The training audience generally does not interact directly with workstations or PSS, but is (ideally) immersed in a wartime-equivalent environment. In some instances, the PSS provide information in a facsimile of real-world formats. A very small number of the PSS support linking real-world command and control (C^2) systems and command, control, communications, computers and intelligence (C^4I) systems to the JTC. More often the linkage requires a manual interface (a "sneaker-net") to convey information between the JTC and the training audience. The various workstations, PSS, C^2/C^4I systems, and manual interface mechanisms provide the outer (external systems) layer of the JTC architecture. During a CT, a system consisting of the JTC and a collection of external systems anticipated for the coming year's training exercises is tested. Unfortunately, the tested system is *never* identical to the one used at any exercise. Subsets of the simulations and external systems are selected, often based on political or cost rationale that supersede technical considerations. New external systems are occasionally used in exercises (to satisfy exercise requirements) that were not tested during a CT.

The test system and fielded system differ by code. The JTC development can be viewed as multilevel development life cycle. Each simulation component comprising the JTC has a separate life cycle (necessary to satisfy primary Service requirements) in addition to that of the JTC. These individual development life cycles must be coordinated to coincide with the JTC life cycle. Currently no formal mechanism exists to facilitate this coordination. Modifications to software can and do occur throughout the year (as dictated by the primary Service development cycle). By organizational agreement, configuration management authority is given to the Service.

The test system and fielded system differ by data. Many military simulations, including those in the JTC, are essentially data driven, i.e., the outcome of any particular event in the model is sensitive (at varying levels of sensitivity) to values in the model database.[19] For the JTC simulations these databases include information regarding the types of battlefield equipment, order-of-battle (hierarchical) information, and details of the geographical region being played. These databases differ greatly between exercises. Exercise Prairie Warrior, for example, includes next-generation (2010) weapons systems in a fictionalized European scenario; whereas Ulchi Focus Lens uses current weapon systems in a Korean defense scenario. The results of any test are tightly coupled to the collection of databases underlying the simulations. As a result, database tests must be conducted prior to each exercise. However, these tests also fail to be definitive. Database parameters may be changed during an exercise, typically to correct an inappropriate action observed by exercise controllers, or to ensure the exercise scenario evolves as required to meet training objectives.

The test system and fielded system differ by personnel. Several interfaces between the training audience and the JTC simulations have been automated (i.e., an electronic link exists between C^2/C^4I systems and the JTC simulations). Many of them have not. This is the primary role of an exercise controller: to communicate with the training audience via real world mechanisms, and interact with the simulations using controller workstations. There is also a level of critical personnel generally referred to as *technical control.* These personnel operate the simulations, the ALSP infrastructure software, and the attendant systems and networks. Both the exercise controllers and technical personnel vary from exercise to exercise. Each can introduce mistakes that unintentionally disrupt the exercise. Novice workstation operators often input unexpected values, producing equally unexpected events in the simulations. System and network personnel will adjust systems and networks differently, resulting in varying behavioral characteristics.

System behaviors are subject to evaluation and modification. Referees in the white cell (see Section 2.4) provide a mechanism for fault tolerance within a training exercise. If an undesired outcome (in terms of training objectives) is detected in the state of the simulation, the exercise white cell can intercede and realign the simulated conditions to better meet the objectives. For example, during a recent exercise, an initial engagement involved a massive cruise missile attack by Opposing Forces (OPFOR) against friendly forces. As a result of operator errors during exercise set-up, many friendly forces did not have fire control for their air defenses. These units were virtually wiped out during the opening attack. Such an imbalance of forces threatened to seriously impact the training objectives for the exercise, so the referees interceded and the friendly force structure was replenished to the expected post-attack level.

---

[19] Davis [1992] provides a discussion of the rationale behind this data-centric design.

4.2 So Why Do Bad Things Happen At Exercises?

In “traditional” settings (i.e., simulation used for analysis) failure of the simulation to produce useful results is essentially a function of simulation invalidity. This is not necessarily the case in the interactive training simulation world. Numerous problems can be traced to causes outside the simulations themselves, including computer failure, network failure, operator, and controller error. To a lesser extent, general software errors occur. Consequently, the items that a substantive portion of our testing efforts focus on, and the corresponding results that lead us to consider the simulations as “sufficiently valid” are frequently not contributing to exercise problems. Although the testing efforts are worthwhile and successful test results desirable, the downside is we end up with a confederation of valid simulations which can fail to support an exercise! For example, in the early days of a recent exercise the network performance began to degrade to the point that the JTC was running so slow (at a ratio of 0.4:1 with real time) that the training audience was becoming affected. After several hours of investigation the problem was tracked to the fact that a workstation had been added to the network that morning and given the same IP address as one of the mainframes hosting the simulations. Of course, this should not happen. A fairly stringent process was in place regarding machine connection to exercise network. But the process was inadvertently subverted and the exercise was jeopardized. How VV&A can (or should) address phenomena like this is an open issue. The important point is that in “traditional” settings, establishing the accuracy of the representation with respect to the objectives is most (if not all) of the battle.[20] In an interactive simulation environment, this is not the case.

4.3 Is A Traditional Approach Infeasible?

The accumulation of factors noted above begs the question: to what extent can model validity be determined during acceptance testing? The answer seems to be a little, but only partially. The differences in scale, composition, data, and personnel demand that much of the validation activity may only be undertaken within the immediate context of an exercise—using the fielded training system, data and personnel as the basis for testing. If this is true, then is there any value in conducting the traditional (component, integration, system, acceptance) tests? Should these tests be abandoned, and all test expenditures redistributed to preexercise efforts? Such course of action would seem unwise. Among other factors, significantly extending the duration of what are already very lengthy events could have a negative impact on the morale of both the training audience and the ancillary support personnel. Such an impact could lessen the value of the training experience. From a technical standpoint, the benefits gleaned from a disciplined, bottom-up testing approach extend beyond validation. They

---

[20] Type I error, or “model builder’s risk,” exists in any modeling situation. This is the risk where results from a valid simulation are dismissed by the decision maker [Balci 1994b].

contribute to early error detection as well as verification, and provide feedback regarding the overall reliability of the system. The most cost-effective solution would seem to lie somewhere in the middle, as it often does; its precise location only identifiable through experimentation.

In the spirit of this evolution, the process described in Section 3 will be modified for the 1997 JTC development cycle. The report resulting from the Confederation Test will no longer be titled "Accreditation Report." Subsequent to the CT, the Review Panel will endorse (or accept), rather than accredit, the JTC. Accreditation authority will shift to the exercise sites.[21] JTC validation activity will shift toward an exercise-centric evaluation. Several initiatives currently underway to support exercise-centric validation are described in Tufarolo and Page [1996].

4.4 A Comparison with the Distributed Interactive Simulation Approach

The Distributed Interactive Simulation (DIS) protocol is the immediate progeny of SIMNET (refer to Section 2.3). DIS is used to support the interaction of entity-level simulations. The target training audience for a DIS exercise is typically at lower echelons than those supported by ALSP, and the training audience generally interacts directly with the simulation. The primary application of DIS is the Synthetic Theater of War (STOW) family of experiments.

A nine-step process for VV&A of a DIS exercise has been defined (see Defense Modeling and Simulation Office [1996]). The steps identified are:

(1) *Develop VV&A Plans.* VV&A planning begins at the earliest stages of DIS exercise planning and development.

(2) *Verify Standards.* Proposed components are tested for compliance to DIS protocols.

(3) *Perform Conceptual Model Validation.* The conceptual model is validated against exercise requirements.

(4) *Perform Architectural Design Verification.* The exercise architecture is evaluated for correctness and completeness.

(5) *Perform Detailed Design Validation.* Validation at this stage ensures that detailed design is correct and complete and maintains traceability to the requirements.

(6) *Perform Exercise Validation.* Examines the degree to which the DIS exercise configuration sufficiently represents the behavior, appearance, performance, fidelity constraints, and interoperability necessary for the application.

(7) *Perform Accreditation.* The V&V conducted for the exercise is reviewed by the accrediting authority.

(8) *Prepare VV&A Reports.* Results are documented and archived.

---

[21] It has been the common practice at several exercise sites for the past few tears to produce an Accreditation Report prior to each exercise.

The DIS nine-step process and the JTC process share more similarities than differences. Arguably, they differ only on the relative periphery. However, the difficulty in comparing the DIS nine-step process directly to the JTC process is that the former is an *abstraction* and the degree to which it may be successfully applied depends on a variety of application-specific factors, including scale, time and cost. However, as a paradigm for VV&A, the DIS nine-step process is well structured and should prove valuable within the STOW effort.

5. A NOD TO THE FUTURE

Both ALSP and the JTC are nearing their respective ends of service. DMSO has sponsored the definition and development of the High Level Architecture (HLA) for M&S. The HLA has been defined to "facilitate the interoperability among simulations and promote reuse of their components [Defense Modeling and Simulation Office 1995, 1]." In a recent memorandum signed by the U.S. Undersecretary of Defense for Acquisition and Technology, Paul Kaminski, the HLA has been endorsed as the standard for all U.S. DoD M&S [U.S. Department of Defense 1996a]. The HLA standard supersedes both ALSP and DIS and all DoD M&S must comply with the HLA, receive a waiver, or be retired by 2001.

The JTC has served as the primary joint, constructive training environment since 1992. However, along with the cost savings realized through the use of existing models, there are also numerous inefficiencies and limitations. While some of these problems can be addressed within the context of the existing systems, others may only be overcome using new approaches and technologies. Accordingly, the JTC is scheduled to be replaced in 1999 by the Joint Simulation System (JSIMS). The transition from the JTC to JSIMS is described in Griffin et al. [1997].

With respect to VV&A, JSIMS and the HLA—with its requirement for a common model representation in the form of a Federation Object Model (FOM)—provides the opportunity for significant cost savings when compared to those associated with integrating disparately conceived, designed and documented legacy systems. Also expected with HLA and JSIMS is a set of completely automated C^4I interfaces, thus removing much of the need for support personnel (controllers) during training exercises. But with HLA and JSIMS also come new challenges for VV&A. Inadequacies in the legacy simulations of the JTC may be numerous, but they are for the most part known quantities. The *historical* validity of the legacy systems will be lost with the new models in JSIMS. Additionally, the time flow mechanism proposed for the HLA Runtime Infrastructure (RTI) is significantly more flexible than that for either DIS or ALSP. The RTI TFM provides the ability to arbitrarily mix and dynamically adjust the temporal relevance and transmission reliability of data across the interconnection network. Adopting the same design philosophy that undergirds a variety of systems, the RTI "has not been designed to preclude abuse [Weatherly 1995]." This is unquestionably a sound design decision given the unclear potential of, and possible uses for, the new technology. However, the impact of arbitrary and dynamic causal relationships on the model validation process is difficult to conceive.

6. CONCLUSIONS

> *When I ask you to take an aspirin, please don't take the whole bottle.*
> HARVEY PENICK, *The Little Red Book*

According to the adage, two topics best avoided when making pleasant conversation are religion and politics. Such wisdom may soon apply to verification and validation—if it does not already. V&V has, unfortunately, attained *buzzword* status. Almost everyone refers to it, most everyone has an opinion about it, but few have an appreciation for its purpose, value and limitations. V&V is misunderstood at all levels: To some managers, V&V is a panacea for the rising costs of software. To some developers, V&V looks like another means to further bureaucratize software development, giving management a *raison d'etre* and placing even more roadblocks in the way of doing the really important (and interesting) work of actually producing software. To some academics, V&V is an avenue for publication and funding through the introduction of yet another "methodology"—without regard to its practicality.

Each of the above perceptions is erroneous, of course. The need for V&V in software systems development *does* have a legitimate basis. As Lewis [1992] points out, software errors have postponed Space Shuttle launches, scrambled the Strategic Air Command, snarled rail and commuter traffic, disrupted telephone networks (see Lee [1991]), and contributed to the loss of the occasional satellite. V&V has both purpose and value. Unfortunately, its application has often been either haphazard and superficial—resulting in a flawed product, and raising questions regarding the value of V&V. Or perhaps just as harmfully, V&V has been overly burdensome—increasing development time and costs disproportionately to the benefits gleaned. Harvey Penick's admonition is too often ignored.

Within the JTC development process, cost-effectiveness serves as the overriding objective. While a variety of factors impede the direct application of "traditional" VV&A processes and techniques, a VV&A process has evolved (and continues to evolve) within the JTC that seems both appropriate and cost-effective. The track record of the JTC as a training vehicle serves as witness to this. Over the past several years, the size (in numbers of actors) and the complexity (in terms of objects and behaviors represented) of the JTC has increased annually while the time and money allotted to testing has remained essentially fixed. The annual JTC exercises, while not without their technical and operational difficulties, have each achieved high levels of success. Hopefully the lessons learned from ALSP and the JTC will serve JSIMS and HLA well. Still several fundamental VV&A challenges remain. For example:

—Many of the classical simulation model validation techniques are statistically based; Balci [1990] identifies 19 such techniques. The application of these techniques requires that the system being modeled is completely observable and data from that system can be collected for comparison with model output. For training simulations this type of model output is generally not definable. Subjective validation techniques, primarily face validation, are used to characterize model validity. However, the opinion of subject matter experts has substantive credibility only in the context of "normal" operating conditions. The evaluation of rare events and other boundary-type conditions remains a significant challenge. Knepell and Arangno [1993] describe a broad framework for the validation of interactive simulations, but the introduction of mathematical and statistical rigor to training environments is an open problem.

—Each use of a training simulation must be evaluated to detect the possibility of *negative* training. Defining the conditions under which such negative training can occur is not a straightforward task. Even greater is the difficulty of actually observing these negative training conditions. Hopkinson and Sepúlveda [1995] describe a case-based analyzer for real-time evaluation of JANUS experiments. The applicability of this approach to large-scale interactive environments merits investigation.

—Performance evaluation represents a significant challenge. A major computer-aided training exercise is a *very* large endeavor, and the costs—while less than those of a live exercise—are significant.[22] Reality dictates that the cost of testing must lie well below the cost of an exercise. As a result, the scale of the Confederation Test is significantly smaller than an actual exercise. Therefore, an exact solution to the performance of the JTC cannot be generated during test. Performance models must be constructed to relate the performance of the tested configuration to any proposed exercise configuration. However, resource limitations, security considerations and other factors often preclude extensive instrumentation and data collection during an actual exercise. Constructing accurate models in the presence of very limited data is difficult.

Research is ongoing in the context of ALSP and the JTC to address many of these open problems, and to identify new, or improved, ways of achieving cost-effective VV&A.

ACKNOWLEDGMENTS

The authors thank Osman Balci for his insights and expertise in the area of VV&A that helped shape the current direction of the VV&A process for the ALSP JTC, and provide the foundations for this paper. The comments and suggestions of the anonymous referees are also gratefully acknowledged.

---

[22] Voss [1993, p. 31] reports that the cost for REFORMER in 1988 (a live exercise) was estimated at $53.9 million and in 1992 (as an ALSP-supported exercise) at $19.5 million.

REFERENCES

Alluisi, E. A. 1991. The development of technology for collective training: Simnet, a case history. In *A Revolution in Simulation: Distributed Interaction in the '90s and Beyond.* L. D. Voss Ed. Pasha Publications, Inc., Arlington, VA.

Balci, O. 1990. Guidelines for successful simulation studies. In *Proceedings of the 1990 Winter Simulation Conference* (New Orleans, LA, Dec. 9–12), 25–32.

Balci, O. 1994a. Principles of simulation model validation, verification, and testing. In *The Handbook of Simulation,* J. Banks Ed. John Wiley and Sons, New York, NY. To appear.

Balci, O. 1994b. Validation, verification, and testing techniques throughout the life cycle of a simulation study. *Ann. Oper. Res. 53,* 121–173.

Banks, J., Carson, J. S., II, and Nelson, B. L. 1996. *Discrete Event System Simulation,* second ed. Prentice Hall, Upper Saddle River, NJ.

Beizer, B. 1984. *Software System Testing and Quality Assurance.* Van Nostrand Reinhold, New York, NY.

Blais, C. 1994. Marine air ground task force (magtf) tactical warfare simulation. In *Proceedings of the 1994 Winter Simulation Conference* (Orlando, FL, Dec. 11–14), 839–844.

Blais, C. 1995. Scalability issues in enhancement of the magtf tactical warfare simulation system. In *Proceedings of the 1995 Winter Simulation Conference* (Arlington, VA, Dec. 3–6), 1280–1287.

Boehm, B. 1986. A spiral model of software development and enhancement. *ACM Softw. Eng. Notes 11,* 4, 14–24.

Bryant, R. E. 1977. Simulation of packet communications architecture computer systems. Tech. Rep. MIT-LCS-TR-188, Massachusetts Institute of Technology.

Chandy, K. M. and Misra, J. 1979. Distributed simulation: A case study in design and verification of distributed programs. *IEEE Trans. Softw. Eng. SE-5,* 5 (Sept.), 440–452.

Chandy, K. M. and Misra, J. 1981. Asynchronous distributed simulation via a sequence of parallel computations. *Commun. ACM 24,* 11 (Nov.), 198–206.

Davis, P. K. 1992. Generalizing concepts and methods of verification, validation, and accreditation (vv&a) for military simulations. Tech. Rep. R-4249-ACQ, The RAND Corporation, Santa Monica, CA.

Defense Modeling and Simulation Office. 1995. *High Level Architecture for Modeling and Simulation Management Plan.* Defense Modeling and Simulation Office. Version 1.6.

Defense Modeling and Simulation Office. 1996. *Verification, Validation and Accreditation (VV&A) Recommended Practices Guide.* Defense Modeling and Simulation Office.

Fischer, M. C. 1994. Aggregate level simulation protocol (alsp) managing confederation development. In *Proceedings of the 1994 Winter Simulation Conference* (Orlando, FL, Dec. 11–14), 775–780.

Griffin, S. P., Page, E. H., Furness, C. Z., and Fischer, M. C. 1997. Providing uninterrupted training to the joint training confederation (jtc) audience during the transition to the high level architecture (hla). In *Proceedings of the 1997 Simulation Technology and Training Conference* (Canberra, Australia, Mar. 17–20). To appear.

Harel, D. 1992. Biting the silver bullet: Toward a brighter future for system development. *IEEE Computer 25,* 1 (Jan.), 8–20.

Hopkinson, W. C. and Sepúlveda, J. A. 1995. Real time validation of man-in-the-loop simulations. In *Proceedings of the 1995 Winter Simulation Conference* (Arlington, VA, Dec. 3–6), 1250–1256.

Jefferson, D. R. 1983. Virtual time. In *Proceedings of the 1983 International Conference on Parallel Processing* (Aug. 23–26), IEEE, 384–394.

Jefferson, D. R. and Sowizral, H. A. 1982. Fast concurrent simulation using the time warp mechanism, part i: Local control. Tech. Rep., The RAND Corporation, Santa Monica, CA.

Jefferson, D. R. and Sowizral, H. A. 1983. Fast concurrent simulation using the time warp mechanism, part i: Global control. Tech. Rep., The RAND Corporation, Santa Monica, CA.

KIVIAT, P. J. 1967. Digital computer simulation: Modeling concepts. Tech. Rep. Memo RM-5378-PR (Jan.), The RAND Corporation, Santa Monica, CA.

KNEPELL, P. L. AND ARANGNO, D. C. 1993. *Simulation Validation: A Confidence Assessment Methodology.* IEEE Computer Society Press, Los Alamitos, Calif.

LAW, A. M. AND KELTON, W. D. 1991. *Simulation Modeling and Analysis,* second ed. McGraw-Hill, New York, NY.

LEE, L. 1991. *The Day the Phones Stopped.* Donald L. Fine, Inc.

LEWIS, R. O. 1992. *Independent Verification and Validation: A Life Cycle Engineering Process for Quality Software.* John Wiley and Sons, New York, NY.

MERTENS, S. 1993. The corps battle simulation for military training. In *Proceedings of the 1993 Winter Simulation Conference* (Los Angeles, CA, Dec. 12–15), 1053–1056.

MILLER, G. AND ZABEK, A. A. 1996. The joint training confederation and the aggregate level simulation protocol. *MORS Phalanx 29,* 24–27.

NANCE, R. E. 1981. The time and state relationships in simulation modeling. *Commun. ACM 24,* 4 (April), 173–179.

NANCE, R. E. 1993. A history of discrete event simulation programming languages. *ACM SIGPLAN Notices 28,* 3, HOPL II (Cambridge, MA, April 20–23), 149–175.

NANCE, R. E. 1981. On time flow mechanisms for discrete event simulations. *Manage. Sci. 18,* 1 (Sept.), 59–73.

NATIONAL INSTITUTE OF STANDARDS AND TECHNOLOGY. 1993. *Integration Definition for Function Modeling (IDEF0).* National Institute of Standards and Technology. Federal Information Processing Standards Publication 183.

NATIONAL SIMULATION CENTER. 1994. *Training with Simulations: Handbook for Commanders and Trainers.* National Simulation Center.

PACE, D. K. 1993. Naval modeling and simulation verification, validation, and accreditation. In *Proceedings of the 1993 Winter Simulation Conference* (Los Angeles, CA, Dec. 12–15), 1077–1080.

PEGDEN, C. D., SHANNON, R. E., AND SADOWSKI, R. P. 1990. *Introduction to Simulation Using SIMAN.* McGraw-Hill, New York, NY.

SARGENT, R. G. 1992. Validation and verification of simulation models. In *Proceedings of the 1992 Winter Simulation Conference* (Arlington, VA, Dec. 13–16), 104–114.

SIKORA, J. AND COOSE, P. 1995. What in the world is ads? *MORS Phalanx 28,* 2 (June), 1–8.

SIKORA, J. AND WILLIAMS, M. L. 1994. Workshop report: Simval '94. *MORS Phalanx 27,* 4 (Dec.), 15–17.

SONALYSTS, INC. 1995a. *ALSP Combat Interactions Interface Control Document for the 1996 JTC.* Sonalysts, Inc.

SONALYSTS, INC. 1995b. *RESA Actor Implementation Document: 1996 JTC.* Sonalysts, Inc.

SONALYSTS, INC. 1995c. *RESA User Guide.* Sonalysts, Inc.

TAYLOR, J. G. 1983. *Lanchester Models of Warfare.* Operations Research Society of America, Arlington, VA.

TUFAROLO, J. A. AND PAGE, E. H. 1996. Evolving the vv&a process for the alsp joint training confederation. In *Proceedings of the 1996 Winter Simulation Conference* (Coronado, CA, Dec. 8–11), 952–958.

U.S. DEPARTMENT OF DEFENSE. 1996a. *DoD High Level Architecture (HLA) for Simulations.* U.S. Department of Defense. Memorandum signed by USD(A&T).

U.S. DEPARTMENT OF DEFENSE. 1996b. *DoD Modeling and Simulation (M&S) Verification, Validation, and Accreditation (VV&A).* U.S. Department of Defense. Directive 5000.61.

VOSS, L. D. 1993. *A Revolution in Simulation: Distributed Interaction in the '90s and Beyond.* Pasha Publications, Inc., Arlington, VA.

WEATHERLY, R. M. 1995. Personal communication.

WEATHERLY, R. M., WILSON, A. L., CANOVA, B. S., PAGE, E. H., ZABEK, A. A., AND FISCHER, M. C. 1996. Advanced distributed simulation through the aggregate level simulation protocol. In *Proceedings of the 29th Hawaii International Conference on Systems Sciences,* Volume 1 (Wailea, HI, Jan. 3–6), 407–415.

WEATHERLY, R. M., WILSON, A. L., AND GRIFFIN, S. P. 1993. Alsp—theory, experience, and future directions. In *Proceedings of the 1993 Winter Simulation Conference* (Los Angeles, CA, Dec. 12–15), 1068–1072.

WILSON, A. L. AND WEATHERLY, R. M. 1994. The aggregate level simulation protocol: An evolving system. In *Proceedings of the 1994 Winter Simulation Conference* (Lake Buena Vista, FL, Dec. 11–14), 781–787.

WITTMAN, R. L., JR. 1995. Case tool integration and utilization within the joint theater level simulation (jtls). In *Proceedings of the 1995 Winter Simulation Conference* (Arlington, VA, Dec. 3–6), 1147–1151.

ZABEK, A. A. 1994. Alsp update. Project Briefing, The MITRE Corporation, Not in Public Domain.

Received January 1996; revised April 1997; accepted April 1997