

# An Architecture for Integrating SCORM-Compliant Instruction with HLA-Compliant Simulation

Vikram Manikonda\*, Preetam Maloor\*, Jacqueline Haynes\*, Susan Marshall‡

\*Intelligent Automation Incorporated  
15400 Calhoun Drive, Suite 400, Rockville, MD 20855  
{vikram, pmaloor, jhaynes}@i-a-i.com

‡US Army PEO STRI  
Orlando Florida  
Susan.marshall1@us.army.mil

Keywords:

HLA, SCORM, Training, Architectures, Intelligent Instruction

**ABSTRACT:** *This paper describes the architecture of the Simulation-based Intelligent Training and Assessment (SITA) system. SITA is a prototype system that integrates HLA compliant simulation with SCORM compliant instruction to provide a more robust and realistic virtual training environment. The SITA architecture is designed to address issues related to SCORM- HLA standards interoperability, cross domain execution, code reusability and integration ease. The overall architecture and implementation details of the prototype system are presented in this paper. Air Traffic Management is adopted as the domain of instruction, and an HLA compliant Collaborative Regional Flow Control Decision Support Tool, integrated with a LMS and SCORM compliant instruction, is used to demonstrate the features of the architecture.*

## 1. Introduction

Comprehensive user training is a key part of the successful deployment and execution of any system or operation. Training provides end-users an opportunity to understand various systems aspects such as its objectives, functionality, capabilities, usability, shortcomings, and user-role, in a controlled environment under the guidance and supervision of an expert. Training also provides an opportunity to assess the expertise of the trainee/user with respect to the operation of the system.

Military training is often provided offline, (classroom tutorials, web-based instruction) in an emulated and/or simulated environment, or in a “hands-on” mode i.e. learning on the job. In several settings, learning on-the-job is a very expensive option. In a military battlefield setting, this is hardly a viable option. Training is provided by emulating a variety of possible battlefield scenarios. Given the numerous entities and dynamic interactions between these entities (internal and external) setting up a detailed training exercise is a complex and expensive task.

With the recent advances in computer simulation, the use of simulations to model and create realistic scenarios is proving to be a very cost effective option. Simulations are increasingly being used in the military and commercial settings to “prep” the trainees and help them reach a certain level of expertise before a field training exercise is performed. Recent developments in distributed simulation even provide the capability to perform team-training where trainees are geographically distributed. Simulations can be used to generate a variety of “what-if” scenarios exposing the user to numerous settings that one might encounter in the real world. Simulations also provide an elegant, cost effective method to re-orchestrate a given scenario, should retraining be needed to achieve mastery over a desired skill (remediation).

One of the successful and widely publicized efforts in simulation-based training include the Modular SemiAutomatic Forces (ModSAF) [1] program, a joint effort between the Defense Advanced Research Project Agency (DARPA) and the U. S. Army Simulation, Training and Instrumentation Command (STRICOM). The goal of ModSAF was to provide a platform within which all services could expand the synthetic battle space. ModSAF has been followed up with Joint Semi-

Automated Forces (JSAF) and One Semi-Automated Forces (OneSAF) [2] programs. These initiatives are aimed at providing an integral simulation service to the Advanced Concepts and Requirements (ACR), Training, Exercises, and Military Operations (TEMO), and Research Development and Acquisition (RDA) domains.

Although the use of simulations as a training tool has increased significantly over the last decade, there has been little progress towards developing an architecture that integrates simulation standards, such as the DoD's High Level Architecture (HLA) [3] and training infrastructures/standards such as the Sharable Content Object Reference Model (SCORM) [4].

While each of these standards, HLA and SCORM, independently strives to achieve reusability of simulations and instructional content, little research has been performed in the direction of developing reusable instructional design that integrates simulation as a part of the instruction and assessment process. In fact, typically the online instruction is conducted in isolation from the hands-on simulation effort.

In this paper we present a prototype architecture for integrating SCORM-compliant instruction with HLA-compliant simulation. Embedding HLA compliant simulation into the SCORM compliant instructional design serves three primary objectives: (i) providing a richer learning environment that enforces the instructional paradigm of being able to practice the skill following instruction; (ii) run-time assessment of learning objectives by capturing and assessing the students performance as the student or team of students interacts with the simulation, and the ability to use the assessment to tailor the instruction, and (iii) leverage compliance with standards to achieve composability of instruction modules and simulation models across different instructional design packages and simulations.

The architecture discussed in this paper is being developed as part of the Simulation-based Intelligent Training and Assessment (SITA) project. The project is funded as a Broad Area Announcement (BAA) through the Joint Advanced Distributed Learning Collaborative Laboratory (Joint ADL Co-Lab) located in Orlando, Florida. ( see <http://www.jointadlcolab.org>) The work is being conducted as a joint effort between the Joint ADL Co Lab and Intelligent Automation, Inc. of Rockville, MD.

The paper is organized as follows. In Section 2, we present the definitions of the terminology that is used in

this paper. In Section 3, we present the vision and requirements that drove the design of the architecture. In Section 4, the architectural details of SITA are presented. Section 5 presents a brief overview of the instructional design being developed for this effort. Conclusions are presented in Section 6.

## 2. Terminology and Definitions

As the architecture attempts to bridge the gap between two fairly distinct communities, in this section we provide a brief overview of some of the commonly used terminology. Readers are referred to [3, 4] for more details.

In the DOD simulation community, the HLA is a simulation standard that provides the blueprint for the integration of existing simulations and provides guidelines for creating future simulations with interoperability in mind. The architecture standard specifies a set of rules, an interface specification to a communications infrastructure called the *Run-Time Infrastructure* (RTI), and a template called the *Object Model Template* (OMT) to use for defining message types for information interchange. The RTI provides the key set of services required of a distributed simulation system. A set of simulations, or *federation*, is integrated using an HLA RTI. The interface between the individual simulations, called *federates* is called the object model, with the specific object model being used in a given federation called the *Federation Object Model* (FOM). The FOM defines the configuration of federates and the communication protocol between the federates through the RTI.

In the education, training and instructional design areas, borrowing from the work of other specification and standards bodies, ADL developed SCORM, as a set of guidelines, specification and standards for creating and deploying e-learning. SCORM aims to foster creation of reusable learning content as "instructional objects" within a common technical framework for computer and Web-based learning. SCORM 2004, the current version, describes (i) a *Content Aggregation Model* (CAM) that describes how to describe, package and define components used in a learning experience; (ii) the *Run-Time Environment* (RTE) that describes the *Learning Management System* (LMS) requirements for managing the run-time environment in terms of the communication protocol between the LMS and *Sharable Content Objects*<sup>1</sup>

---

<sup>1</sup> A SCO represents one or more assets which use the SCORM runtime environment to communicate with different Learning Management Systems and is the smallest unit that can be administered by an LMS [5].

(SCO), and the standardized data model elements used for passing information relevant to the learner's experience with the content; and (iii) *Sequencing and Navigation* (SN) that describes how SCORM-conformant content may be sequenced through a set of learner-initiated or system-initiated navigation events.

### 3. Architectural Vision and Requirements

The overall vision is to develop a framework that integrates SCORM compliant instruction with HLA compliant simulation that will enable the development of a richer instruction and training development environment, while maximizing code reuse across both domains. Several interoperability requirements have driven the current architecture design of SITA. These include:

*Embedded Training:* Embedded training is one of the key driving requirements. For the architecture to fully support the interfacing of a SCORM-compliant LMS with the HLA-compliant simulation, the LMS needs to be able to manage the simulation (start, stop etc), allow for simulation initialization data to be controlled via the LMS based on the training requirements, enable run-time exchange of information between the LMS and federation (assessment data, any run-time simulation parameter that might need to be changed based on the assessment), and possibly save and restore the simulation data.

*Cross Domain Execution:* SCORM-compliant instruction is typically developed for Web-based learning and thus executes across domains. HLA- federations typically execute within the same domain or LAN and are not developed for cross-domain execution. Embedding the simulation into the instruction poses several cross domain execution and security issues that need to be addressed by the framework.

*Run-time Assessment:* Intelligent instruction provides for the capability to gather student performance data while the instruction is being delivered, assessing the data and accordingly sequencing through the courseware contents to provide optimal instruction. Embedding simulation into the instruction requires that the LMS have the capability to gather and assess the user's understanding of the material presented. This poses challenges related to developing appropriate metrics that capture user performance and extracting the appropriate data from the simulation.

*LMS Independence:* To optimize interoperability, LMS-independence is desired. The LMS is typically a proprietary piece of software with non-standardized internal data definitions and data-storage. Only the LMS-SCO interfaces are developed to be SCORM compliant.

To maximize reuse, it is desirable that federation data, such as initialization data, simulation state, and individual trainee data be stored outside the LMS, but possibly indexed (e.g. with userid) so that it can be correlated with the LMS. Use of an external database also allows for use of generalized data storage schemes such as databases.

*Legacy Simulation Reuse:* While the architecture can impose new standards for design rules for instructional packages and simulation federates, the ability to embed already existing HLA compliant simulation into instructional design packages is of significant importance. Rewriting a legacy HLA training simulation is typically not a viable cost option. The infrastructure requires the development of design interfaces that maximize legacy code reuse.

### 4. SITA Architecture

The SITA architecture was developed as a prototype to evaluate the feasibility of developing an architecture to meet the above requirements. To demonstrate feasibility, Air Traffic Management (ATM) was adopted as the domain of instruction and the HLA compliant Collaborative Regional Flow Control (CRFC) Decision Support Tool (DST) [6] was adopted as the simulation federation. The objective of the instructional design is to train Traffic Management Coordinators (TMCs) on the ATM procedures associated with a new ATM concept called regional flow control.

The main components of the SITA architecture, consists of the Learning Management System (LMS), the CRFC-DST simulation federation the RTI-SCO interface module, Simulation Manager and the Launcher/Collector Applet (see Figure 1).

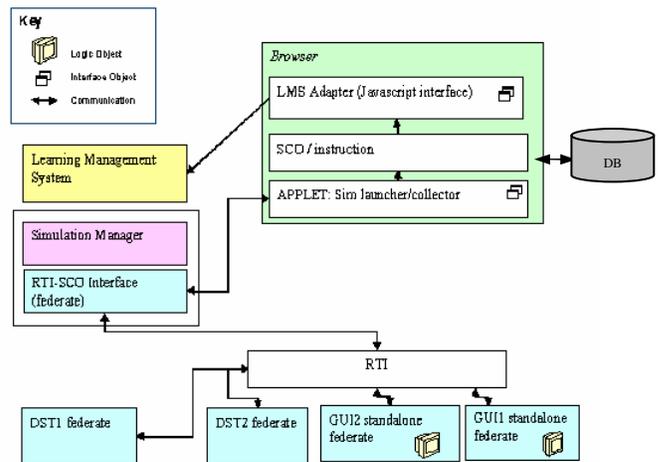


Figure 1: SITA architecture

Information flow in this infrastructure is as follows: Upon initialization, the student’s simulation settings are read from the Learning Management System (LMS) by the Sharable Content Object (SCO). The SCO then launches the simulation by sending a *start simulation* command to the *Simulation launcher/collector* applet, which runs in the same client browser context as the instructional content (SCO). The applet connects to the *Simulation Manager*. The Simulation Manager starts both the RTIexec, and the simulation federates. The GUI federates are displayed on the same client machine as the SCO (but not in the browser). Communication between simulation federates and the LMS is achieved via the *RTI-SCO Interface*, the *Simulation launcher/collector applet* and the *LMS Adapter*. The initialization, status update and termination processes in SITA are shown in Figure 2.

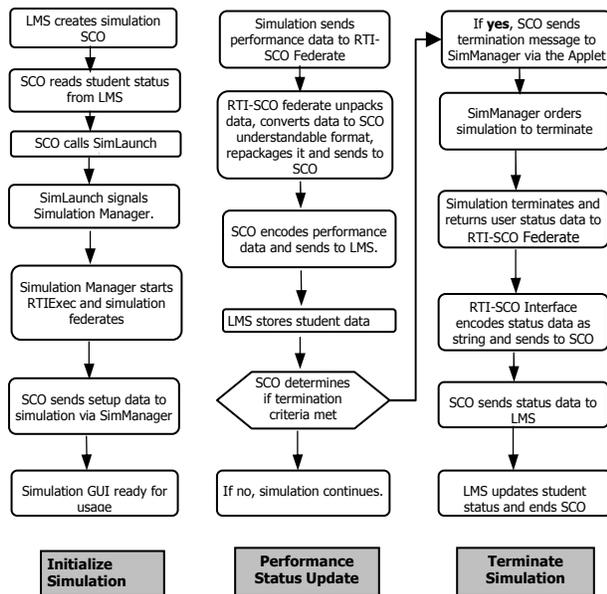


Figure 2: SITA Operation Flowchart

The design of each of the architectural components is discussed in the following sections.

#### 4.1 Learning Management System

The LMS Run Time Environment API [5] is the principle means with which the SCO performs basic simulation and instruction control. It provides a channel for simulation initialization and outcome data to be transferred to the Learning Management System. The RTE supports data models that allow defined set information about SCOs to be tracked by different LMS environments. For this effort, we selected Avilar Technologies’s WebMentor [7], a browser-based Learning Management System that conforms to the SCORM 1.2 standard.

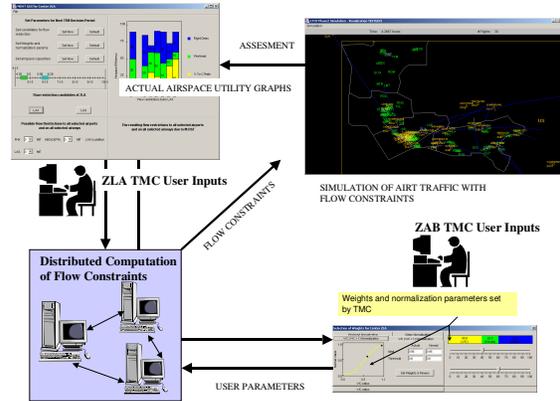


Figure 3: Operational Flow of the Collaborative Regional Flow Control Decision Support Tool

#### 4.2 HLA-Compliant Simulation Federation

As mentioned earlier the Collaborative Regional Flow Control (CRFC) Decision Support Tool (DST) [6] was adopted as the simulation Federation. In addition to being HLA compliant, CRFC is an agent-based simulation thus enabling the evaluation of intelligent instructional paradigms or team training, where an “agent” plays the roles of team-mates.

The CRFC tool takes inputs from TMCs such as jet routes and/or airports on which a flow constraint should be imposed; weights on metrics corresponding to *throughput*, *system wide delay*, and *workload*; and *duration* for which the flow constraints should be imposed and then determines optimal flow constraints across the region boundaries based on these input parameters. These flow constraints are then used to simulate traffic flow and compute the resulting efficiency of the airspace. The resulting efficiency is presented as airspace utility graphs (See Figure 3).

The DST can be used by an individual TMC (with other TMC’s simulated by autonomous agents) or in a team mode, with two or more TMCs collaborating to optimize flow across their boundaries.

The top-level CRFC-DST architecture is shown in Figure 4. At its lowest level the architecture contains a number of communications protocols, including the HLA-RTI for run-time simulation communications, SQL/XML for data initialization and storage, remote method invocation (such as SOAP or CORBA) and standard web HTTP protocols, used outside run-time for access to model and data repositories.

Above this layer, the architecture contains an agent-based simulation framework. The agent-based modeling and

simulation framework, built on IAI's Cybele™ agent infrastructure (see <http://www.opencybele.org>), provides a software layer between the models and the HLA RTI. Cybele is built on top of the Java™ 2 platform and provides the runtime environment for control and execution of agents.

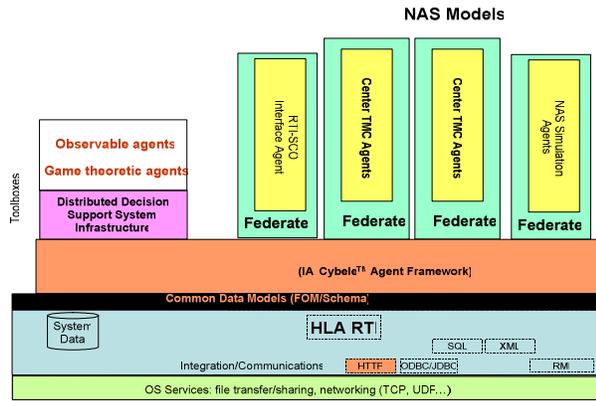


Figure 4: CRFC-DST Architecture

The Applications layer of the architecture consists of applications built on this common core infrastructure. This includes simulation federates as well as utilities such as simulation control, visualization, data collection, and analysis tools.

Integrating the Cybele infrastructure with the HLA RTI involved several steps. First, the native Cybele inter-agent communications service had to be replaced with HLA-compliant communications. Second, the Cybele timer, event management and thread management service were adapted to support time-managed, fast-time discrete event simulation and sender-side filtering. Please refer to [8] for details.

Communication between federates is primarily done using HLA interactions. Conversion between Cybele communications notions (serialized Java objects over message channels) to RTI notions (FOM objects over RTI Data Management or Data Distribution Management) is done by the infrastructure transparent to the user. The infrastructure handles complex objects by converting their complex components into XML strings. These strings are then passed as RTI attributes and are reconstituted at the receiving end.

### 4.3 RTI-SCO Interface

The RTI-SCO Interface module allows the exchange of information between the SCO and the CRFC federates. The CRFC federates are built on the CybeleHLA agent

infrastructure that uses channel spaces to communicate between agents. To operate within the Cybele agent infrastructure, the RTI-SCO interface is wrapped as a CybeleHLA agent federate. Hence, the RTI-SCO interface subscribes to the same channel space for accessing data. The data obtained from the CRFC federates, in the form of serialized objects, is unpacked, and repackaged as a Java data object in a format known to the SCO, before being transmitted to the SCO over http sockets using Java Remote Method Invocation technology [9]. In our current prototype, the main data exchanged between the CRFC-DST (simulation) and the simulation SCO via the RTI-SCO Interface federate is as follows:

- CRFC-DST to SCO: (i) Jet routes on which a flow constraint should be imposed. (ii) Airports on which flow constraints should be imposed. (iii) Weights on metrics corresponding to *throughput*, *system wide delay*, and *controller workload* used by the optimizer. (iv) Duration for which the flow constraints should be imposed and (v) Airspace efficiency metrics computed by the DST after implementing the flow constraints based on the input parameters.
- SCO to CRFC-DST: Initialization data for the simulation.

### 4.4 Simulation Manager

The simulation manager software's primary function is to control the start-up/shut down of CRFC-DST federates as well as the RTI-SCO Interface federate.

The Simulation Manager consists of two major parts: (a) A server component that is present on every machine running a federate. It controls the start/stop functionalities of its federate processes. (b) A client component that issues start/stop commands to each of these servers. Communication between the *client* and *servers* is over http sockets using Java RMI.

### 4.5 Simulation Launcher/Collector Applet

This component runs in the SCO/browser environment as a java applet and interacts with the SCO using JavaScript-Applet communication protocols. It, together with its supporting classes, carries out the following functions.

- Acts as an interface between the SCO and the Simulation Manager, to start or stop the simulation.
- Acts as an interface between the SCO and the RTI-SCO interface, to save and restore user performance and simulation state data.
- Decodes and evaluates effectiveness metrics provided by the CRFC simulation, and provides these to the SCO.

#### 4.6 LMS Adapter

The LMS adapter is a browser-based (JavaScript) interface to LMS functionality. The interface allows the SCO to communicate to the LMS using the SCORM 1.2 Run Time Environment (RTE) standard API [5]. The SCO uses this adapter to store and retrieve user-specific information on simulation state (initial, completed, error), personalization parameters, and performance data on the LMS.

#### 4.7 Database

An external database is used for storing intermediate status information about the simulation and training. This information is necessary for resuming training when the training/simulation is paused by a student.

During the architecture design, it was originally envisioned that the status data would be stored within the LMS using the available RTE 1.2 data model elements. But on further investigation we realized that the data models restrict the storage of information as strings. Hence we opted to store the status data in an external database maintained on the server which hosts the SCO/Applet. Though maintenance of an external database increases the complexity for the architecture, the flexibility provided in terms of information storage/retrieval makes it a viable alternative.

We use MySQL, an Open Source SQL database, that is developed, distributed, and supported by MySQL AB (see <http://www.mysql.com>). Currently, the database server is hosted on the same machine as the Applet server. In the near future, we intend to modify this design to host the database on a dedicated server that remotely communicates with the other components over sockets.

### 5. Instructional Design

To demonstrate the features of the architecture a SCORM compliant instructional design package is being developed (see [10] for more details). The instructional design of SITA demonstrates the use of simulation for interactive instruction, practice, and assessment. The instructions provide activities for individual and collaborative team practice in monitoring and controlling air traffic flow within a designated air space. The goal of the instructional design is to provide a motivating, game-like interactivity in the learning process. The SITA instruction SCOs are designed for achieving the following instructional goals:

1. Pre-assessment of *knowledge*, where the learner is asked specific questions about a portion of the running simulation he/she is observing, where the questions can be modified independently from the simulation (in a SCO).

2. Practice of *skills*, where the learner can use knowledge and information he/she learned through didactic instruction (instructor-directed content acquisition), or interactive instruction (instruction using multiple sequences of information presentation-student activity-feedback-next presentation). Either of these can be used individually or as part of a *team*. For both of these scenarios, the learner(s) receive *immediate feedback* in the form of specific results shown in the simulation.
3. Final assessment, where the learner's relevant knowledge (*facts*) and skills (*performance*) can be assessed within a single, cohesive training and assessment simulation, providing immediate feedback and specific, targeted remediation as necessary.

SITA uses a **decentralized approach** to student assessment. Here, assessment is carried out in two places: (i) in the CRFC-DST simulation where the airspace efficiencies resulting from user-input constraints are assessed against that from DST suggested constraints. (ii) in the SCO, using conventional assessment items such as multiple choice items, short answer items, etc. This approach gives SITA the ability to access not only whether the student has acquired specific information but also whether he/she has the ability to use factual knowledge for task performance.

Preliminary prototype testing and evaluation have demonstrated the feasibility of developing an infrastructure for integrating SCORM-compliant instruction with HLA-compliant simulation.

### 6. Conclusions and Future Work

In this paper, we presented the architecture of SITA, a prototype system that integrates HLA compliant simulation with SCORM compliant instruction to provide a more robust and realistic virtual training environment. The integrated framework development was motivated by the need for developing a richer learning environment where a student can demonstrate his or her understanding of the instructional material presented, by applying the learned information in a realistic simulation environment. Future efforts involve a further investigation into cross domain issues, network security, and API standardization to accommodate a larger class of HLA compliant simulations. Integrating new features, such as sequencing, provided by SCORM 2004 will also be investigated in future efforts.

### 7. References

- [1] Ceranowicz, A. (1994). Modular Semi-Automated Forces, *Proceedings of the 1994 Winter Simulation Conference*, pages. 755-761.

- [2] Wittman, R. L. & Harrison, C. T. (2001). OneSAF: A Product Line Approach to Simulation Development. *Technical Paper, MITRE*.
- [3] High Level Architecture  
<https://www.dmsomil/public/transition/hla/>
- [4] ADL. (2004). SCORM 2004 Overview.  
<http://www.adlnet.org>
- [5] Dodds, P. (2001). Advanced Distributed Learning, Sharable Content Object Reference Model Version 1.2, Release 2001. <http://www.adlnet.org>, Retrieved June 29, 2004
- [6] Satapathy, G., Manikonda, V., Robinson, J. and Farley, T. (2002). En-Route Sector Metering using a Game-Theoretic Approach. AAI Game Theory and Decision Theory Workshop.
- [7] Avilar Technologies, (2003).  
[http://home.avilar.com/products/wm\\_lms.html](http://home.avilar.com/products/wm_lms.html)
- [8] Aronson, J., Manikonda, V., Peng, W., Levy, R. & Roth, R. (2003). An HLA Compliant Agent-based Fast-Time Simulation Architecture for Analysis of Civil Aviation Concepts. *Spring SISO Simulation Interoperability Workshop*, April 1-4, 2003, Orlando, Florida.
- [9] Sun Microsystems, (2003). *Java Remote Method Invocation - Distributed Computing for Java*, White Paper, retrieved June 25, 2004.  
<http://java.sun.com/products/jdk/rmi/reference/whitepapers/javarmi.html>
- [10] J. Haynes, S. Marshall, V. Manikonda, P. Maloor. Enriching ADL: Integrating HLA Simulation and SCORM Instruction using SITA (Simulation-based Intelligent Tutoring System). The Interservice/Industry Training, Simulation and Education Conference (IITSEC), Orlando Florida, Dec 6 -9, 2004 (To appear)

(SITA) prototype. His research interests are in Artificial Intelligence applications in educational technology. He has several years of experience in the areas of Natural Language Processing, Speech Processing and Computer-Human Interaction. He has a Masters degree in Computer Science from Texas A&M University and a Bachelors degree in Computer Science and Engineering from the University of Bombay.

**Jacqueline Haynes** is co-founder, Executive Vice President and Director of the Education and Training Technology Group at IAI. Her background combines education and psychology with AI applications. She received her Ph.D. from the University of Maryland in Curriculum and Instruction, and did post-doctoral work there in artificial intelligence and intelligent tutoring systems. Previously she was a faculty member at the University of Maryland, College of Education. Her research interests include research-based instructional design, tools for Web-based instruction, and reading comprehension.

**Susan Marshall** is a Principal Investigator on Advanced Distributed Learning (ADL) Research for Program Executive Office Simulation, Training and Instrumentation (PEO STRI), supporting the Joint ADL Co-Lab in Orlando. Previously she has been a project engineer on several programs, including the Advanced Gunnery Training System (AGTS) and War fighting Simulation (WARSIM). Susan has twenty years of experience as a civilian engineer for the Navy and Army, mainly in Software acquisition and testing. She has a B.S. in Electrical Engineering (EE) from Marquette University, and a M.S. in Industrial Engineering (IE) in Simulation from the University of Central Florida.

## Author Biographies

**Vikram Manikonda** is a Vice President and the Director of the Distributed Intelligent Systems at IAI. He received his B.E. degree in Electrical Engineering from the Birla Institute of Technology and Science, India, in 1992, his M.S. and Ph.D. degrees, both in Electrical Engineering, from the University of Maryland at College Park, in 1994 and 1997 respectively. His research interests include intelligent control, robotics, motion description languages, multi-agent systems for modeling and simulation, and air traffic control and management.

**Preetam Maloor** is a Research Engineer in the Education and Training Technology Group IAI. He is currently involved in the design and implementation of the Simulation-based Intelligent Training and Assessment