

DESIGN & DEVELOPMENT OF A STANDARD SPACE TRAINER (SST)

**Laura R. Dietz, David S. Coleman, Dr. Winston Bennett, Jr., Dr. Robert T. Nullmeyer,
Mr. Michael E. Newman, Mr. Merrill C. Galligan**

ABSTRACT

Currently, the Air Force Space Command (AFSPC) is using a number of different training systems, simulators, and emulators for space command and control (C2) Initial Skills Training (IST), and Unit Qualification Training (UQT). Each of these training systems or simulators is “stove-piped.” That is, each one uses unique and different system-specific hardware, operating systems, networking capabilities, and costly proprietary software. In addition, each system has its own maintenance contract for sustainment. To address this problem, the Air Force established a vision for a single Standard Space Trainer (SST) platform for C2 operator training employing commercial off-the-shelf (COTS) personal computer (PC) hardware and operating systems.

This paper addresses how the Space and Missile Systems Center (SMC) and Sonalysts, Inc. are transitioning a proof-of-concept satellite operator training system developed under a Phase I and II Small Business Innovation Research (SBIR) Project, into a production (Phase III) SST. The SST proof-of-concept was developed for the SMC and Air Education and Training Command’s (AETC) 533rd Training Squadron (TRS) with assistance from the Air Force Research Laboratory (AFRL). The proof-of-concept trainer provides instructional support for two satellite systems using advanced modeling, simulation, and visualization technologies. The production SST will employ a single common training system software architecture that will launch system-specific space mission simulations to support IST, UQT, and crew proficiency training. The SST Architecture will enable space trainers to be developed that run on standard PC-based COTS hardware without reliance on the specific hardware or software of the actual space system. A Software Development Kit/Application Programming Interface package will provide a Common Architecture to allow the development and integration of new training systems within a flexible and modular plug-in framework.

ABOUT THE AUTHORS

Laura R. Dietz is an Instructional Designer with Sonalysts, Inc. She holds a bachelor’s degree in Psychology from the University of Connecticut, and has more than 9 years of experience in the design and development of advanced training systems and interactive courseware products for military training applications. Ms. Dietz is the Project Leader for the Phase III SBIR SST project for the Air Force.

David S. Coleman is an Executive Vice President and Manager of the Training Systems, Operations, and Technology Group at Sonalysts, Inc. He holds a bachelor’s degree from the U.S. Naval Academy, and has 29 years of experience in the management of advanced military training system and interactive courseware programs for the Navy, Army, and Air Force. He is the Program Manager for Sonalysts Space Training programs including the SST Program and other space training efforts.

Winston Bennett Jr., Ph.D. is a Senior Research Psychologist with the Air Force Research Laboratory, 711th Human Performance Wing, Human Effectiveness Directorate, Warfighter Readiness Research Division, in Mesa AZ. He is the Technical Advisor and team leader for continuous learning and performance assessment research and development. Dr. Bennett leads a team that oversees core research programs and has garnered over \$65M of outside funding investment for the Division. Dr. Bennett graduated from the University of Texas – San Antonio with a degree in psychology in 1982. He received a master’s degree in industrial psychology from St. Mary’s University in 1985. He received his Ph.D. in Industrial Organizational Psychology from Texas A&M University in 1995.

Robert T. Nullmeyer, Ph.D. is a research psychologist with the Air Force Research Laboratory at Mesa, AZ, where he has been since 1981. He has conducted research on training system evaluation, simulator training effectiveness, and team training needs analysis. He earned a Ph.D. in Experimental Psychology from the University of New Mexico in 1979.

Michael E. Newman is the SST Program Manager and the Chief of the Space Training Acquisition Office (STAO), SCNG/ST at the Space and Missile Systems Center at Los Angeles Air Force Base, CA. He holds a bachelor's degree in Mathematics. Mr. Newman has 10 years of military experience as Air Force pilot flying T-38 and KC-10 aircraft and has been involved in space training acquisition as a civilian for the past 7 years.

Merrill C. Galligan is currently assigned as the Chief, Training Support Flight, 533rd Training Squadron. He is responsible to resource the most diverse and dynamic squadron in the group. He directs the production of course resource estimates, training plans, and training systems requirements that set the foundation for space training across the breadth of space operations for AFSPC and AETC. Mr. Galligan has earned degrees in Instructional Technology from the Community College of the Air Force and Eastern Illinois University in Education.

DESIGN & DEVELOPMENT OF A STANDARD SPACE TRAINER (SST)

Laura R. Dietz, David S. Coleman, Dr. Winston Bennett, Jr., Dr. Robert T. Nullmeyer,
Mr. Michael E. Newman, Mr. Merrill C. Galligan

INTRODUCTION

Until recently, the Air Force's (AF) 533rd Training Squadron (TRS) and the 50th Operations Support Squadron (OSS) have been using a number of different training systems and simulators for space command and control (C2), Initial Skills Training (IST) and Unit Qualification Training (UQT). Each of these training systems, or simulators, is "stove-piped," in that they use different hardware, operating systems, networking capabilities, and proprietary software, and have widely divergent, often high costs associated with their development. In addition, each system has its own maintenance concept and contract for sustainment. Since many of the training systems are replicas of the operational systems, the sustainment of the training systems in this way is frequently very costly.

To address this problem, the Space and Missile Systems Center (SMC) Space Training Acquisition Office, Satellite Control Network Systems Group/Space Training (SCNG/ST) and Headquarters (HQ) AF AFSPC/A3T worked closely together to develop the original vision for a single Standard Space Trainer (SST) for Command and Control (C2) operator training that employs commercial off-the-shelf (COTS) personal computer (PC) hardware and operating systems. In addition, as documented in Newman's SST presentation, the SST will employ one common training system architecture that will launch system-specific simulations developed to execute space operations training for a number of different satellite systems. The AF and the technical team have worked together closely to advance the SST vision and to document the detailed requirements of the program. The SST will ultimately be used to support IST, UQT, and crew training. Training will include support for the fundamentals of behavior of common military satellites, sub-systems common to all satellites, space flight, orbital mechanics, and satellite operations (to include contact procedures, ground systems basics, troubleshooting, anomaly resolution, *etc.*).

The SST application of PC-based COTS hardware, and the modeling of only the *user's experience*—not the *operational hardware/software* is predicted to be a source of considerable cost savings. Being tied to real-world hardware has a notable cost ramification. Any

time a lab is set up at any location, it involves high hardware costs and high overhead, including the high costs associated with the establishment of a classified area. The current training approach also limits the number of seats that are available for training. For a PC-based solution, the cost of a PC-based networked local area network (LAN) is minimal and the hardware itself is unclassified.

In addition, the SST vision provides a meaningful advantage for the sustainment of training systems. If a replica of the operational system is used for the training system, the entire training system must be updated every time the operational system changes. This results in a considerable amount of "down-time" for the training system. Following the SST paradigm, when a change is made to the operational system, it does not necessarily mean that a change is required for the training system. If the operational change neither affects what the operator sees on the display, nor changes the way in which he or she executes the procedure, then the PC-based training system does not have to be modified.

TRANSITION TO PRODUCTION SST

As a first step toward realizing the vision of an SST, the Air Force Research Laboratory (AFRL), Mesa, AZ, and the Air Education and Training Command (AETC) provided enhancement funding to Sonalysts under a Phase II Small Business Innovation Research (SBIR) contract in September 2006 to develop a satellite operator proof-of-concept instructional simulation that supports multiple systems. Specifically, the proof-of-concept SST provided instructional simulation support for a subset of the Architectural Evolution Plan (AEP) and Defense Satellite Communications System (DSCS) satellite systems using advanced modeling, simulation, and visualization technologies. The proof-of-concept SST was designed to validate and demonstrate the SST concept and serve as a stepping stone to the development of a production SST system.

The proof-of-concept SST was successful in using a single-system architecture that provided training for two different satellite C2 systems. Due to the success of this application, the AF decided to pursue the development of a production SST system under a

Phase III SBIR contract. The result of this task was production of a set of software requirements that reflected the needs of the Government and industry stakeholders.

Upon completion of an SMC Engineering Team Review and SMC/Son analysts SST Risk Reduction effort, the Air Force moved forward in awarding a Phase III SBIR award for the development of a Production SST. The Production SST will consist of a reusable SST Architecture, and DSCS as the first plug-in training system.

PREVIOUS TRAINING APPROACH

During the investigations leading up to the SST vision, there were two core discussion areas: modeling fidelity, and training features for the instructors. Those two topics will be discussed in the following subsections and the corresponding results and conclusions sections.

Modeling Fidelity

The previous paradigm for AF trainers was to provide operational systems/models to drive the student's experience. These systems have proven to be expensive to develop and maintain, partly due to the complexity of the models and also due to the fact that the actual operational system code and hardware are used to create a trainer. Often these trainers are testbeds of the real system, providing the capability to test real-world operations and troubleshooting tactics.

The SST approach to training is to replicate the experience of using the operational system without actually using one. This will be accomplished by presenting the graphical user interfaces (GUIs), displays, and other representations as they would appear on the real-world system. However, the user experience would not be driven by the actual operational code or system, but by a software solution that would simulate the user experience using techniques found in computer-based games. In this context, the simulation providing the stimuli to the student would not be required to be a high-fidelity model of the underlying mechanics of the real-world logistics, hardware, or processing, but rather would simulate only to the degree of fidelity necessary to provide the realistic training environment required to achieve the training objectives. *Note that with this*

methodology, the training system will be completely independent from the operational testbeds that are delivered by the operational system developers. This modeling methodology has been used for a variety of PC-based training systems for commercial and Government customers. Based on the Risk Reduction analysis the same methodology will be applied for the various SST mission areas (satellite C2, spacelift launch and operations, etc.).

Apart from the SST, engineering testbeds will continue to be acquired and utilized for engineering analysis to accomplish operational software testing, experiment with new procedures, or troubleshoot new anomalies. The SST will not replace the engineering testbeds or their utility. In the current training paradigm, very often the engineering testbed is also used as a training system. Due to the specific requirements of an engineering testbed, there is often a lack of flexibility/functionality that is desired for a training system. One example of this is how the instructor cannot train several students at the same time working on the same contact with current satellite command and control trainers. The result is that in order to train multiple students, each student has to communicate with a different satellite. This makes it difficult for the instructor to present training information that is applicable to all the students in the class, which makes it challenging to teach a group of students on the same concept. The SST Architecture and training systems will not be integrated with a real hardware or software system, so real-world constraints do not apply to this training system.

Methods

With the difference between an engineering testbed and a training system in mind, the training curriculum is analyzed to discover the events that must be presented to the student as stimuli, as well as capturing all the actions the student must be able to perform on the system to achieve the training objectives. To provide the level of fidelity needed for training space platform systems, each of the telemetry sub-systems must be modeled to some fidelity. The difficulty in determining a proper approach to modeling the space platform is compounded by the real-world complexity of the systems. For satellite systems, there could be between 1,000 and 10,000 individual telemetry points that describe the health of the system. To "model"

each point of the system to provide stimulus to the student would be a complicated and expensive task. In addition to the number of telemetry points, the possible behaviors and branching possibilities of a subsystem's state are very large. Sending a particular command to a satellite vehicle sub-system may have effects on many other sub-systems and corresponding ripple effects to many telemetry points. To address these issues, an effective approach to modeling must be devised that will meet training objectives, without introducing extremely complicated modeling that will be costly and time consuming.

During the Risk Reduction effort, the impressions and opinions of multiple instructors were gathered, and the technical team witnessed the operation of current training systems to provide input to this question. The Risk Reduction analysis included input from instructors at the 50th OSS, the 460th OSS, and members of the Spacelift community from the 45th Space Wing. The goal of this portion of the Risk Reduction effort was to determine an appropriate modeling approach to training; this approach is tied to the nature of the student experience (*e.g.*, free play requires more robust modeling). From the observations of the technical team, some of the current trainers do not seem to provide the "cascade effect" of data points when an anomaly occurs, but instead only a subset of data points change, to create the initial stimulus on which the student may act. One instructor indicated that a high-fidelity experience was required. Other instructors indicated that they only train to a specific set of anomalies, and the students only focus on certain data points, since there are too many to have full awareness of each. In addition, when a student fails, the instructors indicate that the student is told to stop and the training session ends (a "No Go"). Therefore, a full free-play capability is not necessarily desired, since once the student diverges from the checklist, the training ends.

Results

In the requirements effort, two different modeling approaches emerged. The first approach is to employ state-based modeling. This methodology would consider only effectively changing data points as required to introduce stimuli to the student. Since the end result of what the student does is only to react to a data value changing on the screen, then the only

modeling required is to update data values appropriately. For example, with a satellite C2 training system the way in which telemetry points would update and change would be determined by the curriculum, so only the stimuli required to satisfy the training objectives would be provided. Other non-critical telemetry points would be represented, but only as values that operate within nominal or expected limits. This method would use a distributed data store to provide telemetry values and definitions of how a value will change over time. A data store of commands and how they affect telemetry points will also be implemented in this methodology. In a similar fashion, the visualization of the various graphs viewed by the Mission Flight Control Officer (MFCO) (velocity versus time (VT), instantaneous impact point (IIP), *etc.*) could be provided based on the data store values maintained by the system. If an event occurs (nominal or anomalous), then a "change set" or cascade of changes could be scripted to occur in conjunction with the event.

The second modeling approach is physics-based modeling. In this approach, key elements of the satellite sub-systems are modeled by code or algorithms to present realistic satellite behaviors and corresponding realistic telemetry values. Key components are simulated in code, including batteries, propulsion, and other key components, with a "gaming" approach to modeling. The values represented by the data points will reflect "reasonable" approximations of how the data point will change in a real system. For example, if a satellite C2 training system simulates a satellite maneuver and the student initiates a burn of two of the thrusters, the satellite should move in the appropriate directions and behave as it would in the real world. However, the magnitude of how far the satellite moves related to how long the thruster burns would be proportional to how the real system would behave, but would not be the real-world magnitudes. In other words, the trainer will not provide the modeling accuracy to such a degree that if a burn is done in the trainer, it will exactly match the real-world system's values. In a similar fashion, the trajectory of a launch vehicle could be simulated with a physics-based modeling approach. However, the trainer is not intended to be used as testbed. The objective is to make the user's training experience with

the SST comparable to their experience with the real-world system.

The advantages of the state-based modeling approach are:

- It is cost-effective to design and implement.
- It is a simplified approach for including new anomalies, by adding the additional scripting logic in the scenario editor.
- It places control in the hands of the instructors to add or tweak existing anomalies.
- It provides the ability to customize telemetry points to reflect the peculiarities of a real-world satellite.

The disadvantages of the state-based modeling approach are:

- Developing training scenarios involve more scripting, and additional work will be required to create a usable Scenario Editor GUI.
- The experience of the event may be impaired slightly or does not fully reflect the real system.

The advantages of the physics-based modeling approach are:

- The benefit of this approach is that the telemetry screens have a system model providing the backbone behind the telemetry values. This allows for cascading effects or ramifications of an anomaly and an improved ability to present realistic behaviors.
- Less scripting is required with this methodology. Initial conditions are required to be set, and then the training models take over the ramifications.
- This approach to modeling facilitates providing basic building blocks for model services that can be included in the architecture. For example, a battery model or a sun sensor model could be packaged in the architecture for later re-use and customization by other vendors.

The disadvantages of the physics-based modeling approach are:

- Additional effort is required to create the model. In addition, the modeling may require more knowledge of the actual system.
- The selection of the components to model is critical to providing enough modeling to support training. The components must ensure all the training objectives are covered properly. Also, the modeling provided must be sufficient for training, so that the approach is not bogged down in providing high-fidelity modeling and simulation.

Based on the analysis conducted in the Risk Reduction effort, SST is employing a hybrid approach to combine the state-based approach and the physics-based modeling method and apply them to the space domain. For telemetry points that are not deemed important for training, the state-based modeling approach can be used to generate realistic values from a data store and update them as required. For important parts of the sub-systems, selective modeling can be applied and code can be written to provide enough modeling detail for training. Any telemetry point changes not covered by the modeling can be controlled by the telemetry data store scripting mechanism. In addition, this approach will support the High Level Perspective View (HLPV) visualization as well. Whatever models are generated will then be included in the SST Architecture to facilitate other vendors' development of mission-specific plug-ins.

Training Features

Through various Space Training Advisory Groups (TAGs), Training Planning Teams (TPTs), and other forums, there have been frequent discussions regarding instructor feedback on the functionality and usability of the training systems currently in use. The current training systems were developed by different companies, under different contracts, and with different stated requirements. This results in a wide variety of training features that are available to the instructor, depending on the system they may be using. During the instructor interviews, the instructors were asked to generalize and identify the most prevalent of the training feature deficiencies that they have encountered. During the development of the proof-of-concept SST, one of the primary goals was to demonstrate the feasibility of addressing the deficiencies identified by the instructors. After

consulting with the instructors at the initiation of the effort, it was determined that the proof-of-concept should address the following generalized items:

- Current trainers are difficult to set up, involving hours of pre-configuration/setup before actual training can begin.
- Current trainers are difficult to shut down, often involving a time-consuming exiting process.
- Some of the current trainers do not have an instructor monitoring function, in which the instructor can view a student's screen on his computer. The inability of the current trainer to view the student's actions increases the instructor workload and also increases the number of instructors required to monitor a simulation session.
- Some of the current trainers do not provide scenarios with pre-established anomalies. Therefore, this requires more instructor "hands-on" interaction when running the labs. In particular, the instructors must remember to insert failures at the appropriate times for a given lab.
- Some of the current training systems do not provide the instructors with the ability to pause, rewind, or fast-forward the simulations.

Methods

During the proof-of-concept effort, the team followed the software engineering process by documenting current and future requirements, and the corresponding design issues to address the training feature items in the proof-of-concept within the resource allocation. As a first step, a Concept of Operations (CONOPS) artifact was produced and delivered to the AF stakeholders describing the proposed functions and features of the proof-of-concept SST. During the development effort, instructors were invited to review the software with the technical team to provide feedback on the instructor training features, as well as the fidelity of the proof-of-concept simulations.

Following the proof-of-concept, the AF updated the SST Operational Requirements generated at the inception of the SST vision. Using the revised requirements and feedback from the proof-of-concept, the Risk Reduction effort included further requirements analysis and documented the requirements for a production training system.

Results

In the proof-of-concept SST, the majority of the training feature items were addressed to show the capability of a PC-based training system to be designed to meet the instructors' needs. The list below summarizes how the training feature items were addressed:

- The proof-of-concept SST has reduced trainer set-up time to just a few minutes—the time it takes to start up the computer, log in to the SST proof-of-concept software, and select the desired training scenario.
- Shutting-down the proof-of-concept SST requires a brief period of only a few minutes to exit out of the application and shutdown the computer.
- The proof-of-concept SST enables the instructors to remotely view each of the six students' displays in a current training session from the instructor console, which allows for quick response to student actions or questions.
- In the proof-of-concept SST, the instructor is able to pause or fast-forward time in the training session (rewind functionality will be available in the production SST system).

With regard to providing instructors with scenarios with pre-established anomalies, it was determined that the production system SST will remove this burden from the instructors by providing scenarios that already have the failures and events pre-planned, so the labs are more consistent and do not involve as much instructor interaction to facilitate the scenario, thus allowing the instructors to focus their time on interacting with the students.

Combined Conclusions

Overall, the proof-of-concept SST was well-received by the instructors at the 533rd TRS, and their experience with it shows the potential of how the full production SST will improve AF space training efficiency and productivity. In the AF SBIR/Small Business Technology Transfer (SSTR) Transition Story, it was documented that "the experience of the 533rd Training Squadron is that required instructor manpower has been reduced significantly, while student evaluation productivity has increased dramatically."

The 533rd TRS has estimated that the fielding of the proof-of-concept SST, which consolidated two simulations into one COTS solution, saved the training squadron \$5 million. In addition, the squadron maintains that the first proof-of-concept simulation is 87% faster and more user friendly for the instructors than their legacy training system.

Based on the feedback received on the proof-of-concept SST and the Risk Reduction analysis, the SST vision was formalized and a Phase III SBIR contract was awarded to commence development of the SST Architecture and a comprehensive DSCS training suite to leverage the architecture. Building the DSCS training suite with the SST Architecture will help to verify that the features of the SST Architecture have been fully identified. This development effort will provide another level of verification/validation of the SST Architecture. Finally, the DSCS training suite will be a demonstration of the architecture and its capabilities.

The SST Architecture will provide a framework for future space training systems. This framework will enable space training systems to be built on a PC-based COTS hardware solution, without reliance on the actual hardware or software of the satellite. A Software Development Kit (SDK)/Application Programming Interface (API) package will be provided to allow the development and integration of new training systems within the SST framework. The SDK will provide common tools and modules for rapid development of training systems, thereby decreasing the cost and time required to develop an individual training system. In addition, a library of models, software modules, and utilities will be provided. Using these tools, training system developers can devote the majority of their development effort to simulating the unique characteristics of the specific systems, thereby reducing training development costs.



Figure 1. SST Architecture Vision

The SST Architecture will provide training services and modeling services. The combination of these services will facilitate the construction of future space training systems. The training services portion of the SST Architecture will provide all the infrastructure, tools, and support to create an effective training system. The training services will include:

- Network Support – The SST Architecture will support setting up training sessions with instructors and students.
- Crew Training Support – The SST Architecture will support setting up training sessions for crew training, with each student assigned to a different position.
- Instructor Control Features – The SST Architecture will provide tools to the instructor to monitor and control an instructor-led training session. The production SST will support multiple training sessions and provide the ability for the instructor to monitor these students efficiently.
- Student Monitoring and Performance Tracking – The SST Architecture will provide the capability to monitor student performance during training sessions and to store this data for reports and reviews.
- Scenario Loading and Execution – The SST Architecture will provide tools to load scenario files and execute them during a training session. Each scenario file will contain goals for the students to accomplish, as well as any scripted events or failures that must occur at designated times during the scenario.
- Scenario Editor – The SST Architecture will provide a Scenario Editor that will allow instructors to create new scenarios to train students. The Scenario Editor will be expandable to meet the diverse requirements of different space systems.
- Replay – The SST Architecture will provide a Replay feature so that student training sessions will be played back for review and instruction. The architecture will provide an API to capture replay data and a framework in which replay data can be displayed in the Replay Viewer.
- Report Generation/Training Management – The SST Architecture will provide a report

generation feature so individual training programs can develop the reports that are needed for student evaluations.

- HLPV – The SST Architecture will provide a two dimensional (2D)/three dimensional (3D) visualization of the environment and the satellite, so the student can see the ramifications of his or her actions. This visualization technology will aid the student in visualizing the effects of satellite commanding and will enhance the student's understanding and emphasize the importance of the mission.
- DMO-S – The SST Architecture will support the addition of plug-ins that will support Distributed Mission Operations – Space (DMO-S).

In the modeling services portion of the SST Architecture, the framework will enable modeling requirements to be met for a space system trainer. In particular, the modeling services will provide a base set of generic models that are required for any satellite-based system and that can be modified to meet specific needs. The modeling services portion of the SST will be expandable to insert new models into the architecture to meet the specific needs of space systems.

FUTURE OF SST

Air Force Space Command is investigating expanding the use of the SST approach outside of satellite command and control areas to include the following subject areas: cyber security, Intercontinental Ballistic Missile (ICBM) missile maintenance officers, Joint Space Operations center (JSPOC), and the spacelift community. Headquarters AFSPC Generals Kresge and Hyten have promulgated a memorandum for all Air Force Space Command leads directing that all future space training systems for combat mission ready (CMR) positions will use the SST Architecture. In addition, a Memorandum for Secretary of the Air Force from General Kehler addressing improvement of Space Console Operator Training stated that “Our changes focus on taking advantage of a streamlined AETC process for major course changes and new course development, shifting our initial training focus from operational units to AETC to produce fully trained operators and building a standard space trainer to improve simulation capabilities for our space operators.”

In addition, the Government and technical team are working to advance the SST vision to the next step by including adaptive training capabilities in future SST training systems. The adaptive training components would be developed in conjunction with the space training system for the SST Architecture, and the resulting adaptive training will be launched from the SST Architecture.

Adaptive Training

With adaptive training, the learner receives individual training (one student per PC) using the Closed-Loop Adaptive Training (CLAT) approach. Closed-loop adaptive training is an emerging concept that integrates object-/objective-based adaptive Interactive Multimedia Instruction (IMI), an Intelligent Tutoring System (ITS), and modeling and simulation technologies. The result is an inherently learner-focused approach that manages mastery of knowledge and skill learning objectives (LOs) dynamically.

A closed-loop adaptive training system combines adaptive interactive multimedia instruction to teach fundamental knowledge with simulation-based coaching for skill training. The goal of the CLAT system is to give each individual learner what they need, when they need it, in order to achieve the goals of the course. To do this, the system moves learners seamlessly between adaptive multimedia and intelligent tutoring environments until they can demonstrate mastery of the targeted LOs.

Instructional Design Methodology

As documented in “Emerging Technologies in Training Development” by McCarthy, Morris, and Castro, a primary objective of the adaptive training approach is to outline an instructional design process that would produce learner-focused training systems. It is not that current instructional design processes do not exist but rather that training planning processes tend to focus on front-end analysis that define requirements for high-fidelity simulation-based training systems. Progress is slow in investments in learning methodologies that can improve training effectiveness and efficiency. Instructor-led classroom instruction or low level (2-3) interactive courseware remain the curriculum glue that produces operators and technicians. The overarching methodology that is defined herein is based on the

premise that training systems should be developed to support efficient and effective knowledge acquisition and skill development for target LOs.

The first step in developing a training system is to adopt a learning theory. These theories, and there are many of them, address how people learn new things. That is, these theories describe what goes on inside a person’s head as he or she learns something in a formal or informal setting.

Second, one must investigate the nature of the content to be taught and the level to which it must be learned. Placing content into a taxonomy brings certain instructional methods into play and removes others from consideration. Together, the learning theory and the content analysis lead one to select a given instructional model. Whereas *learning* theory describes mechanisms inside the learners’ heads, *instructional* models describe how to structure the environment (in our case, the courseware) to facilitate learning of the content.

SUMMARY

To summarize, by using the SST Architecture and associated API/SDK to develop training systems, the goal is to produce space training systems that provide modeling capabilities required to meet the training objectives, while providing standardized functions and features that facilitate the instructor’s use of the training system. Using the training system as the platform, the next phase of the vision is to employ instructional design methodologies to construct adaptive training components to integrate with the space training system to provide individualized training to the student by supporting the instructional needs (both knowledge and skill components) of individuals by employing a seamless combination of adaptive IMI and simulation-based intelligent tutoring.

REFERENCES

- Air Force Research Laboratory (AFRL) Human Effectiveness Directorate (2008). Standard Space Trainer (SST) Offers Expanded Simulation Features. *Air Force SBIR/SSTR Transition Story*, WPAFB08-5259, 1-2.
- Kresge, Stanley T., USAF Brigadier General, & Hyten, John E., USAF Brigadier General (2008).

- Standard Space Trainer (SST) for Mission Ready Training. *Memorandum for AFSPC Command Leads SMC/CV*, 1-2.
- Kehler, Robert C., USAF General (2008). SECAF Tasker to Improve Space Console Operator Training. *Memorandum for SECAF*, 1.
- McCarthy, James, Morris, John, and Castro, Kim (2000). Emerging Technologies in Training Development. SBIR Final Report delivered to AFRL, 67.
- Newman, Mike (2006). Standard Space Trainer (SST) Presentation.