

Available online at www.sciencedirect.com**SciVerse ScienceDirect**

Procedia Computer Science 16 (2013) 1082 – 1091

Procedia
Computer Science

Conference on Systems Engineering Research (CSER'13)

Eds.: C.J.J. Paredis, C. Bishop, D. Bodner, Georgia Institute of Technology, Atlanta, GA, March 19-22, 2013.

Designing an Experiential Learning Environment for Logistics and Systems Engineering

Douglas A. Bodner^{a*}, Jon P. Wade^b, William R. Watson^c, George I. Kamberov^b^aGeorgia Institute of Technology, Atlanta, Georgia 30332, USA^bStevens Institute of Technology, Hoboken, New Jersey 07030, USA^cPurdue University, West Lafayette, Indiana 47907, USA

Abstract

Systems engineering increasingly addresses the system lifecycle, as opposed to its more traditional role focusing on design and development. This new situation results in part from the recognition that upstream design and deployment decisions have potentially significant cost and performance implications post-deployment. For military systems, the role that typically addresses post-deployment issues is the logistician. Over the system lifecycle, it is important that the traditional roles of systems engineer and logistician understand issues faced by one another, as well as joint cost and performance implications. This paper presents the design of a role-based experiential learning environment for logisticians involved in military sustainment. This design leverages the generic components of an existing single-learner technology base, the Experience Accelerator, for presenting and controlling the learner experience, plus simulating program outcomes resulting from learner decisions. This technology base has been used to create a learning experience for a lead systems engineer in charge of designing and developing a new unmanned aerial vehicle (UAV) system. In this new environment, the logistician learner interacts with systems engineers during UAV system acquisition and sustainment, learns about systems engineering issues and their effect on logistics, tries to influence upstream systems engineering decisions, and also performs logistics functions.

© 2013 The Authors. Published by Elsevier B.V. Open access under [CC BY-NC-ND license](https://creativecommons.org/licenses/by-nc-nd/4.0/).

Selection and/or peer-review under responsibility of Georgia Institute of Technology

Keywords: Experiential learning, experience accelerator, system lifecycle, logistics

1. Introduction

Systems engineering increasingly is called on to address issues in the lifecycle of systems, as opposed to its more traditional role that focuses on design and development. This results from the recognition that decisions made by systems engineers in the design and development phases have important implications for cost and performance in downstream system lifecycle phases such as production and sustainment. For military systems, it is primarily

* Corresponding author. Tel.: +1-404-894-2363.

E-mail address: doug.bodner@gatech.edu.

production engineers and logisticians who address technical issues in these two downstream phases. In doing so, they engage in a substantial amount of functional work, but they also engage with systems engineers and others in working committees known as integrated product teams (IPTs), which are set up as a multidisciplinary means for different functional stakeholders to interact and understand one another's perspectives in defining a product or process or in delivering recommendations. Such IPTs provide forums for these stakeholders to understand how decisions in one functional area (e.g., systems engineering during the design phase) can affect outcomes in others (e.g., cost in sustainment), and to make trade-offs when there are potential conflicts between desired outcomes of different functions.

As systems become more complex, it is increasingly important that people in functional roles such as systems engineers and logisticians have effective means, either formal ones such as IPTs or informal ones such as social networks, whereby each role can understand the effects that decisions in other functions have on their function and vice-versa. Moreover, it is equally important that people in functional roles learn how to use such means to transmit knowledge across functions, as well as influence decisions in other functions in making needed trade-offs. This requires both technical knowledge (e.g., the technical aspects of interactions between functions) and professional knowledge (e.g., how to elicit issues of concern and to influence decisions). For example, designing a system to be modular may result in reduced sustainment cost, as repairs and upgrades to deployed systems tend to be less costly, but it may also result in increased costs for design and development [1]. The systems engineer and logistician, in this case, must understand the technical aspects of the interaction between modularity and cost and use influence or negotiation to make the right trade-off.

However, most educational programs that teach functional area topics focus mainly on the technical knowledge for the particular function, not on cross-domain knowledge or on the knowledge needed to use multidisciplinary methods for knowledge elicitation and influence. This latter type of knowledge typically is gained by on-the-job experience and can take many years. In today's environment, there is a critical need to improve acquisition program workforce training to reduce persistent problems in program outcomes related to technical performance, schedule and cost [2]. One avenue to improve training involves use of educational technology, in particular role-based experiential learning. This paper explores the creation of such a learning environment in which a learner assumes the role of a logistician who must not only understand issues in his or her functional area, but also must interact with systems engineers to understand the implications of decisions in that functional area on logistics. This environment is based on the Experience Accelerator, an existing prototype learning environment used for systems engineering workforce development [3, 4].

The remainder of this paper is organized as follows. Section 2 reviews research being done in the area of experiential learning environments that harness technology to provide role-based learning. The current prototype version of the Experience Accelerator is described in Section 3. This prototype focuses on the traditional role of the systems engineer in terms of the experience delivered to learners, but it contains a technology infrastructure that can be generalized to other type of learning experiences. Section 4 presents a design for an experience where a chief logistician must manage the technical aspects of the sustainment operations of a program after providing input into the design and development phases of the program, whose technical aspect were managed by a chief systems engineer. Section 5 discusses an assessment method to be used in evaluating the effectiveness of the Experience Accelerator in terms of this new experience. Finally, Section 6 concludes and provides avenues of future research.

2. Experiential Learning

In recent years, educational technology has received wide attention as a means to augment other forms of education and training. In particular, we focus on the concept of an experiential interactive learning environment. An experiential interactive learning environment operates similarly to a role-based digital game, in that the learner assumes a role associated with a particular domain and engages in role-play within a digital environment representing the domain. The digital environment may assume different forms, from a simple text-based display to an immersive three-dimensional world. In an experiential interactive learning environment, of course, the role-play is aimed at furthering educational outcomes.

These types of environments often use Kolb's concept of experiential learning, which consists of four key elements through which a learner cycles in a continuous spiral of learning [5]. The four elements are (i) exposure to a concrete experience, (ii) reflection on that experience, (iii) generalization of the experience and formation of

abstract concepts based on the generalization, and (iv) application of these concepts to the concrete experience. In an experiential interactive learning environment, the concrete experience is replaced with a simulated world, and the learner interacts with it, causing the simulated world to change. Through these interactions and world changes, the learner can cycle through different experiences in the same domain or context, thus enabling a spiral learning process. Another concept typically utilized by these environments is constructivism, a theory that posits that learners construct their own knowledge rather than simply adopting pre-packaged knowledge. Goals for constructivist instruction include solving problems, reasoning, thinking critically and using knowledge both actively and reflectively, while conditions for effective constructivist learning include use of complex and realistic environments, use of social negotiation, multiple perspectives and learning modes for learners, encouragement for self-learning and self-awareness of knowledge construction [6]. Yet another concept is that of role-playing educational experiences, which can be used to teach proper actions, vocabulary and ways of thinking in a particular domain [7]. While real social and physical environments have traditionally been used, researchers have concluded in recent years that computer-based environments are a suitable alternative [8, 9].

Educational games and experiential interactive learning environments have been deployed in a number of domains, including logistics. Logistics provides an interesting application for these environments, as it is characterized by risk and disruptions, interactions with other parties (e.g., suppliers, transport providers, customers) and multiple performance metrics (e.g., cost, on-time delivery, penalties, profit). Thus, a number of efforts have addressed experiential learning for commercial logistics applications, with the intent to teach such concepts as inventory management, meeting customer expectations, cost analysis and minimization, and emergent supply chain phenomena such as the bullwhip effect [10-13]. Military sustainment logistics encompass these concepts, but also have characteristics not commonly found in commercial applications. Some features that differ from commercial applications include the need for collaboration between logisticians and systems engineers, the use of performance-based logistics, the public-private nature of the sustainment enterprise, the risks in the industrial base from supplying complex parts unique to one program/customer, and the changing mission profiles over the sustainment lifecycle. We are not aware of educational technologies that address these issues, which are increasingly important for military systems, as fiscal realities are likely to cause systems already deployed and currently in the pipeline to have longer-than-planned operational service.

3. Experience Accelerator Architecture and Technology

To address related challenges associated with workforce development for systems engineers in the U.S. Department of Defense, the Systems Engineering Experience Accelerator (SEEA) has been designed and developed [3, 4]. The intent is to provide an experiential interactive learning environment that supports skill development across a range of competencies that a senior systems engineer should possess. The learner assumes the role of a chief systems engineer for a major DoD program. In the current prototype version, the learner is the chief engineer for a program designing and developing a new unmanned aerial vehicle system. While this prototype addresses a particular learning experience, the Experience Accelerator is intended to be a generic concept and technology set that supports the four phase cycle of experiential learning, constructivist approaches to learning, and role-based experiential learning for multiple types of experiences in systems engineering and other similar domains. The Experience Accelerator combines social and technical aspects of a domain, as both are important in experiential learning for workplace domains. In this section, the architecture and technology of the Experience Accelerator is described, along with the current systems engineering experience.

3.1. Experience Accelerator architecture

The Experience Accelerator consists of five main modules that interact to provide a learning experience. These modules are generic in the sense that different experiences can be designed and accommodated via different datasets, as shown in Fig. 1. It should be noted that the Experience Accelerator uses open-source technology for these modules wherever possible. The modules function and interact with one another as described below. Further details of the Experience Accelerator architecture are described by Wade et al. [14].

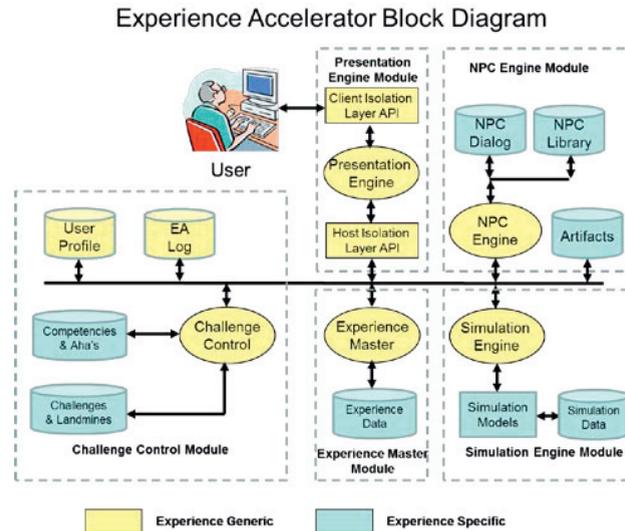


Fig. 1. Experience Accelerator architecture

- *Presentation engine* – Provides a user interface to the learner. This interface displays the status of the simulated program and lets the learner interact with a simulated program office. This interaction includes exploration of archived documentation on the program, querying of non-player characters about issues of interest regarding the program, and entering decisions about the program to correct perceived problems.
- *Experience master* – Maintains current value of state variables related to program status and performance and manages the invocation of other modules as needed. The experience master controls the flow of events and provides updated information from the other modules to the presentation engine for display. It also takes the learner inputs from the presentation engine and invokes internal functions or other modules as appropriate.
- *Non-player character engine* – Provides a number of non-player characters (NPCs) and associated dialog sessions based on the state of the program. The non-player characters have roles (e.g., program manager), and the learner is tasked with querying them about various aspects of program performance to determine the true state of the program and its underlying problems.
- *Simulation engine* – Maintains the detailed state of the program and advances this state via simulation models, incorporating learner decisions about the program. The simulation engine takes as input a base state of the program and then advances this state using any changes the learner may have entered. It also generates artifact charts detailing the program status and performance over time (e.g., costs incurred, schedule and progress indicators, system performance estimates relative to targets, etc.).
- *Challenge control engine* – Maintains learner history and skill levels and provides parameter-tuning to increase or decrease the difficulty of the experience based on learner knowledge and skills, and his/her success.

3.2. Experience Accelerator technology design summary

Within this architecture, the Experience Accelerator technology supports a single-learner experience and operates using a phase-based approach to supporting a learning experience. This is consistent with the systems lifecycle, which typically is divided into different phases. For instance, in defense acquisition major phases of the system lifecycle include engineering & manufacturing development and production & deployment. To complete a phase requires meeting certain performance targets and passing a review based on program status, with a baseline schedule for meeting the targets. If the targets are not met by the scheduled date, it is possible to push back the schedule. If performance falls substantially below expected progress, a program breach might occur, and/or the program might be terminated. In addition to the phases in the learning experience, the Experience Accelerator provides an initial orientation phase and a capstone reflection phase for summative learning on the overall experience.

Within each phase, the technology is designed to guide the learner through a set of cycles. Cycles give the learner several opportunities to make progress and correct problems during the span of the phase. A cycle consists of two parts – a simulated day (compressed in time for the learner) and then a simulated program advancement. During the simulated day, the learner must gather information about the program status and performance and must make critical decisions to correct current or anticipated problems. The technology supports the following variety of passive and active approaches to information gathering:

- Receiving email updates on program status and issues or email queries from non-player characters,
- Reviewing program background information,
- Reviewing program status and performance reports from the previous cycle (or prior cycles or phases), or
- Engaging with a non-player character via telephone or email to get answers to questions about program status, as well as risks and problems/issues.

Dialog with non-player characters is scripted and utilizes a spoke-and-hub formalism in which the learner can engage in a particular topic (i.e., “hub”) and then drill down into different levels of detail (i.e., branching via “spokes”), then move onto another topic (“hub”) [3]. The dialog is state-based, in that certain dialog is triggered if the program is in a desired state, whereas different dialog is triggered in other states. Similarly, email content is state-based. The learner’s dialog consists of a set of scripted question choices, as well, since the Experience Accelerator does not support natural language processing.

The learner has access to a program recommendation form, which may be used to make decisions to address anticipated or current program problems. For instance, the learner may decide to recommend adding resources in a particular area to increase the progress rate. After the learner has gathered all the information desired, he or she then uses this form to make decisions. At this point, the simulated day ends, and the experience master invokes the simulation engine to advance the state of the program for the remainder of the cycle, using the learner’s decisions as input. The learner then sees results on the next simulated day.

The experience master, simulation engine, NPC engine and challenge control engine are implemented using Java™. They operate on XML files that contain text-based content for NPC dialog sets, emails, simulation models, etc. Digital recordings can also be accommodated for dialog. The presentation engine uses Adobe Flash™ to present forms, documents and program status, including a graphical program status dashboard.

3.3. Experience design and a systems engineering experience

A particular learning experience starts with a set of competencies and generic learning moment concepts to be taught [3]. For the systems engineering experience, a competency model for a chief systems engineer for DoD programs was specified based on several existing models [15]. From this model, two competencies were selected for the systems engineering prototype experience – systems thinking and problem solving and recovery. A generic learning concept is found in many situations within the domain of the target experience. For example, in systems engineering one such learning concept is that upstream decisions have important downstream consequences.

Using the generic technology base and the competencies and learning concepts to be taught, a particular learning experience can be designed using the following steps (with iteration between steps), summarized in Table 1 with the corresponding examples from the systems engineering experience. Note that detailed phase design consists of four elements, and these elements are designed concurrently.

4. Logistics Experience Design

In the logistics experience, the learner is assigned the role of chief logistician for the same DoD program as in the systems engineering experience. The learner is concerned with providing input into and gathering information from the UAV program’s acquisition phases, as well as with managing the technical aspects of sustainment. In particular, competencies desired include supportability analysis, reliability and maintainability, and configuration management. The generic learning concept is the same as with the systems engineering experience, namely that upstream decisions affect downstream outcomes. The experience supports these objectives by having the learner (i) specify the sustainment network based on supportability analysis, (ii) determine reliability and adapting maintainability plans accordingly, (iii) manage multiple configurations for different missions and multiple generations of

technologies due to upgrade availability, and (iv) interact with systems engineering and production engineering NPCs who demonstrate the effect of upstream decisions on the logistics function.

Table 1. Experience design steps and systems engineering examples

Design step	Systems engineering experience
Select a particular scenario (e.g., a particular DoD program) for the experience that supports the desired learning and develop a storyboard for the desired experience elements.	The learner is assigned as the new chief systems engineer of a UAV program subsequent to Preliminary Design Review. The program has a lead contractor and three sub-contractors developing the airframe/propulsion, the command & control and the ground station, respectively. The scenario addresses (i) systems thinking via the concurrent development and integration of these component systems, (ii) problem solving and recovery by having the learner identify and resolve problems, many of which are handed off by the previous chief systems engineer, and (iii) the effect of upstream decisions on downstream outcomes via modeling the lifecycle from development to low-rate initial production.
Define the program's high-level performance measures, their targets and deviations from target that cause breaches.	High-level performance measures include schedule, cost, quality and technical performance. Targets are various points in the program are defined, as well as percentage deviations from targets.
Define phases and cycles within phases.	Four experience-related phases are included – pre-integration development (ending with Critical Design Review), system integration (ending with Flight Readiness Review), flight test (ending with Production Readiness Review), and low-rate initial production. Each has between two and four planned cycles. Program background information artifacts should include status at beginning of experience, projected budgets, and domain information (e.g., UAV systems).
As part of detailed phase design, specify program status indicators and targets that operate at a more detailed level than high-level performance measures.	Schedule for each phase is operationalized by progress measures for meeting a set of entrance criteria needed to perform the critical review meeting that caps the phase (e.g., Critical Design Review). Cost is modeled via Earned Value Management [16], which is tracked for each contractor and rolled up into an overall program cost status. Quality is operationalized by identified defects in software design and implementation for the command & control system. Finally, technical performance is operationalized using range as the key performance parameter (KPP) with supporting technical performance measures (TPMs) of drag, weight and propulsion efficiency.
As part of detailed phase design, specify challenges and landmines that the learner will face, as well as correct learner actions to address them and incorrect possible actions. Design a learner recommendation form to accommodate these actions.	Challenges include weight growth in the command & control system and its effect on range, high levels of software defects in the command & control systems, and schedule issues for the airframe/propulsion and the command & control systems. A landmine occurs at the beginning of system integration when wind tunnel tests indicate that drag is significantly higher than estimated. Correct learner actions are specified, as well as incorrect ones. The do-nothing option is usually incorrect. The learner recommendation form encompasses these choices and also allows the review meeting to be pushed forward or back.
As part of detailed phase design, create characters and state-based dialog,	Characters include the program manager, the previous chief systems engineer (available only at experience beginning), the chief engineer for the lead contractor and a government test representative. A human resources representative and a mentor are included to aid with orientation and feedback, respectively. Detailed dialog is developed to support states associated with good program performance and poor performance in different areas due to challenges.
As part of detailed phase design, create simulation models and output charts.	A simulation model is specified for each experience phase. Sub-models include contractor engineering workforces, the UAV system and its technical performance estimates, the entrance criteria for review meetings, software defects for command & control, and earned value management. Variables whose values span different phases are specified, as are variables that the learner can manipulate.
Design feedback to learner based on correct and incorrect actions that were selected	Feedback on performance is provided in the reflection phase. It consists of the learner's decision at each cycle, the best decision (and congratulations if the learner made it), and the rationale for why the best decision should have been made.

The sustainment program uses a mixture of government and contractor-based support (operating under a contractor logistics support contract awarded to the prime development/production contractor). The program's high-level performance measures consist of the following:

- System availability (KPP) – average percentage of systems available to perform an assigned mission
- System reliability (quality) – mean time between system failures (i.e., system unable to perform)
- Overall cost – annual sustainment cost for all deployed systems
- Cost per flight hour – overall cost divided by number of flight hours

4.1. Phase and cycle design

The phases for this experience consist of some overlap with the systems engineering experience, wherein the learner gathers information and tries to influence decisions. These phases are described in Table 2.

Table 2. Phases for Logistics Experience Accelerator

Phase	Duration and cycles	Description
Engineering & manufacturing development	Three years and two cycles	This phase corresponds to the first and second phases of the systems engineering experience. The UAV system is being developed and components integrated in preparation for flight testing. The learner needs to gather information and influence decisions that will affect sustainment and plan the sustainment network (repair facilities, part suppliers, personnel, fuel, etc.).
Flight test	One year and one cycle	This phase corresponds to the flight test phase of the systems engineering experience. The learner is tasked with confirming previously gathered information and finalizing the sustainment network.
Low-rate initial production (LRIP)	One year and two cycles	This phase corresponds to the same phase in the systems engineering experience accelerator. The learner starts taking delivery of systems, some of which have problems due to not-fully-mature designs and production processes.
Full-rate production	Two years and four cycles	The system is in full production with some schedule slips. There is contention for parts between production (needed for new systems) and sustainment (needed for repairs). The issue of repairable parts becomes critical, as these can be used for either new or existing systems.
First post-production year	One year and four cycles	Production ceases due to government cutbacks. The learner now manages the sustainment, including maintenance, repairs and technology upgrades. The system has been deployed in two theaters.
Second post-production year	One year and four cycles	The learner continues managing sustainment as mission needs change constantly. New technologies become available for system upgrades, and the learner must manage sustainment of different configurations.

4.2. Detailed phase design

A logistics planning IPT operates during acquisition, and the learner chairs this committee. Its members are available as NPCs with whom the learner can interact and gather information.

- Program manager (logistics planning IPT member) – government executive in charge of overall program.
- Chief systems engineer (logistics planning IPT member) – government technical lead for engineering & manufacturing development phase of program.
- Chief systems engineer for prime contractor – prime contractor technical lead for engineering & manufacturing development phase of program.
- Chief production engineer (logistics planning IPT member) – government lead for managing production.
- Testing representative – government lead for flight testing and system acceptance for production.
- Commander for first deployed theater – customer who is using deployed systems in Operations Theater 1.
- Commander for second deployed theater – customer who is using deployed systems in Operations Theater 2.
- Chief logistician for prime contractor (logistics planning IPT member) – prime contractor lead for logistics.
- Chief logistician from previous generation program – available intermittently.

During the experience, the learner is faced with a number of challenges and landmines, as summarized below (including mitigation strategies):

- The chief systems engineer has to make a decision as to whether the air vehicle will have a modular next-generation design or will reuse a previous generation non-modular design. This design will increase his cost, and since his program is behind schedule and over budget, he is very reluctant to adopt the new design. The learner should gather evidence from other similar programs that the new design could reduce overall lifecycle cost and make his or her case to the program manager, while arguing that the chief systems engineer needs increased budget to make this happen.
- Estimates for reliability of key system components from the acquisition phase are incorrect and overly optimistic. The learner should query members of the sustainment planning IPT to determine doubts about estimates and also query the prime contractor chief systems engineer and the logistician for the previous generation program, whom he or she met at a conference the previous year.
- Systems from the LRIP have reliability problems and need components replaced with more mature variants.
- A new surveillance technology becomes available in the first post-production year that dramatically improves the reconnaissance capabilities of the system. There were indications of this technology in its developmental stages provided in program background materials. The learner should have reviewed these documents and queried the chief systems engineer as to his or her opinion on the technology's suitability for adoption. Upon determining that it is suitable, the learner should modify the sustainment plan accordingly.
- A terrorist organization takes control over a large area of a failed nation-state that neighbors the territory in Operations Theater 1 and is thought to be making plans for large-scale strikes either in Europe or the United States. Systems are ordered to be transferred from Operations Theater 2 to Operations Theater 1 for strike missions. The learner should query both theater commanders to determine the new requirements and readjust the sustainment network accordingly.

In addressing these challenges and landmines, as well as normal responsibilities, the learner has the following options. Note that these are recommendations to the program manager. In the current Experience Accelerator, recommendations are approved automatically; however, in future versions, the learner may be required to provide justification.

- Adjust reliability estimates for system components.
- Reconfigure the sustainment network (add or remove repair facilities and/or part suppliers).
- Reallocate inventory, change storage capacity, or change replacement/upgrade production rate.
- Request that component parts be allocated from production to meet sustainment needs.
- Add/remove workforce resources (e.g., field service representatives from the contractor).
- Add training (e.g., so soldiers can perform increased maintenance/repair on line replaceable units).
- Approve/defer technology upgrades (improves short term availability but degrades long term mission support).
- Change assignment of different system configurations to missions.

The simulation module takes these inputs and advances the program state. It returns the high-level program performance noted above, plus lower-level information such as reliability of individual system components, inventory levels and assignments, sustainment network status (e.g., supplier downtimes), on-time delivery of components for upgrade/repair, etc. It should be noted that the logistics experience requires that a new simulation technology be developed, one that will be predictive of sustainment network performance to provide decision support for the learner's configuration of this network during the acquisition phases, plus readjustments afterward.

4.3. Summary

This section has presented a summary of the design for a logistics experience in DoD sustainment, within the framework of an existing technology base for experiential learning environments. Additional work would address enriching the competency model and generic learning concepts for logistics, similar to results for the systems engineering experience. Additionally, feedback to the learner would be developed address the decisions made in terms of addressing challenges and operations of the experience.

5. Evaluation Methods

While the Logistics Experience Accelerator has not yet been implemented, it is important to start planning an evaluation strategy to determine its effectiveness as a learning environment. Lessons learned from current evaluation of the Systems Engineering Experience Accelerator provide a foundation for evaluating the Logistics Experience Accelerator. Typical of many evaluations of educational technology, two types of evaluation are used. Informal evaluation provides quantitative and qualitative feedback intended primarily to learn lessons that can be applied to improving the technology. Formal evaluation, on the other hand, usually focuses on quantitative assessment using statistical experimental design and analysis, often with qualitative feedback to enrich the results.

5.1. Informal evaluation

For the Systems Experience Accelerator, two different types of informal assessment are conducted [17]. The first uses audience participants at tutorial sessions that demonstrate the Experience Accelerator via a walk-through to systems engineers. Participants are asked a set of questions, using a seven-point Likert scale, in three different categories: (i) their perception of the usefulness of the Experience Accelerator, (ii) their perception of the importance of its different components, and (iii) their perception of how well the EA supports various features. In addition, the survey instrument features free form questions for opinions on the best aspect of the Experience Accelerator and the most important desired improvement. Results to date include the following [17]:

- Responses indicate a strong belief in the Experience Accelerator's usefulness, as well as general agreement that that the look and feel is appealing.
- All components of the Experience Accelerator are rated as highly important. These include the supported systems engineering competencies, the Experience Accelerator architecture, the design of the experience, tools for designing experiences, systems dynamics simulation models, and assessment and evaluation.
- All components of the Experience Accelerator were perceived to be largely supported in the current version.

The second uses subject matter experts who have substantial experience in systems engineering leadership positions for DoD programs. They are first asked to undergo the learning experience. Then they are asked to take a more detailed survey than the tutorial participants, also using a seven-point Likert scale. Questions address a number of points, including usability and navigability, authenticity compared to a real DoD program, perceived learning effectiveness of the Experience Accelerator for novices, and their emotional and intellectual engagement with the learning experience. Interaction to date with subject matter experts has resulted in expert knowledge on program characteristics and the mental models used by systems engineers to make decisions in such environments. While both sets of evaluations are on-going, these methods have already proven useful and are likely to be adopted, with necessary domain adjustments, for the logistics version.

5.2. Formal evaluation

Formal evaluation currently is in the design stage for the Systems Engineering Experience Accelerator. There is little published literature on how to evaluate such a learning environment for systems engineering. The current plan is to assess its effectiveness in transitioning novice systems engineers to exhibit behaviors that are more in line with systems engineering experts, both in terms of technical proficiency and professional competency. A similar strategy will be pursued for the Logistics Experience Accelerator. However, one additional goal will be to assess its effectiveness in teaching interaction with the systems engineering function to generate successful outcomes.

6. Conclusions and Future Research

This paper has discussed the design of an interactive learning experience for logisticians in the context of a DoD program (acquisition and sustainment). It utilizes an existing technology infrastructure for hosting learning experiences, and it demonstrates conceptually that this infrastructure is generic and can support different experiences via its phase-based experience formalism. The overall goal is to meet workforce challenges in sustaining today's complex systems, many of which are expected to have longer-than-expected service lives.

Future research consists of implementing, testing and evaluating the learning effectiveness of the logistics experience. In addition, there are plans to enhance the technology infrastructure. These fall into three major categories – additional features, multi-learner capability and experience development tools. Additional features include such items as workplace distractions (e.g., administrative emails) and ability to attend multi-NPC meetings rather than just the current one-on-one learner-to-NPC interaction. Capability for multi-learner experiences is intended to provide multi-disciplinary, team-based training, which would be quite useful for both systems engineers and logisticians. Finally, experience development tools are needed to help foster a community of developers who can use the Experience Accelerator technology infrastructure to build a variety of different experiences to support their own workforce development or other educational needs. These types of tools should provide powerful front-end graphical user interfaces for such tasks as dialog specification, simulation model development and parameter tuning, output report design and configuration management.

Acknowledgements

This material is based upon work supported, in whole or in part, by the U.S. Department of Defense through the Systems Engineering Research Center (SERC) under Contract H98230-08-D-0171. SERC is a federally funded University Affiliated Research Center managed by Stevens Institute of Technology. The authors thank all members of the Experience Accelerator team, as well as the subject matter experts who have provided guidance for authenticity of the Experience Accelerator.

References

1. D.A. Bodner, F. Rahman and W. B. Rouse, Addressing Cost Increases in Evolutionary Acquisition, Proc. 7th Annual Acquisition Res. Symp, Vol. 1 (2010) 329-45.
2. Government Accountability Office, DOD's Efforts to Rebuild Capacity Have Shown Some Progress, GAO-12-232T, Washington, DC, 2011.
3. A. Squires, J. Wade, B. Watson, D. Bodner, M. Okutsu, D. Ingold, et al., Investigating an Innovative Approach for Developing Systems Engineering Curriculum: The Systems Engineering Experience Accelerator, Proc. 2011 ASEE Annual Conf. (2011).
4. A. Squires, J. Wade, B. Watson, D. Bodner, R. Reilly and P. Dominick, Year One of the Systems Engineering Experience Accelerator, Proc. 2012 Conf. on Systems Engineering Res. (2012) 267-72.
5. D. Kolb, *Experiential Learning*, Prentice Hall, Englewood Cliffs, NJ, 1984.
6. M.P. Driscoll, *Psychology of Learning for Instruction*, 3rd ed., Pearson Education, Inc., Boston, 2005.
7. D.W. Shaffer, *How Computer Games Help Children Learn*, Palgrave Macmillan, New York, 2006.
8. J.S. Brown, A. Collins and P. Duguid, Situated Cognition and the Culture of Learning, *Educational Researcher*, 18 (1989) 32-42.
9. J. Herrington and R. Oliver, Critical Characteristics of Situated Learning: Implications for the Instructional Design of Multimedia, in J. Pearce and A. Ellis (eds.), *Learning with Technology*, University of Melbourne, Parkville, Victoria, 235-62, 1989.
10. E.G. Anderson and D.J. Morrice, A Simulation Game for Teaching Service-Oriented-Supply Chain Management: Does Information Sharing Help Managers with Service Capacity Decisions?, *Production and Operations Management*, 9 (2000) 40-55.
11. M. Holweg and J. Bicheno, Supply Chain Simulation – A Tool for Education, Enhancement and Endeavour, *Intl. J. Prod. Econ.*, 78 (2002) 163-175.
12. P. Kaminsky and D. Simchi-Levi, A New Computerized BeerGame: A Tool for Teaching the Value of Integrated Supply Chain Management, in H.L. Lee and N.G. Shu Ming (eds.), *Global Supply Chain and Technology Management*, Production and Operations Management Society, 1998, 216–225.
13. J. Nienhaua, A. Ziegenbeina and P. Schoenslebena, How Human Behaviour Amplifies the Bullwhip Effect. A Study Based on the Beer Distribution Game Online, *Production Planning & Control: The Management of Operations*, 17 (2006), 547-557.
14. J.P. Wade, G. Kamberov, D.A. Bodner, A.F. Squires, The Architecture of the Systems Engineering Experience Accelerator, Proc. 2012 Annual INCOSE Intl. Symp. (2012).
15. A. Squires, J. Wade, P. Dominick and D. Gelosh, Building a Competency Taxonomy to Guide Experience Acceleration of Lead Program Systems Engineers, Proc. 2011 Conf. on Systems Engineering Res. (2011).
16. Q. Fleming and J. Koppelman, *Earned Value Project Management*, 3rd ed., Project Management Institute, 2005.
17. J. Wade, W. Watson, D. Bodner, G. Kamberov, A. Squires, B. Cox, et al. Developing the Systems Engineering Experience Accelerator (SEEA) Prototype and Roadmap – Phase 2 Final Report, unpublished tech. report, Systems Engineering Research Center, Hoboken, NJ, 2012.