



S Y S T E M S
E N G I N E E R I N G
R E S E A R C H C E N T E R

**RT 194: Design and Development Tools for the
Systems Engineering Experience Accelerator – Part 4**

Technical Report SERC-2019-TR-004

May 10, 2019

Principal Investigator: Dr. Jon Wade, Stevens Institute of Technology

Co-Principal Investigator: Dr. Doug Bodner, Georgia Institute of Technology

Research Team:

Stevens Institute of Technology: Peizhu Zhang

Sponsor: Office of the DASD (Systems Engineering)

Copyright © 2019 Stevens Institute of Technology, Systems Engineering Research Center

The Systems Engineering Research Center (SERC) is a federally funded University Affiliated Research Center managed by Stevens Institute of Technology.

This material is based upon work supported, in whole or in part, by the U.S. Department of Defense through the Office of the Assistant Secretary of Defense for Research and Engineering (ASD(R&E)) under Contract HQ0034-13-D-0004 (TO 0046).

Any views, opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the United States Department of Defense nor ASD(R&E).

No Warranty.

This Stevens Institute of Technology and Systems Engineering Research Center Material is furnished on an “as-is” basis. Stevens Institute of Technology makes no warranties of any kind, either expressed or implied, as to any matter including, but not limited to, warranty of fitness for purpose or merchantability, exclusivity, or results obtained from use of the material. Stevens Institute of Technology does not make any warranty of any kind with respect to freedom from patent, trademark, or copyright infringement.

This material has been approved for public release and unlimited distribution.

TABLE OF CONTENTS

Table of Contents	iii
List of Figures	iv
List of Tables	iv
Executive Summary	1
Introduction	2
Critical Need	2
Experiential Learning and the Systems Engineering Experience Accelerator	2
Research Overview	4
Objectives.....	4
Proposed Work.....	5
Vignette Experiences	7
Vignette Selection.....	7
Research Completed	8
Simulation Modeler Extensions	13
Prior Work.....	14
EA – Simulator Interface.....	14
State-Chart Simulation Support	17
Unit Testing and Integration Testing.....	21
Learning Assessment Tool Extensions	21
Prior Work.....	21
Functionality Updates	22
Unit Testing and Integration Testing.....	25
Infrastructure Upgrade	25
Prior Work.....	25
New Functionality.....	26
Unit Testing and Integration Testing.....	26
Conclusion.....	27
Appendix A: List of Research Related Publications.....	29
Appendix B: Cited and Related References	31

LIST OF FIGURES

Figure 1. Relationship between tools and SEEA components	3
Figure 2. Wright Brothers' Experience Main Interface	9
Figure 3. Wright Brothers' Experience Budget Allocation Exercise	10
Figure 4. UK MoD Experience Main Interface	11
Figure 5. HMS Tempest Experience Start Phase.....	12
Figure 6. New Briefing Page	13
Figure 7. Program state-chart.....	19
Figure 8. Simulation interface.....	20
Figure 9. Main View of the Learning Assessment Tool.....	22
Figure 10. Learning Assessment Tool Class View.....	22
Figure 11. Behavioral Analysis Tool	24
Figure 12. Assessment Engine Performance Comparison	25

LIST OF TABLES

Table 1. Learning Vignette Decision Matrix.....	7
Table 2: Interface Components and Functionalities.....	16
Table 3. SEEA tool development plan	27

EXECUTIVE SUMMARY

The Department of Defense faces many challenges related to deploying new systems to meet the needs of the warfighter. Among these are compressed development times and increased required capabilities. There is a real need for systems engineering expertise in the acquisition enterprise to guide the technical design, development, integration and testing of such systems.

At the same time, there is a potential skills gap due to an aging workforce, many of whom are near retirement, and the experience curve needed to bring new talent up-to-speed. Often, this experience curve takes many years and spans multiple programs. The idea behind the Systems Engineering Experience Accelerator is to accelerate the experience curve so that new talent can become proficient much more quickly to meet DoD needs. This is done by using educational technology, and specifically the notion of role-playing in immersive environments where a learner not only learns the technical skills associated with a systems engineering position, but also critical skills in persuasion and decision-making needed to deploy technical skills effectively.

The original SEEA was developed to let the learner play the role of a chief systems engineer in an acquisition program for a new unmanned aerial vehicle system. It was developed using standard programming tools. Clearly, though, there are many different systems engineering skills and competencies needed beyond one such educational experience. The goal then became to develop a library of such experiences. To do so quickly and cost efficiently, it was determined that a higher-level set of tools were needed to support development, especially tailored to individuals without substantial programming expertise. Thus, an effort was initiated to create a suite of such tools.

This report details the fourth part of development of these tools. They consist of three suites of tools. Simulation Modeler enables an experience designer to create and tune simulation models and output charts that advance the state of a simulated world with which the learner interacts in the role of a systems engineer. Experience Builder lets the designer specify the different learning phase, cycles, events and communications experienced by the learner. Learning Assessor lets the designer determine what learning data to capture and how to analyze it to determine the effectiveness of different learning experiences.

INTRODUCTION

Educational technology is used increasingly in a variety of domains, ranging from high school and college-level courses, from on-the-job training to advanced skills development. One of the appealing features of education technology is that it can be used not only to convey the material being taught, but also the context in which the material should be applied. This can be accomplished through role-playing environments in which a learner interacts with an automated system providing the context, or it can be accomplished through technologies that connect multiple learners who interact to solve problems and apply solutions in context. Systems engineering is one field for which educational technology has potential for application.

CRITICAL NEED

Systems engineering is a multidisciplinary practice and is as much of an art as it is a science. While a waterfall model of education can provide a background of domain-centric knowledge, it is not until this knowledge is put into practice in an integrated, real world environment that a systems engineer can develop the necessary insights and wisdom to become proficient. In the workplace, these learning events are often distributed sparsely over time such that an engineer may only see a complete system life cycle over a period of several years. As a result, the maturation time from completion of formal studies to becoming a seasoned systems engineer is unacceptably long, particularly when contrasted with the clock speeds of today's society in which career change is the norm rather than the exception, particularly among the young.

Clearly, there is a critical need to promote rapid skill development of systems engineers, in particular senior systems engineers, across the DoD and government workforces, as a large cohort of personnel is nearing retirement age. In addition, new systems engineering skills are needed to address important societal needs in national security, homeland security, airspace management and disaster recovery. These domains involve large-scale, systems-oriented solutions with increasingly limited budgets. Educational technologies hold the promise of providing customized learning exercises based on real-world situations to reduce the reliance on extensive on-the-job training that is the hallmark of current workforce development.

EXPERIENTIAL LEARNING AND THE SYSTEMS ENGINEERING EXPERIENCE ACCELERATOR

The Systems Engineering Experience Accelerator (SEEA) project created a new approach to developing the systems engineering workforce which augments traditional, in-class education methods with educational technologies aimed at accelerating skills and experience with immersive simulated learning situations that engage learners with problems to be solved. Although educational technology is used in a variety of domains to support learning, the SEEA is one of the few such technologies that supports development of the systems engineering workforce.

While the existing technology infrastructure and experience content is useful, it is limited in its ability to support a community of educators and developers. The SEEA was developed with a

goal of transitioning to an open-source sustainment model that will provide long-term support for a community of educators and learners in creating learning exercises to address their specific needs. Currently, it is difficult to design and develop educational content without significant knowledge of the SEEA design.

In Parts 1-3 of the EA Tools research task (RT-123, RT-146, and RT-164), the efforts focused on developing a set of tools specifically for educators and developers outside the SEEA research and development team, to support their design and development of learning modules for their use. Each of the tools maps to a particular set of components in the overall SEEA architecture, as shown in **Error! Reference source not found.**

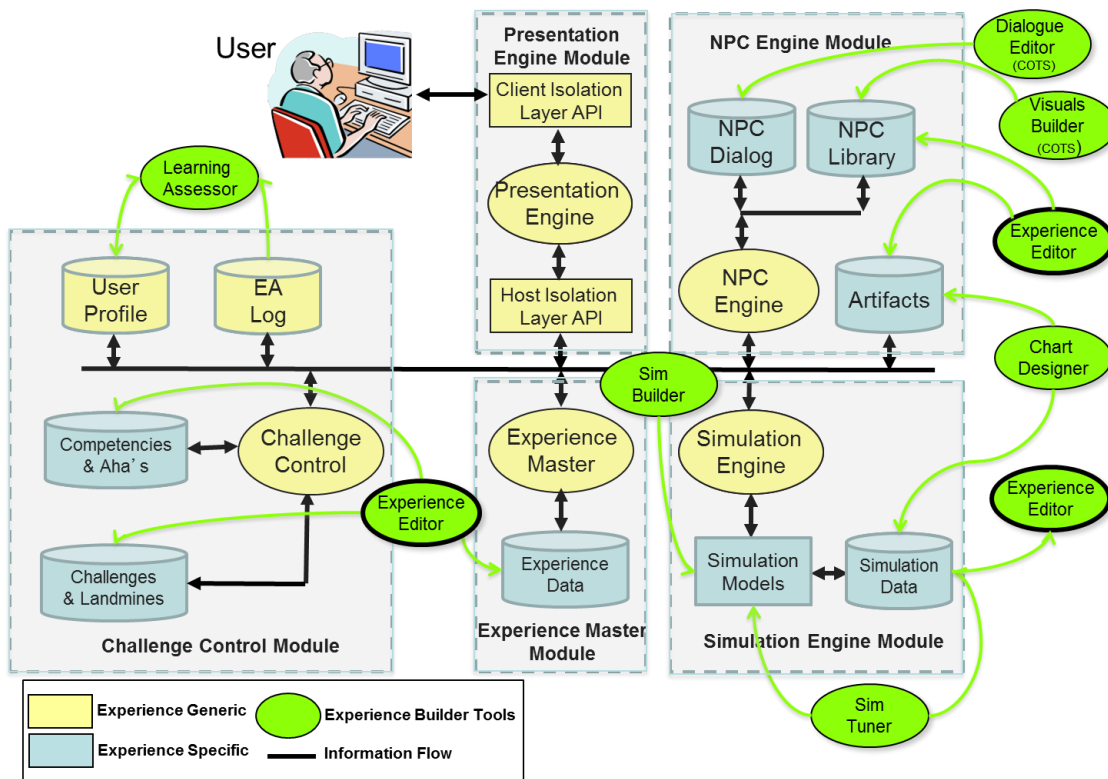


Figure 1. Relationship between tools and SEEA components

For instance, the simulation tools map to the simulation models and chart outputs (artifacts). The experience builder tools map to the experience master, NPC library and dialog, and artifacts. The tools development concentrated on the following set of tools prioritized by the likelihood of having the most impact on facilitating module development:

- **Simulation Modeler** – Includes the Sim Builder, Sim Tuner and Chart Designer
 - **Sim Builder** - Simulation model builder using libraries/templates
 - **Sim Tuner** - Parameter tuner that automates the tuning of parameters to yield desired outputs via batch processing of different combinations of settings

- **Chart Designer** – Specification GUI that allows the experience designer to specify simulation output charts
- **Experience Builder** – Integrates the Phase and Event Editor, and the Artifact Integrator.
 - **Phase Editor** - GUI-based tool for phase, cycle and event specification, with code generation
 - **Event Editor** - GUI or text-based tool to specify events and their triggers, with code generation
 - **Artifact Integrator** - Application that allows designer to take artifact files, such as design documents and enter it into EA application with automatic recompilation and re-linking
- **Learning Assessor** - Assessment tool-suite that provides automated performance scoring and decision comparisons against proven baselines

The remainder of this report is organized as follows. The research hypothesis is discussed first, along with measurable outcomes and the overall research goal. Then the vignettes are discussed. This is followed by a discussion of the updates made to each of the tools – Simulation Modeler and Learning Assessor. Afterward, the upgrades made to the SEEA infrastructure are described. The evaluation of the tools is provided next. Finally, the Conclusions and future research are presented.

RESEARCH OVERVIEW

OBJECTIVES

This project seeks to create a set of tools to enhance capabilities to rapidly and cost-effectively create experience modules that can be used by the current generic Experience Accelerator technology set to deliver a variety of learning experiences tuned to the needs of particular workforces. The following is the Problem Statement, Hypothesis, Measurable Outcomes and Research Goal for this project.

Problem Statement: *Traditional systems engineering education is not adequate to meet the emerging challenges faced by the Department of Defense and others in system acquisition and sustainment. New educational technologies such as the Systems Engineering Experience Accelerator hold the promise of facilitating a rapid skill and experience accumulation for the workforce to meet these challenges. However, to have scalable effect, such technologies cannot rely on extensive programming and low-level development to create a rich set of experiential learning modules needed to accelerate systems engineering workforce development.*

Hypothesis: *The Experience Accelerator technology will scale to support a community of developers engaged in creating modules for their organizations' use if tools are developed that allow educators and other non-programmers to create, maintain and evolve experiential learning modules.*

Measurable Outcomes: *The outcomes from this research will be measured in two main ways. First, educators and others interested in creating experiential learning modules will provide qualitative feedback on the effectiveness and efficiency of the Experience Accelerator toolset in creating experiential learning modules based on their use of the design and development tools. Second, the number of such developers who commit to create modules for their organizations' use will be tracked, as well as the number of organizations and variety of different application areas.*

Research Goal: *Validate the hypothesis through the creation of design and development tools for experiential learning modules that maximally leverage the current Experience Accelerator research, technology and content.*

While these tools have been developed and deployed in prototype form, additional research is needed to evaluate their effectiveness and extend their functionality. The original experience developed for the SEEA was a multi-phase experience in which a learner assumes the role of chief engineer of a program developing a new unmanned aerial vehicle. This experience lasts from shortly after preliminary design review until low-rate initial production (if successfully completed). This provides perspective on a major portion of a system lifecycle.

However, the SEEA user community has expressed interest in experiences that are more tightly focused on particular aspects of a systems engineering development effort. In addition, the UAV experience has a large programmatic component. While recent work added a technical trade study to the experience, the community has also expressed interest in experiences that emphasize the technical aspects of systems engineering. By partnering with institutions that can host such experiences, the Project Part 4 goal is two-fold:

- (i) to evaluate the existing toolset via development of several focused experiences, and
- (ii) to instantiate an “experience lifecycle” whereby experiences are conceived, developed, deployed and curated within a partnerships of subject matter experts, host institutions, developers, teachers, educational technology specialists and others.

Finally, two needs for additional tools have been identified. The first is in the area of modeling support for simulation beyond the currently supported system dynamics approach. It is critical that other types of simulations be supported by the SEEA. The second is in the area of learning assessment and analysis. The previous work needs to be extended in this important area.

PROPOSED WORK

Part 4 concentrates in two main areas – (1) the development of complete condensed learning experiences using the EA Tool Set and (2) enhancements in specific, critical EA Tool Set areas.

The first area is to use the tools to develop a number of short duration, learning “vignettes” that are targeted at specific learning objectives that can be administered as online single learner experiences. The sponsor would target their usage outside of a standard course as a standalone educational experience.

The following are potential candidates for this work:

- Wright Brothers experience

- UK MoD modification
- Mini-experiences extracted from the existing UAV experience:
 - Stand-alone trade-study (from current UAV experience)
 - Readiness decision for CDR
 - Experience focused on understanding TPMs
- Leverage work from ongoing third-party development efforts (e.g., AFIT)

The following is a description of the work items for this effort:

1. Development of “Vignette” Experiences
 - a. Engage learning objectives team (user institutions) and experience concept team (SMEs)
 - b. Select targeted set of experiences from list
 - c. Identify learning objectives
 - d. Storyboard experience concepts
 - e. Develop experiences
 - f. Test and validate experiences
 - g. Deploy experiences
 - h. Assess learning against learning objectives for experiences
 - i. Archive experiences in library
 - j. Provide evaluation of tools based on their use in the vignette development
 - k. Identify opportunities for new tools

The second area is the extension of the existing tools in two specific areas. The first focus is to create new functionalities for the learning assessor tool. As more experiences are developed with differing learning objectives, it is critical to support new functions to assess learning against these objectives. In addition, this work will be critical for learning assessment for online experiences in which there is limited interaction between learners and instructors. The second focus area is to extend the simulation functionality within the SEEA to encompass simulation methods beyond the existing system dynamics approach. This is essential as new experiences are developed that require simulation capabilities that are not compatible with systems dynamics approaches.

The following is a description of the work items for this effort:

1. Enhance existing tools
 - a. Extend learning assessment tool
 - i. Develop a performance assessment engine which evaluates a learner’s competency by comparing their performance to an experts’ performance and by comparing their performance against historical data.
 - ii. Add a function to generate an objective score based on the experience performance and decision-making process.
 - b. Extend simulation capability to methods beyond system dynamics
 - i. Create and document an interface between the EA and generic simulators
 - ii. Develop a tool for specifying state-chart simulations (including XML specification for models)
 - iii. Provide demonstration capability for state-chart simulations in the simulation execution engine using the newly developed interface

VIGNETTE EXPERIENCES

VIGNETTE SELECTION

Due to the limitation of resources, only a sample of learning vignettes will be implemented. A rating process has been conducted by the team to determine the list of learning vignettes for implementation. The rating process focused on the following factors:

- **Audience** – Has a large audience who would be willing to participate in the experience. Systems and Software Engineers, at any level from novice to expert.
- **Learning Value** - How well does it teach important SE capabilities? Does it impart other learning?
- **Time Efficient** – Needs to be effective in less than 90 minutes.
- **Realistic** - How technically relevant is the trade? Is it realistic?
- **Fun Factor** - How interesting is the trade from the student's perspective?
- **Ease of Implementation** - How easily can the idea be incorporated into the existing experience?

The rating for each factor is on a 1-5 scale, where 1 means poor and 5 means very high. Each factor has a weight of 0-10 based on its importance. The team rated the six candidate learning vignettes based on these factors and ranked the experience from highest to lowest. These ratings are based on the team's best judgement. The results are shown in Table 1 below. The top three candidates are planned for implementation.

Table 1. Learning Vignette Decision Matrix

	Audience	Learning Value	Time Efficient	Realistic	Fun Factor	Ease of Design	AVG	Weighted Total
Wright Bros	5	4	4	5	4	3	4.17	204
UK MoD	4	3	5	4	4	5	4.17	195
Systems Thinking	5	NA	5	NA	3	3	4.00	193.5
Robot Game	4	4	3	2	5	2	3.33	168
EA: Readiness for CDR	4	5	1	5	3	2	3.33	167
EA: Trade-Study	2	4	4	4	2	4	3.33	156
EA: TPM	1	2	2	3	2	2	2.00	93

The weightings for each factor are as follows:

Factor	Audience	Learning Value	Time Efficient	Realistic	Fun Factor	Ease of Implement
Weight	10	10	8	7	8	5

For the learning vignette candidates, learning objectives are identified through the rating process. For the top three candidates, the learning objectives are as follows:

- Wright Brothers Experience
 - Manage risks and options
 - Trade study between different benefits and c
- Systems Thinking Game
 - N/A
- UK MoD Experience
 - Familiarize mission critical communication
 - Conduct individual investigation through communication to find underlying issues
 - Prepare investigation report to supervisor with discoveries to backup claims
- Robot Game
 - Conduct trade study on different options
 - Use systems thinking
- EA: Readiness for CDR
 - Understand critical design review
 - Familiarize the CDR process
 - Manage KPP and schedule for CDR preparation
- EA: Trade Study
 - Conduct trade study on UAV actuators
 - Develop trade study matrix
 - Determine the best option
- EA: TPM
 - Understand total productive maintenance
 - Develop TPM plan

The final list of learning vignettes planned for implementation are: Wright Brothers Experience, Systems Thinking Game, and UK MoD Experience.

RESEARCH COMPLETED

Wright Brothers Experience

A preliminary prototype of the Wright Brothers Experience has been created. The experience is based on an existing assignment used by the Defense Acquisition University (DAU). The DAU assignment focuses on how the Wright Brothers modified the designs of their existing fliers to meet requirements for a new flier that would be purchased by the U.S. Army. The experience lasts five-month in-game time for the development process.

In the assignment, student learners are given a workbook with various artifacts (e.g., original values for KPPs/TPMs and the relationships between them, potential schedules and budgets, etc.). They are then asked to examine these artifacts and answer a series of questions.

As a five-month long experience, the whole story can be divided into five phases with each phase representing one month of development time for the Wright brothers. Student learners will be required to go through the artifacts and then make decisions on design changes. At the beginning of each phase, learners can retrieve important information about the project and the army's requirements by examining the updated artifacts, NPC dialogues and flying machine status. They then need to make decisions on possible changes to be made to the project. After each phase, the learner's input will be input into the simulator which will then be updated to the new status of the project and attributes of the flying machine. Figure 2 shows the main interface for the Wright Brothers' Flyer experience.

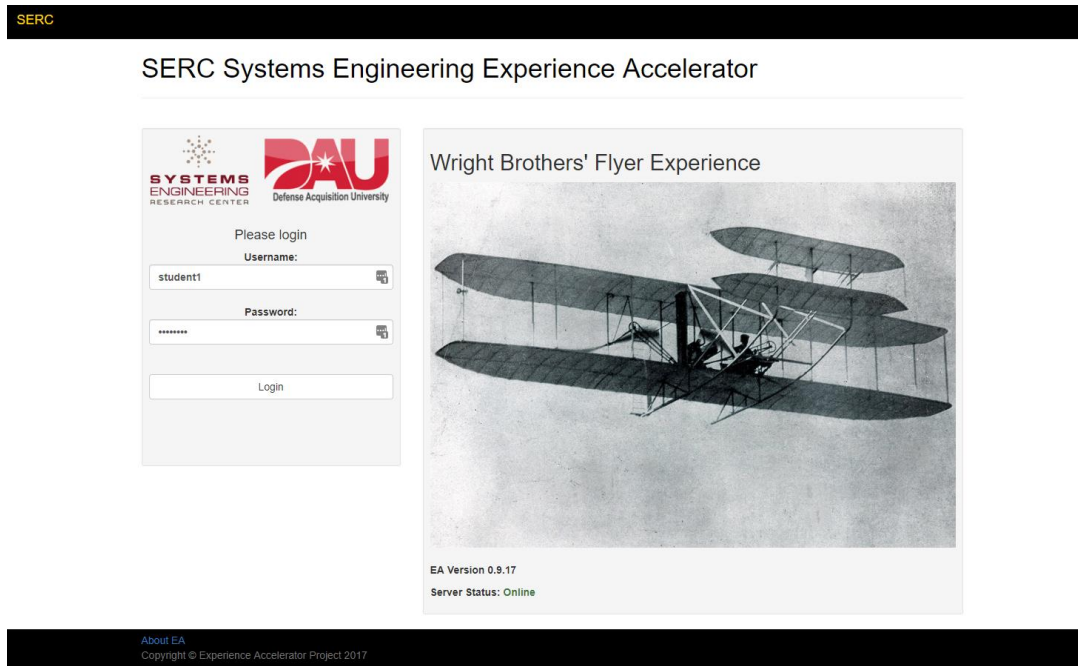


Figure 2. Wright Brothers' Experience Main Interface

At the beginning of the experience, a number of questions were asked to the students about the scope, risks technical measures and the schedule. Figure 3 shows the budget allocation exercise of the Wright Brothers' Flyer experience that learners need to work on.

SERC Home Library FAQ Meetings Mails Notification (0 new) Fast Forward Next Stage Quit

Allocate the Budget

Orville's Time-Phased Budget

	15-Mar-08	31-Mar-08	15-Apr-08	30-Apr-08	15-May-08	31-May-08	15-Jun-08	30-Jun-08	15-Jul-08	31-Jul-08	15-Aug-08	31-Aug-08
PMB (cum)	\$0	\$1,617	\$5,628	\$12,364	\$14,567	\$15,200	\$15,776	\$16,393	\$16,923	\$17,528	\$18,392	\$19,563
CCB	\$25,000	\$25,000	\$25,000	\$25,000	\$25,000	\$25,000	\$25,000	\$25,000	\$25,000	\$25,000	\$25,000	\$25,000
MR	\$5,437	\$5,437	\$5,437	\$5,437	\$5,437	\$5,437	\$5,437	\$5,437	\$5,437	\$5,437	\$5,437	\$5,437

This is Orville's budget plan. Do you see any problems with it? If you think improvements could be made, what information would you need to be able to improve it? WHY?

Problems, improvements and why

[Submit](#)

About EA
Copyright © Experience Accelerator Project 2017

Figure 3. Wright Brothers' Experience Budget Allocation Exercise

UK MoD Submarine Experience

The UK MoD experience focuses on a submarine maintenance scenario where the learner is responsible for finding issues with the submarine by communicating with other personnel in the team. Failing to communicate with peers in a timely fashion results in a catastrophic outcome. Since this experience design does not require a simulation engine, experience creation required only the utilization of the experience building tools. The experience has been updated and improved to be a standalone experience ready for deployments. Figure 4 shows the main interface for the new UK MoD Submarine Experience.

SERC Systems Engineering Experience Accelerator

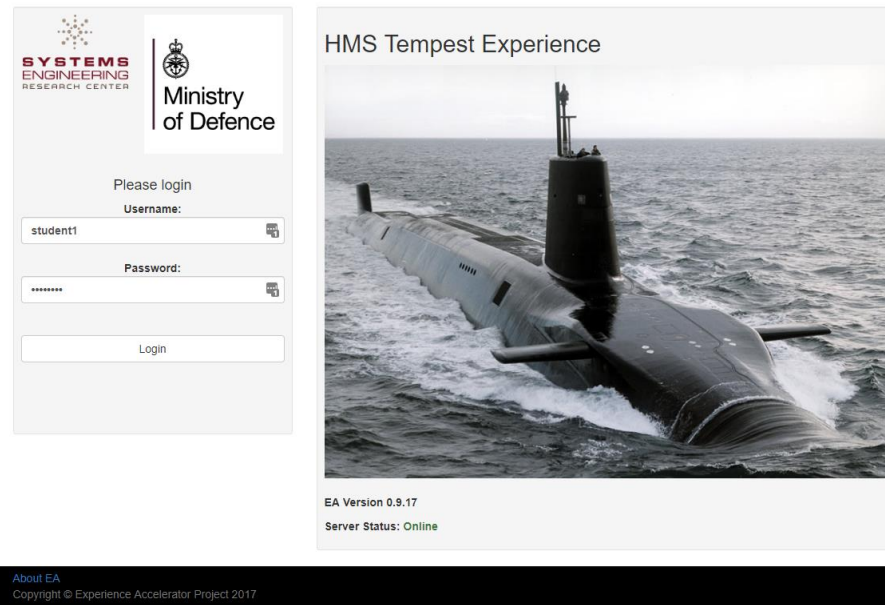


Figure 4. UK MoD Experience Main Interface

The experience is divided into Five phases:

1. New User Orientation Phase – welcoming the learner and provides the learner background information about the situation.
2. Start of Experience Phase – introducing the learner to the situation and asks the learner to make quick decisions based on limited information provided.
3. Diagnosis Phase – learner investigates the potential issues by communicating with peers in a timely and appropriate manner.
4. Reporting Phase – learner report to supervisor the findings and recommendations regarding the issues and solutions.
5. Debriefing Phase – depending on the learner action, different situation will happen and learner receives feedbacks about his/her experience performance.

For the new user orientation phase, an email is sent to the learner requiring him/her to read through five technical documents and proceed. The Artifact Integrator was used to convert and integrate the PDF formatted documents. The email was composed in the Artifact Integrator and then referenced from the Event Editor where the email event was created. One limitation of the tools is that they currently do not support the creation of user interface elements, therefore some coding efforts were needed to create a proceed button on the user interface (UI). The experience has been updated to use the HTML5 engine for better compatibility and usability.

Figure 5 shows the start of the investigation phase where a message directing the tasks for the learner is displayed.

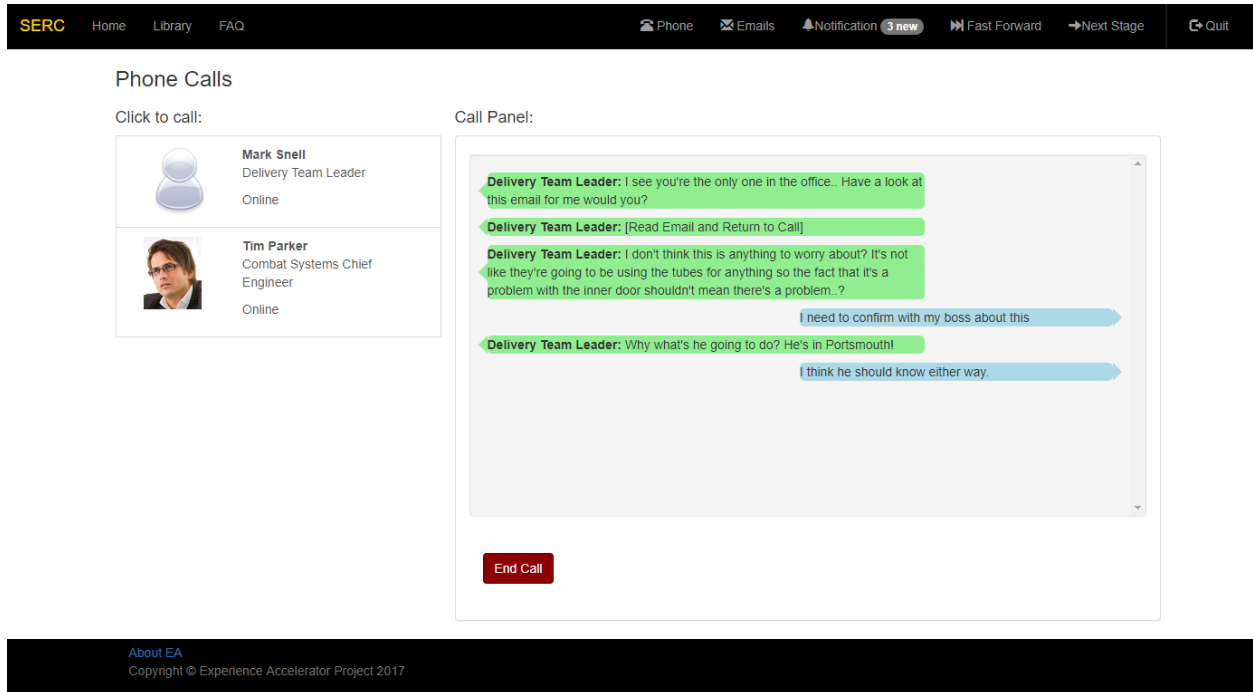


Figure 5. HMS Tempest Experience Start Phase

As the learner concludes the experience, a news article is presented to the learner representing the consequences of the learner's decisions. Figure 6 showing the News Briefing Page from the Debriefing Phase.

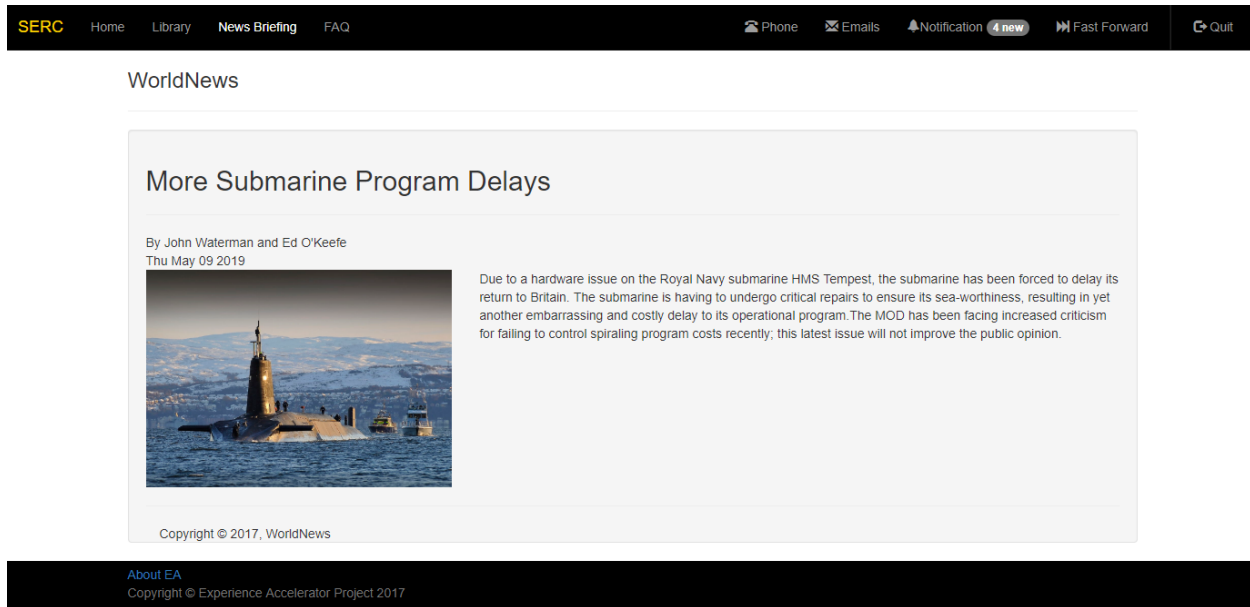


Figure 6. New Briefing Page

The UK MoD Submarine experience has been deployed to an online server for external access. The temporary server IP address is: <http://192.241.166.160>.

Systems Thinking Game Experience

Due to time constraints, the Systems Thinking Game Experience was designed and prototyped, but was not fully implemented.

SIMULATION MODELER EXTENSIONS

The Simulation Modeler provides tools for the experience designer to create simulation models used in the SEEA. These simulation models are executed by the SEEA to advance the state of the simulated world in time. For instance, in the UAV experience, the simulated world consists of the development program. The simulation models represent this world and advance its state over time through execution of one of the models.

The learner has control over some of the variables in the simulation model. At decision points during the experience, the learner enters recommendations, which may change the values of variables or parameters in the simulation. Such changes alter the state trajectory of the simulated world. Ideally, such changes would address perceived problems in the simulated project. However, the learner may discover that the problems are not addressed adequately, or that new problems are created.

The simulation models are based on the systems dynamics paradigm of simulation (Forrester, 1961; Sterman, 2000). In system dynamics, a flow system is used to model state changes in the world. Rates at which flow occurs are governed by equations. These equations can be used to

model positive feedback reinforcing loops, in which a particular phenomenon may grow. Likewise, they may be used to model negative balancing feedback loops, in which a steady-state is achieved. In addition to feedback loops, such phenomena as lags and communication overhead can be represented.

System dynamics has been applied a variety of domains. Relevant to the SEEA are domains such as project management and earned value management (Kefalas, 1998; Madachy, 2008).

PRIOR WORK

Previously, two main tools were developed under the Simulation Modeler (Wade et al., 2015; Wade et al., 2016). The Sim Builder provides a graphical user interface for building systems dynamics models. Prior work focused on improving the existing tool by adding sub-models and other features to enable model modularity and reuse through model component libraries. The Sim Tuner allows the experience designer to execute simulation models outside of the SEEA to see how they behave over time. Variables and parameters can then be adjusted to achieve desired behavior. In addition, the designer can take the role of the learner and test different possible learner recommendations to see how they impact the overall performance measures of interest in the experience.

In addition to the Sim Builder and Sim Tuner, a prototype tool was developed to allow the experience designer to specify charts. This tool, the Chart Designer, was matured previously (Wade et al., 2017).

EA – SIMULATOR INTERFACE

Previously, the interface between the main Experience Accelerator engine and the simulator has been defined in ad-hoc terms based on the systems dynamics implementation of the simulator. This interface has consisted of four main components:

- The run-time parameters needed to execute the simulator for a particular cycle, resulting in advancement of the experience state for the learner. These currently include:
 - Simulation executable (jar file that executes the experience state advancement)
 - Simulation model file name (containing XML specification of systems dynamic model that corresponds to the experience state and advancement rules)
 - Input file name (to which the Experience Accelerator engine provides learner-specified variable values from the recommendation form)
 - Run length of the cycle in time units (currently days)
 - Output file name (containing the time-stepped values of simulation variables recorded as the state advance mechanism is executed)

- Parameters representing the phase identification and the cycle identification within the particular phase for the experience
- Variables that the learner can change via the recommendation form, resulting in modifications to the experience state that are promulgated through the state advancement. The variables are coded in the Experience Accelerator components that access the recommendation form when the learner makes his or her selections and then writes the values to the input file read by the simulator execution code.
- The experience state, which is the output from a particular cycle of the simulation state advancement. This is currently in CSV format with each variable having its value recorded in time-stepped fashion in a particular column. The Experience Accelerator can then access this to extract current state values as represented by simulation variables.
- The charts that provide visualizations of the experience state over time for key performance indicators. The simulator engine generates these as image files for use by the Experience Accelerator.

We have designed the following more generic and formalized interface for simulators. First, we note that the interface should support two audiences – the instructor and the learner. Currently, the interface mainly supports the learner. There is no mechanism for an instructor to interact with the Experience Accelerator to modify any of the simulator’s execution or output. An instructor, for instance, may want to parameterize a simulation model to support different difficulty levels or different learning objectives. Currently, such instructor support would need to be achieved by having different simulation models developed for each difficulty level of set of learning objectives.

In addition, we note that there are four main simulation paradigms that are potentially of interest in Experience Accelerator applications. These include system dynamics (SD), discrete event (DE), agent-based (AB) and state-chart-based (SC). There are two main drawbacks of the current interface with respect to these paradigms.

The first is that the current simulation model is largely hard-coded, with values for parameters and variables specified within the model file. Typically, simulations read input files, either spreadsheets or databases, to populate parameters and variables within a model. For example, an agent-based simulation may read a table from a database to create a population of agents. A discrete-event simulation may read from a spreadsheet a set of time-stamped transaction arrivals. Using different input datasets, rather than varying parameters and variable values individually, can be a powerful way to parameterize experiences for different difficulty levels or learning objectives.

The second relates to the discrete nature of events that happen in an experience. This is especially relevant to the discrete-oriented simulation paradigms (discrete-event, agent-based, and state-chart-based). There is currently no method in the EA-simulator interface to schedule

an event to occur at a point in time during cycle execution. The current interface supports changing a variable at the beginning of a cycle only. Thus, a landmine or challenge cannot easily be introduced in the middle of a cycle.

We introduce the following interface components to overcome the various limitations described above, which are shown in Table 2.

The first component is the instructor specification interface. The instructor would use this to set the difficulty level of the experience, customize it to meet certain learning objectives, or provide different data-driven scenarios for learners.

Table 2: Interface Components and Functionalities

Interface component	Functionalities
Instructor scenario specification	<p>Read input file with values of variables</p> <ul style="list-style-type: none"> • Supportable by input file of existing SD simulator • Supportable by AB, DE and SC simulations, subject to their having a read file functionality • Would require generation of input files for different scenarios
Specification of future landmine events	<p>Schedule an event</p> <ul style="list-style-type: none"> • Supportable in existing SD simulator mid-cycle by dividing the cycle execution into two commands and running the second execution with an input file having the new value of the variable to be changed • Supportable in SC and DE simulations by calling method to schedule event on event list • Supportable in AB simulations by calling method to invoke agent behavior invoking the desired event given that a specialized agent is modeled in the simulator to allow this
Run time parameters	<p>Provide simulator with needed execution information, including executable command, model file, input data file, output data file, phase/cycle parameters</p> <ul style="list-style-type: none"> • Supported in existing SD simulator • Supportable in AB, DE and SC simulators. However, it should be noted that the phase/cycle nature of execution makes saving state information between cycles necessary in simulations. The SD simulator does this. However, each individual simulator must be verified to see whether this is supported or needs to be developed.
Experience state	<p>Provided access to simulation output for visualizations, NPC dialog, status displays</p>

	<ul style="list-style-type: none"> • Supported in existing SD simulator by time-stamped CSV variable table • Supportable in AB, DE and SC simulators, but in format of (time, value) for each variable. Discrete-event oriented simulators do not execute at each time-step, but rather they often skip directly to the next scheduled event on the event list.
Learner-changeable variables	<p>Provide access to variables that can be changed to implemented learner decisions/recommendations</p> <ul style="list-style-type: none"> • Supported by SD simulator, since variables are coded as learner-decidable or not. Recommend table that links variable name to recommendation stored in interface. • Supportable in AB, DE and SC simulators, but depend on simulator implementation.
Status charts	<p>Provided visual output of status and advancement</p> <ul style="list-style-type: none"> • Supported in existing SD simulator • Supportable in AB, DE and SC simulators, but would need to be programmed using the output data as input

STATE-CHART SIMULATION SUPPORT

State-chart simulations are frequently used in systems engineering during the design of a system when the system's dynamic behavior is of interest. One of the foundational representations of state-charts is provided by Harel (1987).

In state-charts, a system or sub-system operates in discrete states. Transitions between those states occur as events. A simple machine can be represented using "off" and "on" states. More complex behavior can be introduced with such states as "failed" or "set-up." From a modeling perspective, the task is to identify the various states of interest and feasible transitions between them.

A state-chart typically has an initial state into which the model is set upon start. A state may have a pre-defined duration, after which it transitions to another state. Alternatively, an outside event may cause a transition, for example an operating turning a machine on. Transitions may have guard conditions that prevent their occurrence if some condition is not met. States may have events that are triggered upon entry to the state or exit from the state.

States may be in the form of composite states. For instance, a state may have multiple states encapsulated within it. That is, when a system enters a composite state, it enters one of the

encapsulated states. It may transition between these encapsulated states while remaining in the composite state. Exit from the composite state implies exit from the encapsulated states.

States may also be in the form of parallel states. A system may be in multiple states simultaneously. Often, these multiple states apply to different sub-systems.

The World Wide Web Consortium (W3C) has set the State Chart Extensible Markup Language (SCXML) as a standard for representing state-charts (W3C, 2019). One of the primary implementations of open-source state-chart models is the Apache Commons SCXML (Apache, 2019). Both of these are current as of 2015. This effort uses the Apache Commons toolkit to provide state-chart simulation functionality.

Apache Commons SCXML consists of three primary components:

- The engine provides a set of Java classes that enable a generic event-driven state machine to execute
- The domain provides the context for the state-chart model, namely the system being modeled. It consists of an XML-based state-chart representation and a Java-based behavioral representation that augments the state-chart.
- The bridge provides a two-way communication link between the two so that state machine execution is enabled for the particular domain.

This effort has created a state-chart simulation augmenting these tools for a military acquisition program for a next-generation jet fighter. One of the primary tool additions is the simulation infrastructure in the form of an event calendar to enable scheduling of events within state-chart simulations. This is critical for simulations, as events comprise the dynamic behavior of the simulation.

The acquisition program being modeled executes in a series of phases. Each phase is modeled as a state. Transitions occur as the program successively moves through its various phases, as shown in Figure 7. The program design state is the initial state. In this state, various decisions about how the program will operate are undertaken. These include whether the program will use a prototype or not, and whether the program will employ concurrency. The program will make several variants, so another decision is which variant the program will focus on first.

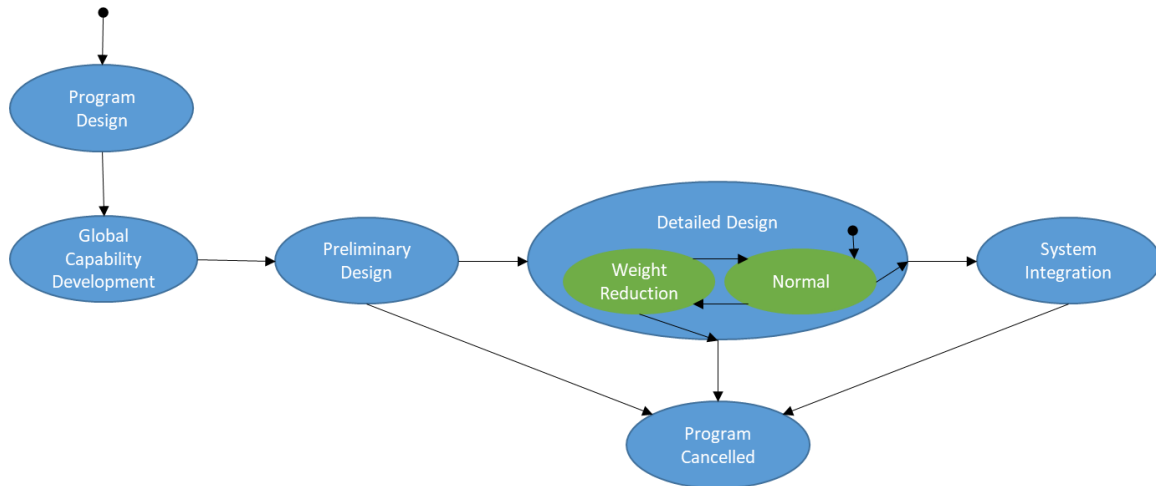


Figure 7. Program state-chart

Once program design is complete, the program enters a global development capability phase in which the various contractors collaborate with the lead systems integrator to develop the infrastructure needed to design and develop the fighter system. Once this is achieved, the program enters a preliminary design state that, if successful is capped by a preliminary design review (PDR). If successful, it transitions to a detailed design state. Detailed design is a composite state. The program may be in a normal state within detailed design, or it may be in a weight reduction state, in which there is active effort to reduce the estimated aircraft design weight. A successful detailed design state results in a critical design review (CDR). The program then transitions to a system integration phase.

It should be noted that the program can also transition to a Program Cancelled state from any of preliminary design, detailed design or system integration.

Figure 8 shows the interface of the state-chart simulation. The learner can select from the three program design decisions. Once “Program Design Complete” is selected, the simulation starts and executes through the various phases. The current phase (or state) is shown highlighted at the bottom. It should be noted that minimal effort has been put into the interface. When integrated with the Experience Accelerator, the interface will be part of the Experience Accelerator. Thus, the program design choices would be implemented as something similar to a learner recommendation form.

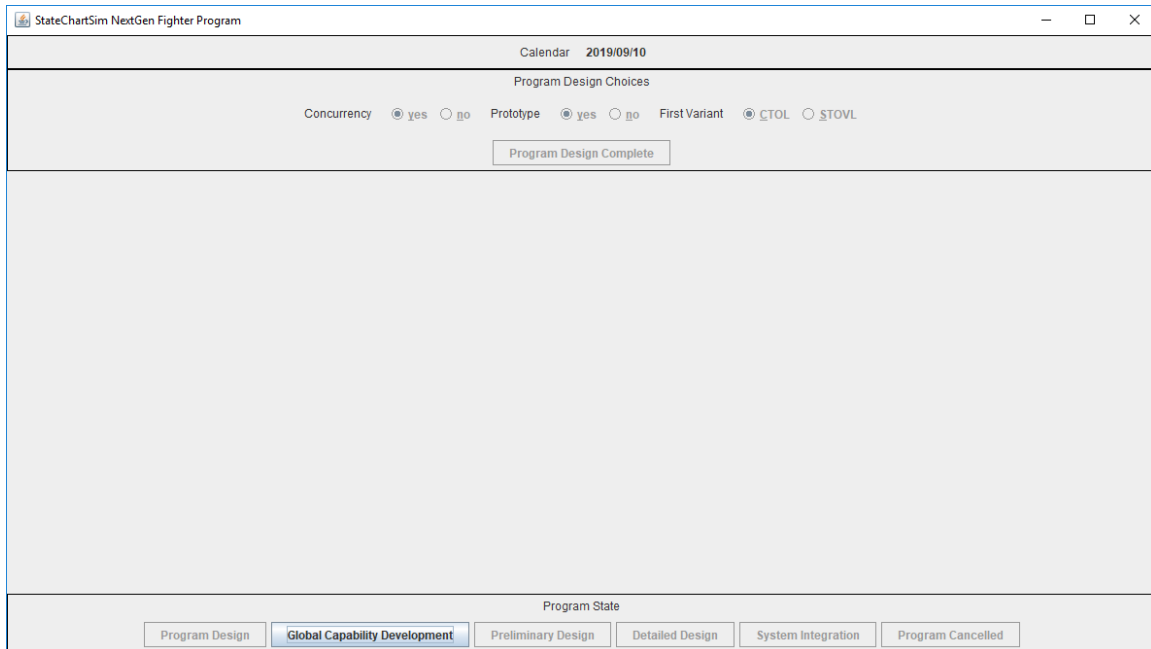


Figure 8. Simulation interface

The simulation consists of the following components:

- Program state-chart specifies the states and the transitions between them, as well as behavior methods to execute on entry and on exit from the states. This is implemented as an XML file.
- Program behavior and attributes are implemented in a program class. These include the methods that are invoked when the program enters or exits a state. They also include such attributes as weight and cost and how they change over time. These attributes also include the program design attributes decided by the learner, and the program design attribute values affect how weight and cost grow over time. State transition events can be scheduled based on values of these attributes. Thus, the attributes can determine whether the program advances to its next phase or is cancelled.
- The user interface is implemented as a Java class, and it provides the learner with basic information about the simulation and recommendation input. It also provides the user with the ability to transition between states.
- The event calendar infrastructure needed to schedule simulation events.
- The state chart execution and bridge are supplied by SCXML classes, and these interface with the domain components above.

UNIT TESTING AND INTEGRATION TESTING

During the development of the Simulation Modeler tools, unit testing and integration testing were performed. Unit testing was performed when a new feature was added to ensure that it performed as intended. Integration testing was performed, as well, to ensure that changes did not adversely affect existing functionality in the tools. During this process, a variety of defects were identified and remediated.

LEARNING ASSESSMENT TOOL EXTENSIONS

The Learning Assessment Tools measure the efficacy of the experiences by analyzing the data recorded throughout the learners' participation. Traditionally, learning assessment has been done through examinations and experts' reviews and opinions on students' work. However, most approaches emphasize comparing learners' performance against those of the experts' and less about evaluating the actual learning performance of individuals. There has been much research in the domain of systems engineering education attempting to find the best way to assess students' understanding and learning about systems engineering (see the Reference section of this report). Though simulation has been widely adapted by systems engineering learning, it has yet to be used to assess learner competencies and learning performance in systems engineering and technical leadership.

Learning assessment is a critical component of accelerated learning. It is imperative to understand individual learning and the efficacy of the various learning experiences. This is critical both in determining the capabilities of the learner, but also enable the continual improvement of the capabilities of the learning experience.

PRIOR WORK

In previous phases, research was done to create the high-level design of the Learning Assessor. A Learning Assessor prototype was then developed and demonstrated. The current effort involves the continued development of the Learning Assessor tool, assessing its capabilities through external evaluation, and determining desired features for future development.

Learning Assessment Tools – This tool analyzes the subject's activities, decisions, project performance and self-assessments to determine the subject's competency and learning level achieved. This work involves the development of the logging ability to collect and record the subject's inputs, and a tool to analyze the results. The SEEA infrastructure has been updated to perform the data logging and collection task. The tool is capable of importing the recorded learner data and performing visualization tasks to help experience designers and instructors understand the learners' performance.

Implemented functionalities are (see Figures 9 and 10):

- Collect and format experience data from EA and process data
- Visualize the data gathered from users' experience
- Display users' decision-making data
- Compare one experience against another (historic experience or experts' experience)
- Compare students' data within a class against each other in stack chart

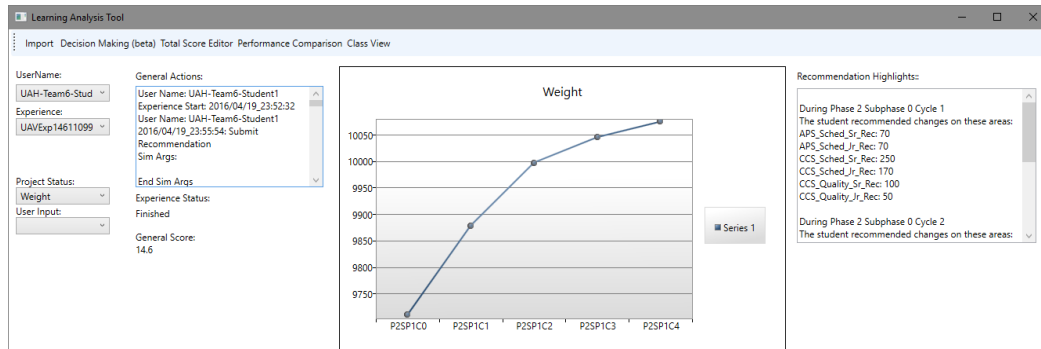


Figure 9. Main View of the Learning Assessment Tool

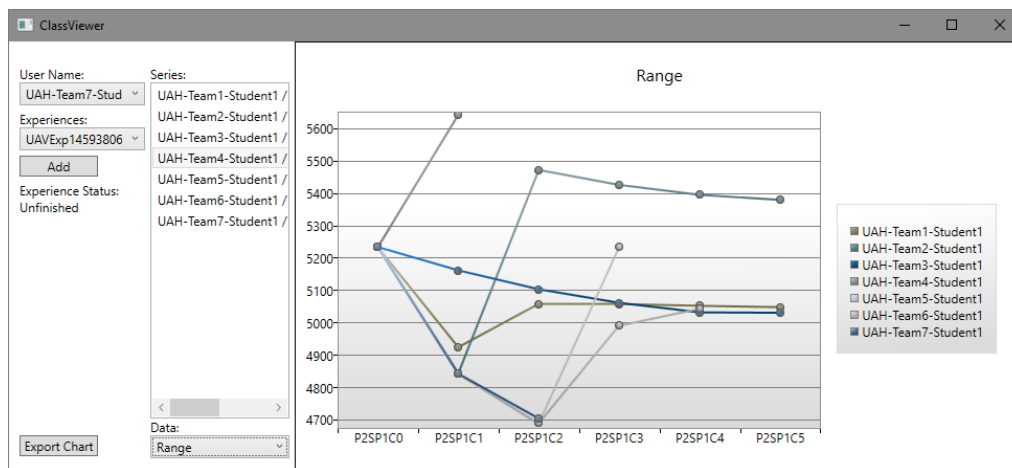


Figure 10. Learning Assessment Tool Class View

New functionalities for the learning assessment tool focus on aiding the instructors and teachers when he or she is analyzing the students' performance and decision-making data.

FUNCTIONALITY UPDATES

A new python-based behavioral analysis tool has been created for analyzing learner behavior and performance. The tool automatically tracks the learner's behavior, recommendation inputs and

recommendation rationale, and digests the information into a readable text form for instructors to easily gain insight into the learner's decision-making process. As an output, the tool generates a report of the summary of the learner's recommendation changes together with recommendation rationales, in the order of phases and cycles. The report also demonstrates how the system was reacting to the learner's input by showing related project status changes. Figure 11 illustrates the behavioral analysis tool and its output.

To better analyze learner's behavioral data and decision-making process, four Experience Accelerator Behavioral Archetypes were identified. Each archetype has multiple classifiers that can be fed into a machine learning model to determine which archetype best fits the subject and at what confidence level.

- Investigators - Make no major changes early on, conduct thorough investigation, make major changes late.
- Observers - Make no major changes early on, did not thoroughly investigate, only observe the trend and make major changes late.
- Responders - Make major changes throughout, did not thoroughly investigate, changes respond to situations.
- Researchers - Make major changes early on, conduct thorough investigation, observe the trend.

The archetypes can be used to better analyze the learner's overall performance. For each action/challenge, there are related background thinking and decision-making processes. These processes would be better analyzed utilizing the archetypes.

The machine learning model is currently being fine-tuned using performance data from North Carolina A&T State University (NCAT) students and will be updated to be incorporated into the learning assessment tool for future analysis.

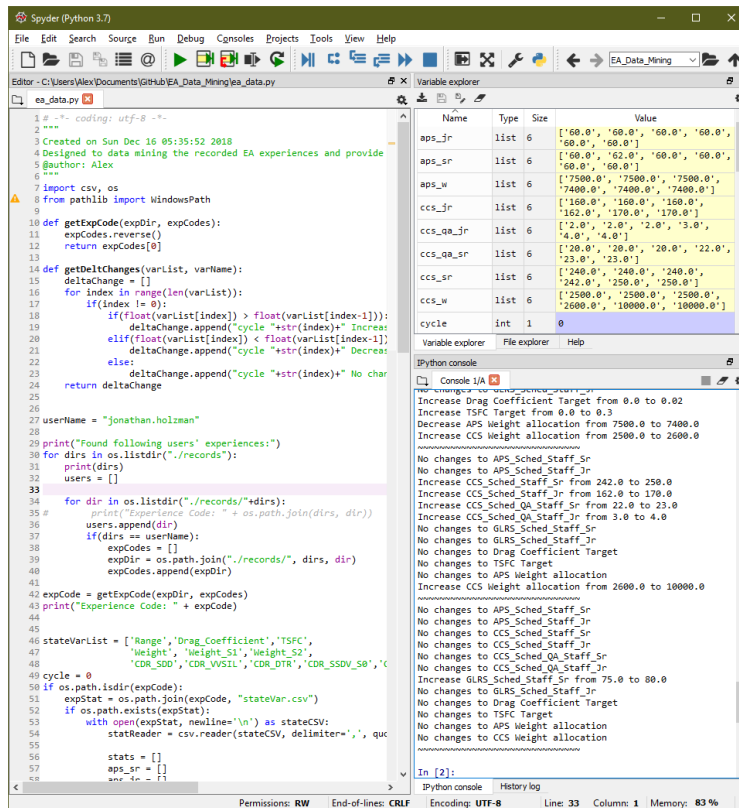


Figure 11. Behavioral Analysis Tool

The Learning Assessor's assessment engine has been developed for generating an objective score representing the learner's performance. It utilizes the experience comparison feature from the Learning Assessor to determine the learner's performance. The engine was developed using C# in console mode thus allowing it to be used by SEEA and third-party applications by piping through its inputs and outputs interfaces.

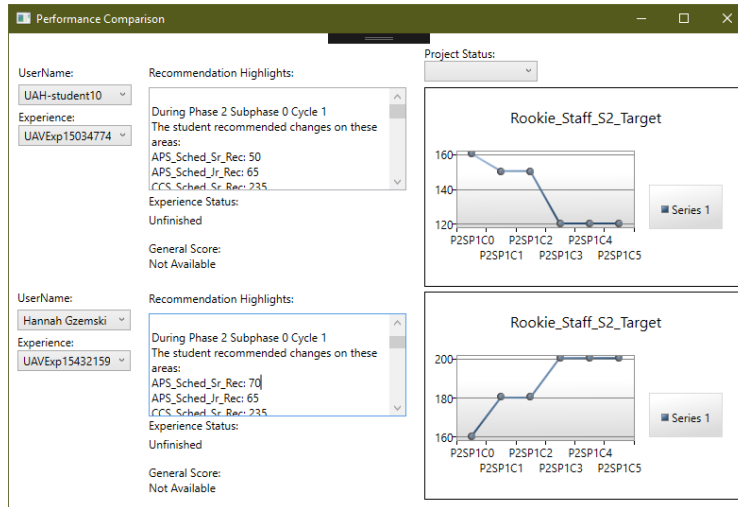


Figure 12. Assessment Engine Performance Comparison

UNIT TESTING AND INTEGRATION TESTING

During the development of the Learning Assessment tools, unit testing and integration testing were performed. Unit testing was performed when a new feature was added to ensure that it performed as intended. Integration testing was performed, as well, to ensure that changes did not adversely affect existing functionality in the tools. During this process, a variety of defects were identified and remediated.

INFRASTRUCTURE UPGRADE

During the Part 4 project development, the EA infrastructure was updated to support the previously described new features for the system. The prior work and new work are summarized below.

PRIOR WORK

Prior work involved updating the infrastructure, including both the server and client end, to support the use of the tools. The updates include the support to use experience building tools which required updates in the server-side code to allow external configurations in the learning experiences. The infrastructure also provides some of the required updates to allow data recording for the use of the learning assessment tool.

The following are the functions that were completed in the Experience Accelerator infrastructure to support the EA tools and HTML5:

- **Update of network protocol from TCP socket to Websocket** - The prior Java TCP socket protocol is not compatible with HTML5 and Javascript. For the HTML5 version to work, a

new Websocket protocol based on the HTTP server was developed. The updated server supports the Websocket protocol and thus supports the HTML5 version of the EA.

- **Move certain client functionalities to server side** - The server infrastructure was updated to include multiple client functionalities to make the HTML5 version possible.
- **Conversion of current artifacts** - The current artifacts were converted to support the HTML5 version of EA. This involves the conversion of documents from SWF format to the PDF format. Also, some form files were recreated in HTML5 format.
- **Rebuild dialog engine for HTML5** - The dialog engine has been rebuilt and updated for supporting the HTML5 version of the EA. The server dialog parser also was updated to support JSON format of dialog files.

NEW FUNCTIONALITY

The following are the functions that were completed in the Experience Accelerator infrastructure during this research period to support new features and new tools:

- Added new functionality to dialog engine, allowing variable assigning functions from dialogue scripts. With new and improved dialog UI, the user no longer needs to click to advance conversation.
- Reflection report has been improved to provide more information to the learner with an updated score generation function.
- Experience Server has been updated to allow collection of more detailed data about learners' performance and behaviors.
- Experience Server has been upgraded to allow upload of local Excel files, enabling the collection of learner's local trade study matrix worksheet.

UNIT TESTING AND INTEGRATION TESTING

During the development of the infrastructure upgrade, unit testing and integration testing were performed. Unit testing was performed when a new feature was added to ensure that it performed as intended. Integration testing was performed, as well, to ensure that changes did not adversely affect existing functionality in the tools. During this process, a variety of defects were identified and remediated.

CONCLUSION

This report has presented results from the fourth part of a project to create tools to support design, development and measurement of learning of experiences in the Experience Accelerator. Each Part has built on the successful completion of its previous part. The tools development efforts fall into four major categories:

- simulation tools for building and testing simulation models that mimic the behavior and results of acquisition programs that focus on system design and development,
- experience building tools that provide the structure for such system engineering experiences and the events that occur in them,
- learning assessment tools to measure the efficacy of the experience
- EA infrastructure changes to support the tools and evolution of the EA (HTML5 upgrade).

Table 3 shows the work plan for the three parts of the effort.

Table 3. SEEA tool development plan

Time	Simulation Tools	Experience Builder Tools	Learning Analysis Tools	EA Infrastructure
Part 1	Develop prototype Sim Builder and Tuner tools	Develop prototype Phase and Event Editor, and/or Artifact Integrator tools	Research and create high level design of Learning Assessor tool	Update EA infrastructure to support tools
Part 2	Extended functionality of Sim Builder and Tuner tools, make available to broader community. Determine new functionality as needed	Refine Phase and Event Editor, and/or Artifact Integrator tools, make available to broader community. Open source. Determine new functionality as needed	Develop prototype Learning Assessor prototype	Update EA infrastructure to support tools
Part 3	Extend functionality of Sim Builder and Tuner tools, make available open source. Determine new functionality as needed. Support existing tools, Specify new tools	Extend functionality of tools, make available open source. Support existing tools. Specify new related tools	Develop Learning Assessor tool, make available to for external evaluation. Specify future functionality	Update EA infrastructure to support tools
Part 4	Specify generic EA-simulator interface for different simulations and provide state-chart simulation support	Support existing tools.	Extend functionality of Learning Assessor tool. Support existing tools.	Update EA infrastructure to support tools.

The risk and mitigation plan worked well in supporting the successful completion of all the objectives of this task that did not have dependencies on other work. In particular, the focus on agile development was important in allowing desired and high-priority functionalities to be developed.

- **Risk:** This task was not funded until March 15, 2018, even though the period of performance started on February 13, 2018. This increases the risk that the work will be completed by this end of work date.
- **Mitigation strategy:** Modify the contract to have a period of performance that is consistent with the funding start date, perhaps through a no-cost extension. If contract modification is not possible, compress schedule as much as can be accommodated.
- **Risk:** The likelihood of feature creep that prevents the completion of “vignette” experiences and tools to provide a complete environment for Experience creation.
- **Mitigation strategy:** Continue to utilize agile development practices to ensure that the highest value features are being developed at all times. The number and capabilities of the “vignette” experiences can be adjusted based on available time and resources.
- **Risk:** Ensuring the availability of graduate student staff with the knowledge and capabilities required for effective, efficient tool development despite gaps in funding.
- **Mitigation strategy:** Keep current graduate students engaged as much as possible with stop gap funding from other sources. Determine other sources of students or software research developers.

The maturity of the Experience Accelerator tools has been tested in the development of vignette experiences. Few new features were identified as being necessary for this work. In addition, the newly developed simulator interface provides the opportunity to support a wide-range of new experiences that do not utilize a systems dynamics simulation approach. Finally, the new learning assessment capabilities will be used in one of the author’s ongoing doctoral research.

There continues to be interest in the use of the Experience Accelerator in education. On the academic side, at the time of this report, the EA is being used at Air Force Institute of Technology, Georgia Tech, Stevens Institute of Technology, and University of Alabama in Huntsville. The EA is also being used by the United Kingdom Ministry of Defence. Finally, the EA is now being used in the industrial sector by a major aerospace corporation.

Much of the future work now consists releasing the EA and its tools in open source forms and the development of the open source community and ecosystem. There is significant interest in government, industry and academia to support these efforts. It is time to expand the use of the EA beyond the SERC and its collaborating partners.

APPENDIX A: LIST OF RESEARCH RELATED PUBLICATIONS

Bodner, D., Wade, J., Squires, A., Reilly, R., Dominick, P., Kamberov, G., & Watson, W. (2012). Simulation-based decision support for systems engineering experience acceleration. In *Proc. 2012 IEEE Systems Conference*, Vancouver, BC, Canada.

Bodner, D., Wade, J., Watson, B., Kamberov, G. (2013). Designing an experiential learning environment for systems engineering and logistics. In *Proc. 2013 Conference on Systems Engineering Research (CSER)*, Atlanta, GA.

Hao, K. (2015). Design and development tools for the experience accelerator. Thesis, Department of Computer Science, Stevens Institute of Technology, Hoboken, NJ.

Squires, A., Wade, J., Dominick, P., Gelosh, D. (2011a). Building a competency taxonomy to guide experience acceleration of lead program systems engineers. In *Proc. 2011 Conference on Systems Engineering Research (CSER)*, Redondo Beach, CA.

Squires, A., Wade, J., Watson, B., Bodner, D., Okutsu, M., Ingold, D., Reilly, R., Dominick, P., Gelosh, D. (2011b). Investigating an innovative approach for developing systems engineering curriculum: The Systems Engineering Experience Accelerator. In *Proc. 2011 American Society for Engineering Education (ASEE) Annual Conference and Exposition*, Vancouver, BC.

Squires, A., Wade, J., Watson, W., Bodner, D., Reilly, R., Dominick, P. (2012). Year one of the Systems Engineering Experience Accelerator. In *Proc. 2012 Conference on Systems Engineering Research (CSER)*, Rolla, MO.

Turner, R., Bodner, D., Kemp, D., Rodriguez, Y., Wade, J., Zhang, P. (2017). SE Simulation Experience Design: Infrastructure, Process and Application. In *Proc. 27th Annual INCOSE International Symposium*, Adelaide, Australia.

Wade, J. P., Bodner, D. A., Watson, W., Turner, R., Hinkel, J., & Zhang, P. (2015). RT 123: Design and Development Tools for the Systems Engineering Experience Accelerator – Part 1. Technical Report No. SERC 2015-TR-016-4, Systems Engineering Research Center, Hoboken, NJ.

Wade, J. P., Bodner, D. A., Turner, R., Hinkel, J., & Zhang, P. (2016). RT 146: Design and Development Tools for the Systems Engineering Experience Accelerator – Part 2. Technical Report No. SERC 2016-TR-108, Systems Engineering Research Center, Hoboken, NJ.

Wade, J. P., Bodner, D. A., Rodriguez, A., Liu, J., Turner, R., & Zhang, P. (2017). RT 164: Design and Development Tools for the Systems Engineering Experience Accelerator – Part 3. Technical Report No. SERC 2017-TR-107, Systems Engineering Research Center, Hoboken, NJ.

Wade, J., Kamberov, G., Bodner, D., Squires, A. (2012). The architecture of the Systems Engineering Experience Accelerator. In *Proc. International Council on Systems Engineering (INCOSE) 2012 International Symposium/European Conference on Systems Engineering (EUSEC)*, Rome, Italy.

Zhang, P., Bodner, D. A., Turner, R. G., Arnold, R. D., Wade, J. P. (2016). The Experience Accelerator: Tools for development and learning assessment. In *Proc. of the 2016 American Society for Engineering Education (ASEE) Annual Conference and Exposition*. Washington, DC: ASEE.

Zhang, P., Wade, J., Turner, R., Bodner, D., Thomas, D. (2017). SEEA: Accelerated Learning and Learning Assessment for Systems Engineering Education. In *Proc. 2017 Conference on Systems Engineering Research*, Los Angeles, CA.

APPENDIX B: CITED AND RELATED REFERENCES

Journal and Conference Articles and Reports

- Ardis, M., McGrath, E., Lowes, S., & Lam, S. (2012). Research on building education & workforce capacity in systems engineering. Interim Technical Report SERC-2012-TR-019-1, Systems Engineering Research Center, Stevens Institute of Technology, Hoboken, NJ.
- Armstrong, J., & Wade, J. (2015). Learning systems engineering by teaching it. In *Proc. 2015 INCOSE International Symposium*, 211-220.
- Arnold, R., & Wade, J. (2015). A definition of systems thinking: A systems approach. *Procedia Computer Science*, 44, 669-678.
- Baker, A., Navarro, E. O., & van der Hoek, A. (2005). An experimental card game for teaching software engineering processes. *Journal of Systems and Software*, 75(1-2), 3-16.
- Belland, B. R., French, B. F., & Ertmer, P. A. (2009). Validity and problem-based learning research: A review of instruments used to assess intended learning outcomes. *Interdisciplinary Journal of Problem-Based Learning*, 3(1). Available at: <http://dx.doi.org/10.7771/1541-5015.1059>.
- Brown, J. S., Collins, A., & Duguid, P. (1989). Situated cognition and the culture of learning. *Educational Researcher*, 18(1), 32-42.
- Burandt, S. (2011). Effects of an educational scenario exercise on participants' competencies of systemic thinking. *Journal of Social Sciences*, 7, 54-65.
- Chapman, G. M., Martin, J. F. (1995). Computerized business games in engineering education. *Computers & Education*, 25(1-2), 67-73.
- Chen, W.-F., Wang, T.-L., Su, C.-H., & Wu, W.-H. (2008). Work in progress - A game-based learning system for software engineering education. In *Proc. 2008 IEEE Frontiers in Education Conference*.
- Ebner, M., & Holzinger, A. (2007). Successful implementation of user-centered game based learning in higher education: An example from civil engineering. *Computers & Education*, 49(3), 873-890.
- Futter, M. (2014). World of Warcraft subscriptions slide by 800,000. Game Informer. GameStop. Retrieved August 5, 2014.
- Harel, D. (1987). Statecharts: A visual formalism for complex systems. *Science of Computer Programming*, 8(3): 231-274.
- Hopper, M. & Stave, K. A. (2008). Assessing the effectiveness of systems thinking interventions in the classroom. In *Proc. 26th International Conference of the System Dynamics Society*.

Jain, A., & Boehm, B. (2006). SimVBSE: Developing a game for value-based software engineering. In *Proc. 2006 Conference on Software Engineering Education & Training*.

Klieme, E., & Maichle, U. (1991). Erprobung eines Modellbildungssystems im Unterricht. Bericht über eine Pilotstudie zur Unterrichtsevaluation. Bonn: Institut für Test- und Begabungsforschung.

Kopainsky, B., Alessi, S. M., Pedercini, M., & Davidsen, P. I. (2009). Exploratory strategies for simulation-based learning about national development. In *Proc. 27th International Conference of the System Dynamics Society*.

Kopainsky, B., Pirnay-Dummer, P., & Alessi, S.M. (2012). Automated assessment of learners' understanding in complex dynamic systems. *System Dynamics Review*, 28(2), 131-156.

Kopainsky, B., & Saldarriaga, M. (2012). Assessing understanding and learning about dynamic systems. In *Proc. 30th International Conference of the System Dynamics Society*.

Kopainsky, B., & Sawicka, A. (2011). Simulator-supported descriptions of complex dynamic problems: Experimental results on task performance and system understanding. *System Dynamics Review*, 27, 142-172.

Morsi, R., & Jackson, E. (2007). Playing and learning? Educational gaming for engineering education. In *Proc. 2007 IEEE Frontiers in Education Conference*.

Okutsu, M. (2009). Teaching engineering design principles via a serious game format. Report, Purdue University, West Lafayette, IN.

Ossimitz, G. (2000). Teaching systems dynamics and systems thinking in Austria and Germany. *Proc. 18th International Conference of the System Dynamics Society*.

Pirnay-Dummer, P., Ifenthaler, D., & Spector, J. M. (2010). Highly integrated model assessment technology and tools. *Educational Technology Research and Development*, 58(1), 3-18.

Plate, R. (2010). Assessing individuals' understanding of nonlinear causal structures in complex systems. *System Dynamics Review*, 26, 19-33.

Rimiene, V. (2002). Assessing and developing students' critical thinking. *Psychology Learning & Teaching*, 2, 17-22.

Rouwette E.A.J.A., Größler A., Vennix J.A.M. (2004). Exploring influencing factors on rationality: a literature review of dynamic decision-making studies in system dynamics. *Systems Research and Behavioral Science*, 21(4), 351–370.

Skaza, H., & Stave, K. (2009). A test of the relative effectiveness of using systems simulations to increase student understanding of environmental issues. In *Proc. 27th International Conference of the System Dynamics Society*.

Skaza, H., & Stave, K. A. (2010). Assessing the effect of systems simulations on systems understanding in undergraduate environmental science courses. In *Proc. 28th International Conference of the System Dynamics Society*.

Spector, J. M. (2006). A methodology for assessing learning in complex and ill-structured task domains. *Innovations in Education and Teaching International*, 43(2), 109-120.

Spector, J. M., Christensen, D. L., Sioutine, A. V., & McCormack, D. (2001). Models and simulations for learning in complex domains: Using causal loop diagrams for assessment and evaluation. *Computers in Human Behavior*, 17(5), 517-545.

Stave, K. A. (2011). Using simulations for discovery learning about environmental accumulations. In *Proc. 29th International Conference of the System Dynamics Society*.

Stave, K. A., Beck, A., & Galvan, C. (2014). Improving learners' understanding of environmental accumulations through simulation. *Simulation & Gaming*, 46(3-4), 270-292.

Steif, P. S. H. M. (2006). Comparisons between performances in a statics concept Inventory and course examinations. *International Journal of Engineering Education*, 22(5), 1070-1076.

Sterman, J. D. (2010). Does formal system dynamics training improve people's understanding of accumulation? *System Dynamics Review*, 26, 316-334.

Stewart, M. (2012). Joined up thinking? Evaluating the use of concept-mapping to develop complex system learning. *Assessment & Evaluation in Higher Education*, 37(3), 349-368.

Taran, G. (2007). Using games in software engineering education to teach risk management. In *Proc. 2007 Conference on Software Engineering Education & Training*.

Walbridge, M. (2008). Analysis: Defense of the Ancients – An Underground Revolution. <http://www.gamasutra.com/>, retrieved June 23, 2008.

Weinerth, K., Koenig, V., Brunner, M., & Martin, R. (2014). Concept maps: A useful and usable tool for computer-based knowledge assessment? A literature review with a focus on usability. *Computers & Education*, 78, 201-209.

Books and Book Chapters

Cohen, S. (2006). *Virtual Decisions: Digital Simulations for Teaching Reasoning in the Social Sciences and Humanities*. Mahwah, NJ: L. Erlbaum Associates.

Driscoll, M. P. (2005). *Psychology of Learning for Instruction* (3rd Ed.). Boston: Pearson Education, Inc.

Dörner, D., & Wearing, A. (1995). Complex problem solving: Toward a (computersimulated) theory. In *Problem Solving: The European Perspective*, Frensch, P. A., & Fubke, J. (eds.), Psychology Press, 65-102.

Forrester, J. W. (1961). *Industrial Dynamics*. Pegasus Communications.

Gibson, D., Aldrich, C., & Prensky, M. (eds.) (2007). *Games and Simulations in Online Learning: Research and Development Frameworks*. Hershey, PA: Information Science Pub.

Herrington, J., & Oliver, R. (1995). Critical characteristics of situated learning: Implications for the instructional design of multimedia. In *Learning with Technology*, Pearce, J., & Ellis, A. (Eds.). Parkville, Vic: University of Melbourne, 235-262.

Herz, J. C., & Macedonia, M. R. (2002). Computer games and the military: Two views. *Defense Horizons*, 11.

Jones, K. (1997). *Games and Simulations Made Easy: Practical Tips to Improve Learning through Gaming*. London: Kogan Page.

Karakul, M., & Qudrat-Ullah, H. (2008). How to improve dynamic decision making? Practice and promise. In *Complex Decision Making: Theory and Practice*, Qudrat-Ullah, H., Spector J.M., & Davidsen, P.I. (eds). Springer/NECSI: Berlin; 3–24.

Kolb, D. (1984). *Experiential Learning*. Prentice Hall, Inc.

Madachy, R. (2008). *Software Process Dynamics*, Wiley.

Novak, J. D., & Gowin, D. B. (1984) *Learning How to Learn*. Cambridge University Press, Cambridge.

O'Neil, H. F., & Perez, R. S. (2008). *Computer Games and Team and Individual Learning*. Amsterdam: Elsevier.

Prensky, M. (2001). *Digital Game-based Learning*. New York: McGraw-Hill.

Shaffer, D. W. (2006). *How Computer Games Help Children Learn*. New York: Palgrave Macmillan.

Sterman, J. D. (2000). *Business Dynamics: Systems thinking and modeling for a complex world*. McGraw Hill.

Van Ments, M. (1999). *The Effective Use of Role-Play: Practical Techniques for Improving Learning*. London: Kogan Page.

Theses

Kefalas, N. (1998). Project Management Utilizing System Dynamics and Design Structure Matrices in Conjunction with the Earned Value System, M.S. Thesis, Massachusetts Institute of Technology.

Web Resources

Apache, 2019. Apache Commons SCXML, <http://commons.apache.org/proper/commons-scxml/index.html>, retrieved 5/7/2019.

W3C, 2019. State Chart XML (SCXML): State Machine Notation for Control Abstraction, <https://www.w3.org/TR/scxml/>, retrieved 5/7/2019.