
AC 2011-1295: INVESTIGATING AN INNOVATIVE APPROACH FOR DEVELOPING SYSTEMS ENGINEERING CURRICULUM: THE SYSTEMS ENGINEERING EXPERIENCE ACCELERATOR

Alice F Squires, Stevens Institute of Technology

Alice Squires has nearly 30 years of professional experience and is an industry and research professor in Systems Engineering at Stevens Institute of Technology in the School of Systems and Enterprises. She is a Primary Researcher for the Body of Knowledge and Curriculum to Advance Systems Engineering (BKCASE) and Systems Engineering Experience Accelerator projects. She has served as a Senior Systems Engineer consultant to Lockheed Martin, IBM, and EDO Ceramics, for Advanced Systems Supportability Engineering Technology and Tools (ASSETT), Inc. Alice previously served as a senior engineering manager for General Dynamics (GD), Lockheed Martin (LM) and as a technical lead for IBM. Alice is a lifetime member of Beta Gamma Sigma (Business), Tau Beta Pi (National Engineering), and Eta Kappa Nu (National Electrical Engineering) Honorary Societies and is an International Council on Systems Engineering (INCOSE) Certified Systems Engineering Professional (CSEP) in both base and Acquisition (CSEP-Acq). She is in the process of completing her doctorate dissertation in "Investigating the Relationship Between Online Pedagogy and Student Perceived Learning of Systems Engineering Competencies" and her research interests include systems engineering competency development, systems thinking and systems engineering education. Alice is the Chair of the Systems Engineering Division of ASEE and has a Masters in Business Administration (MBA) and Bachelors of Science in Electrical Engineering (BSEE). Alice received the Stevens Institute of Technology Provost's Online Teaching Excellence Award in 2007.

Jon Wade, Ph.D., Stevens Institute of Technology

Jon Wade, Ph.D. is the Associate Dean of Research at the School of Systems and Enterprises at the Stevens Institute of Technology. Dr. Wade's research interests include the transformation of systems engineering, Enterprise Systems and Systems of Systems, and the use of technology in technical workforce development. He has over 20 years of experience in the research and development of Enterprise systems at IGT, Sun Microsystems and Thinking Machines Corporation. Dr. Wade is a graduate of the Massachusetts Institute of Technology.

Douglas A. Bodner, Georgia Institute of Technology

Douglas A. Bodner is a senior research engineer in the Tennenbaum Institute at the Georgia Institute of Technology. His research focuses on computational analysis and decision support for design, operation and transformation of enterprise systems. His work has spanned a number of industries, including aerospace and defense, automotive, electronics, energy, health care, paper and pulp, semiconductors and telecommunications. Dr. Bodner is a senior member of the Institute of Electrical and Electronics Engineers (IEEE) and the Institute of Industrial Engineers (IIE), and a member of the American Society for Engineering Education (ASEE) and the Institute for Operations Research and Management Science (INFORMS). He is a registered professional engineer in the State of Georgia.

Masataka Okutsu, Purdue University

Masataka Okutsu is a Postdoctoral Researcher in the School of Aeronautics and Astronautics at Purdue University. Dr. Okutsu directed development of virtual-world software for an aerospace design course (which he co-taught). With research background in astrodynamics, Dr. Okutsu has designed spacecraft trajectories for human missions to Mars and robotic missions to the outer planets.

Dan Ingold, University of Southern California

Mr. Ingold is a Senior Research Analyst and PhD student with the Center for Systems and Software Engineering at the University of Southern California (USC). Mr. Ingold has over 30 years of experience in the development of software-intensive systems, and prior to joining USC was CEO of a firm that developed specialized systems for defense C4ISR and industrial applications. His research interests are in the application of hybrid agile/plan-driven techniques to the development of large-scale, software-intensive systems. Mr. Ingold received his BS in Computer Science from Purdue University, and MS in Computer Science from USC.

Peter G. Dominick, Ph.D., W.j. Howe School of Technology Management, Stevens Institute of Technology

Peter G. Dominick, Industry Assistant Professor teaches leadership development courses within the W.J. Howe School of Technology Management at Stevens Institute of Technology. His research focuses on leadership development processes with a particular focus on social cognitive dimensions of behavior change. Pete received his Ph.D. in Applied Psychology from Stevens, earned his MA in Organizational Psychology from Columbia University, and completed his undergraduate studies in Industrial and Labor Relations at Cornell University. He has received the Howe School's Outstanding Teacher Award and also the Institute's Harvey N. Davis Award for Distinguished Teaching.

Richard R. Reilly, Stevens Institute of Technology

Richard R. Reilly holds the Ph.D. in Organizational Psychology from the University of Tennessee and is an Emeritus Professor in the Howe School of Technology Management, Stevens Institute of Technology. Dr. Reilly joined the Stevens faculty in 1982 where he developed and led the Ph.D. program in Technology Management. Before joining Stevens, Dr. Reilly was a research psychologist for Bell Laboratories, the Educational Testing Service and AT&T and has been a consultant to Fortune 500 and governmental organizations. He is on the Editorial Board of Personnel Psychology, and the International Journal of e-Collaboration and is a Fellow of the American Psychological Association and the American Psychological Society. He has authored four books and over 70 publications related to organizational behavior and project and team performance. Dr. Reilly's most recent books include *Blockbusters: Five Keys to Developing Great New Products* published by Harper-Collins and *Uniting the Virtual Workforce*, published by Wiley & Sons. Dr. Reilly has developed and taught courses in Research Methods, Multivariate Statistics, Leadership, and Judgment and Decision Making. He currently serves as a technical advisor to the Office of Naval Research Benchmarking and Best Practices Center in Manufacturing.

William R. Watson, Purdue University

William Watson is an Assistant Professor of Educational Technology in the Department of Curriculum and Instruction at Purdue University and the director of the Purdue Center for Serious Games and Learning in Virtual Environments. His research interests include the critical, systemic change of education to focus on learner-centered learning environments, including customized and personalized learning through the application of technology such as video games, virtual environments, and learning management software.

Don Gelosh, ODDRE/Systems Engineering

Dr. Don Gelosh is the Deputy Director for Workforce Development in the OSD Directorate of Systems Engineering. He provides expertise in workforce development, competency models and assessments, and knowledge management with over 34 years of systems engineering experience from the US Air Force, government, industry, and academia. While serving in the Air Force, Dr. Gelosh worked as a systems engineer on the Space Shuttle as a member of NASA's Vehicle Integration and Test Team where he was responsible for communications and payload integration and ensuring the Shuttle was ready for launch. Dr. Gelosh also taught electrical and computer engineering at the Air Force Academy in the early 90's and later served as Deputy Department Head for Electrical and Computer Engineering at the Air Force Institute of Technology. Don received his PhD in Electrical Engineering from the University of Pittsburgh in 1994, a MS in Computer System Design from the University of Houston at Clear Lake in 1989, and a BS in Electrical Engineering from the Ohio State University in 1981. He also holds an INCOSE CSEP-Acquisition certification and is DAWIA Level III certified in SPRDE Systems Engineering.

Investigating an Innovative Approach for Developing Systems Engineering Curriculum: The Systems Engineering Experience Accelerator

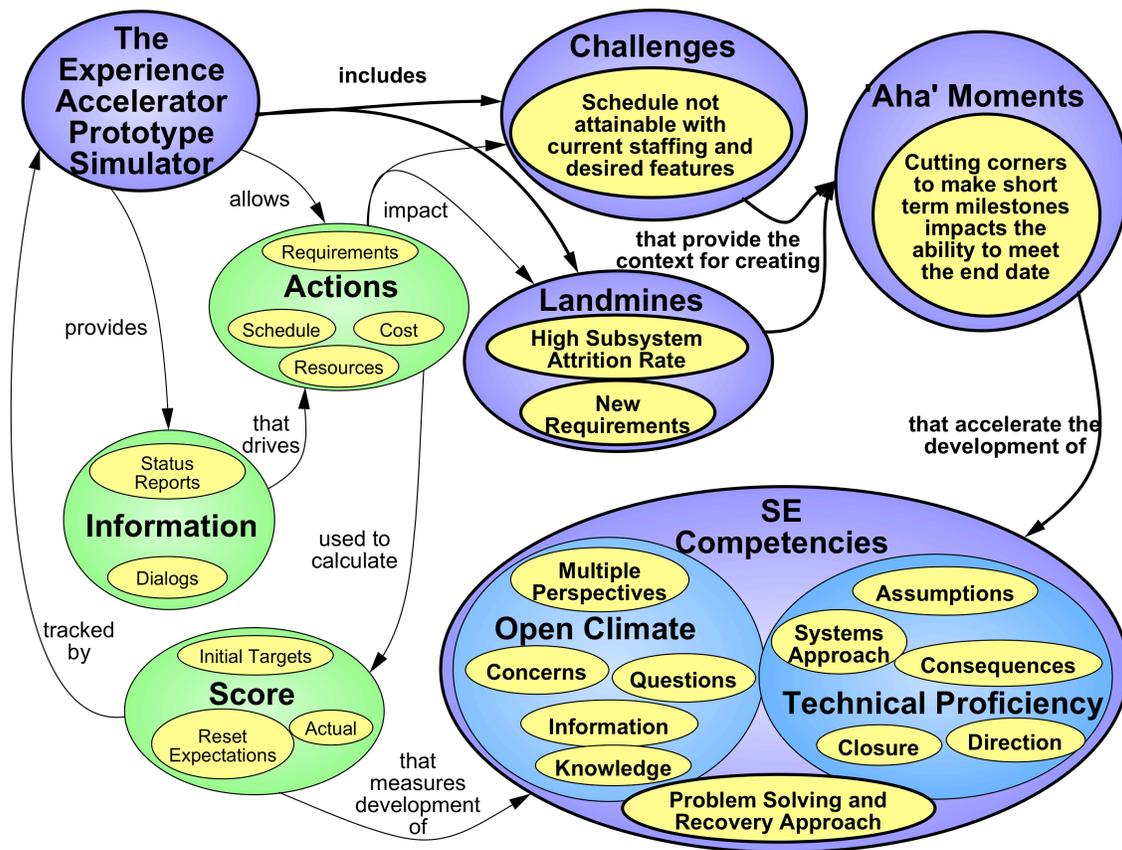
Abstract

The systems engineering Experience Accelerator (ExpAcc) is a research project in the early stages of definition and development that is focused on validating the feasibility of leveraging simulation technology to create a series of experiences that will accelerate the maturity of systems engineers. This paper leverages the approach being defined for the research project as the basis for a set of recommendations for developing systems engineering curriculum for the live classroom. The focus of the research project is to create a computer-based simulator prototype that provides an integrated, experience based learning environment intended to accelerate the learning of critical systems engineering competencies. However, the goal of providing realistic, emotionally engaging simulated experiences customized to the learner's specific needs can be applied not only to a computer simulation, but also to a live classroom simulation. In both cases, customization can be based initially on the learner's self reported assessment of their competencies and learning preferences, and subsequently on the learner's progression through the simulation. This paper introduces the research questions and research hypothesis being investigated on the research project; describes the approaches being used to create the simulator prototype that is still under development (not yet available for testing); and includes recommendations for developing systems engineering related experiences in the classroom that could potentially accelerate the student's learning of selected systems engineering competencies.

1.0 Introduction

Systems engineering educators are struggling to address workforce development needs required to meet the emerging challenges posed by increasing systems complexity¹ and the widening gap in systems engineering expertise in the workforce.² The systems engineering Experience Accelerator (ExpAcc) research project was conceived as a critical response to these needs and challenges. The project was initiated to validate the use of technology to potentially create an experiential, emotional state in the learner coupled with reflective learning so that time is effectively compressed and the learning process of a systems engineer (SE) is significantly accelerated as compared to the rate at which learning would occur naturally on the job. The purpose of the research project is to test the feasibility of a simulated approach for accelerating systems engineering competency development in the learner.

A notional diagram of how the various concepts developed for the ExpAcc are related is shown in Figure 1. The development team has a threefold challenge to balance the development of the simulator technology (ExpAcc) that supports displayed content (shown in green) that, in turn, supports the developed concepts (shown in purple). The goal is to effectively create challenges and landmines that support the user's experience of the necessary "Aha" moment. The intent is that by experiencing the "Aha" moment, the user transitions to a more advanced level of understanding in the targeted competency, in this case "Problem Solving and Recovery Approach".



Alice Squires 1/31/2011

Figure 1. Notional Diagram of the ExpAcc Prototype Simulator

The testing of the prototype (future) will support evaluation of the theoretical capabilities of the developed system and provide guidance for the continuing development of the ExpAcc simulator going forward. The initial users of the simulator will be members of the acquisition workforce in training at the Defense Acquisition University (DAU). The simulator is also planned as a component of graduate level technical leadership programs and executive level training

2.0 Background and Overall Motivation

In *The Art and Science of Systems Engineering*, Mr. Harold Bell, Director Advanced Planning and Analysis Division, NASA Office of Chief Engineer, is quoted as saying: “A great systems engineer completely understands and applies the art of leadership and has the experience and scar tissue from trying to earn the badge of leader from his or her team.”³ (p. 2) Historically, competent systems engineers have developed their “scar tissue” by gaining the necessary insights and wisdom through both failures and successes, in an integrated real world environment. In the workplace, however, the learning events that result in the development of “scar tissue” are distributed, sometimes sparsely, over time. In addition, a common benchmark time for the development of a competent SE is a minimum of about 10-15 years.⁴ Given there is a shortfall of SEs in the global workforce today⁵ and no readily available source of SEs to replace the top SEs

in the retiring baby boomer generation, the time to develop competent SEs needs to be significantly shortened.

The primary goal of the ExpAcc, once it is developed, is to accelerate the maturation of SEs in the workforce by providing the opportunities to earn “scar tissue” through realistic, engaging simulation. These tailored experiences will allow the learner to feel the consequences of success and failure in a simulated environment so they can gain the necessary insights and wisdoms to mature as a SE, and yet not jeopardize the lives of others or compromise their careers. The initial target audience of the ExpAcc program is lead program SEs in the acquisition workforce who are required to effectively manage complex systems throughout their lifecycle from an acquisition/acquirer viewpoint in a typical program office. The initial focus is on maturing these leads to prepare them for executive assignments.

Computer Game Based Education

An increasing number of practitioners and researchers are advocating digital games (also referred to simply as video games) as a promising form of instruction which can both engage students and strengthen skills important in the current information age.⁶⁻¹⁰ The Federation of American Scientists has identified digital games as well-suited to educating for today’s knowledge economy, providing motivating instruction for the development of higher order thinking skills, expertise, teamwork, and problem solving.⁷ And yet, Gardner notes the overwhelming number of studies which show that top-rated college and high school physics students have repeatedly demonstrated an inability to apply their knowledge in new situations.¹¹ The higher order thinking and problem solving skills promoted by video game playing highlight the impact that games can have on their players and the strong benefit they can provide in transforming today’s educational methods to approaches more suitable for developing the modern workforce.

Effective educational games encourage learning in ways that fit well with established learning theories. This includes the role of feedback represented in behaviorism, the role of play and of cognitive disequilibrium, assimilation and accommodation discussed by Piaget, the role of story in narrative psychology, and the role of meaningful context in learning that informs situative cognition. A number of these earlier theories, namely situative cognition and Bruner and Vygotsky’s emphasis on the role of socialization and interaction in learning impacted the development of constructivism.¹² Constructivism takes the point of view that learners construct their own knowledge, rather than being recipients of pre-existing knowledge, as described by behaviorist approaches. In order to construct knowledge, learners interact with authentic and challenging problems. By working through such problems, learners become self-directed and capable of applying skills and knowledge in new contexts. Driscoll lists “problem solving, reasoning, critical thinking, and the active and reflective use of knowledge” (p. 393) as the goals of constructivist instruction and provides the following constructivist conditions for learning: (1) Embed learning in complex, realistic and relevant environments; (2) Provide for social negotiation as an integral part of learning; (3) Support multiple perspectives and the use of multiple modes of representation; (4) Encourage ownership in learning; and (5) Nurture self-awareness of the knowledge construction process.¹²

While situated learning originally called for the requirement of authentic tasks completed in

authentic, social and physical environments;¹³ Herrington and Oliver note that numerous researchers have since identified computers as a suitable alternative for producing an authentic context.¹⁴ In adapting situated learning to interactive multimedia, the authors stress the importance of not focusing solely on the multimedia but also on the individual learners as well as the implementation of the multimedia and their interactions with it. It is the inter-relation of these three components: the learner, the implementation, and the interactive multimedia which together meet the requirements of an efficient situated learning environment and which all must be taken into account. The interplay of these three components is equally relevant to learning environments using video games, a form of interactive multimedia.

Shaffer calls his educational games “epistemic games” because they situate the player in a particular role, as a soldier, or a theme park manager, or an astronaut, and in playing out that role, the player becomes familiar with a specific epistemology, a knowledge-base composed of particular actions, vocabulary, and ways of thinking.¹⁵ This echoes Gee’s discussion of how games situate knowledge and is a reflection of situated cognition.¹⁶ The role of meaningful context, authentic problems, appropriate feedback and challenge, the familiarization with a specific epistemology through role-playing, and the importance of social interaction all inform our approach to a methodology of educational game design that incorporates these learning theories and their descriptions of how learning occurs.

Systems Engineers: Born or Made?

Ericsson has historically focused on the question of expert performance and whether experts are born and not made. In 1994, he concluded:

For a long time the study of exceptional and expert performance has been considered outside the scope of general psychology because such performance has been attributed to innate characteristics possessed by outstanding individuals. A better explanation is that expert performance reflects extreme adaptations, accomplished through life-long effort, to demands in restricted, well defined domains. (p. 744)¹⁷

Similarly, good systems engineers may evolve from engineers that seek out and are given new and challenging opportunities early on in their career and consistently over time in the form of challenging projects that consistently lie beyond their previous level of competency. It doesn’t seem to matter so much whether these engineers are successful or fail but rather what they take away from the experience and whether they continue to pursue and be given these opportunities over time. However, if an engineer fails, they may not be given another challenging opportunity and their development stops. A similar concept is discussed in *The Expert Mind* where Ross refers to 'effortful study' - continually tackling challenges that lie just beyond one's competence - to build a grandmaster in chess.¹⁸ The challenge for us as educators is how to speed up that process through training -- how to ensure the challenges presented lie just beyond one's competence so that motivation is preserved as new challenges are tackled. It is not uncommon for instructors to focus on the more general level of experience when they teach a topic. Under this scenario, an “advanced” learner is dissatisfied; a learner with little or no background is overwhelmed. Adaptive learning systems have the ability to challenge a student just beyond

their current level of expertise - by measuring the student's baseline competence and then going from there. This is the strategy employed on the ExpAcc project.

ExpAcc Simulator Prototype User Interface

Our first-year simulator prototype allows users to experience scenarios that are representative of systems engineers on acquisition programs. The user interface is relatively simple, designed similar to a computer desktop screen. A user would progress the scenario forward as he or she converses with the non-player characters (NPCs) via “e-mail” and “video conference,” while accessing text-based or multi-media resources that are made available throughout the learning experiences.

One of the baseline requirements for the project is to support the ability for the ExpAcc to run on computers both at home and at work. However, typical users in many organizations may not have administrative privileges to install or download new software. Also, requiring software installation or downloads could be problematic in certain work environments due to potentially lengthy certification processes. Finally, the online software required must be compatible to the secure network of respective institutions. Based on these considerations, the ExpAcc prototype will run on an Adobe® Flash® Player, a web-browser “plug-in” that is already used widely, eliminates the need for downloading or installing special software, and runs on any operating system (i.e., Windows, Mac, or Linux). From the development point of view, Flash is also attractive because of its rich off-the-shelf features that are already available and hence help expedite the software development. Also, in the prototype design, no user profile data will be stored on the client machines, but rather all information will be stored securely on the server along with the simulation program and simulator content.

3.0 ExpAcc Research Hypothesis and Questions

The ExpAcc research project hypothesis is:

By using technology we can create a simulation that will put the learner in an experiential, emotional state and effectively compress time and greatly accelerate the learning of a systems engineer faster than would occur naturally on the job.

The following are an evolving set of broad research questions that this research will select a subset from for consideration on the project:

1. General Education:
 - a. How do you determine the appropriate systems engineering, systems thinking and technical leadership competencies and capabilities for use in an experience accelerator?
 - b. What are highly effective means of gaining expertise in complex systems and how is this expertise best provided?
 - c. What is the right balance between experiential learning, lecture based learning and independent study?

- d. How do you tune the learning experiences above to make them most appropriate for the participant's particular learning style?
2. ExpAcc Effectiveness:
 - a. Which is more effective under what circumstances – single-user or multi-user?
 - b. What is the right balance between exploratory learning, access to experts, and text-based help/lecture material?
 - c. How do we determine the participant's level of satisfaction and engagement?
 - d. How do we determine the impact of the learning – behavior, productivity, capability?
 - e. How do we evaluate “experience acceleration”?
 3. ExpAcc Technology:
 - a. How do you create an open experience architecture that provides the appropriate tradeoffs for flexibility vs. efficiency?
 - b. How do you select the most appropriate technologies?
 - c. How do you ensure the validity of the experience (simulation)?
 - d. What needs to be done to make the experience seem authentic?

4.0 ExpAcc Concepts

As shown in Figure 1, the ExpAcc research project has defined several concepts that will be used to drive the content of the simulator prototype. These are:

- A competency taxonomy,
- A collection of “Aha” moments, and
- A set of challenges and land mines.

Competency Taxonomy

One goal of the ExpAcc is to increase the learner's level of systems engineering competency with each use. However, individual companies, consortiums, and professional societies have each developed SE competency models creating a challenge (or opportunity) in the determination of which systems engineering competency model provides the best fit. Descriptions and comparisons of commonly used or available SE competency models can be found through a variety of sources.¹⁹⁻²¹ The ExpAcc research team chose to combine the following three models into a single competency taxonomy for the project:

1. The Systems Planning, Research Development, and Engineering (SPRDE) Systems Engineering (SE) and Program Systems Engineer (PSE) competency model, known as the SPRDE-SE/PSE.
2. The SERC Technical Lead Competency Model
3. A Critical/Systems Thinking Competency Model

The SPRDE-SE/PSE competency model is comprised of 29 competency areas with 45 unique elements of competency defined. These are grouped according to three primary “units of competences” – analytical, technical management, and professional. The analytical unit covers

13 competencies related to the technical base for cost and aspects of the system life cycle. The technical management unit addresses 12 competencies focused on the technical side of project management. The professional unit covers the broader competencies of communication, problem solving, systems thinking and ethics.²²

The SERC Technical Lead Competency Model, shown in part below, includes 12 primary categories of competencies and 71 unique competencies; the 12 primary categories are:²³

1. professional and leadership development
2. enterprise leadership and management
3. resource management
4. business acumen
5. risk and security
6. program assessment and recovery
7. project conception
8. project planning, management, and control
9. systems engineering thinking and perspective
10. technical management
11. production, product transition, and operations
12. technical acumen

The first 11 categories covered broad areas of systems engineering and technical leadership while the 12th category focuses on the specific technical discipline expertise and the associated domain.

The critical/systems thinking competency model is comprised of nine specific competencies as shown below:²⁴

- Critical thinking:
 1. strategic thinking
 2. essential thinking
- Systems thinking:
 1. incorporate multiple perspectives
 2. engage in abstract thinking
 3. define/develop within “fuzzy scope” boundary
 4. understand/diversity of operational contexts
 5. identify relationships and dependencies (inter and intra)
 6. understand complex system behavior
 7. reliably predict impact of change to system

The resulting systems engineering competency taxonomy leverages the core competencies of systems and critical thinking with the extension of five additional areas of competency: systems engineering technical leadership, systems engineering implementation, technical management, program management, and other broad-based professional competencies. The final integrated model contains six main areas of competency, 20 sub-categories, and 87 unique competencies as

shown in Figure 1.²⁵

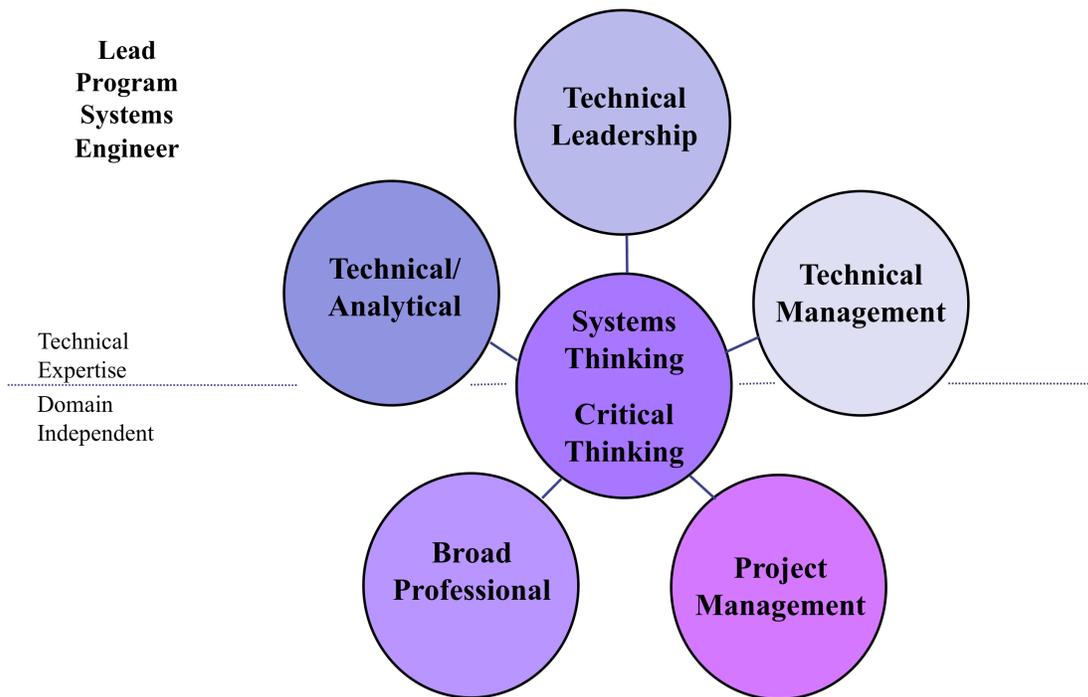


Figure 1. ExpAcc Competency Taxonomy²⁵

“Aha” Moments

Creating the appropriate “scar tissue” requires the proper experience with the proper amount of emotional tension coupled with reflection on the meaning of the experience and incorporating into one’s experience set. This process of reflective learning “involves returning to the experience (replaying it in the mind and/or recounting it to others), attending to the feelings accompanying the experience and its memory, re-evaluating the experience and drawing lessons from it.”²⁶ Making use of empirical evidence and the recognition of patterns/anti-patterns are important for the creation of “right” heuristics. Making sense of the experience generally requires the formulation of a mental model to transform this into an internalized principle that can be effectively recalled and utilized in the future. In order to create a realistic authentic environment for maturing as a systems engineer, the ExpAcc gathered “Aha” moments – moments that reflected a valuable insight or career turning point or significant lessons learned – from ongoing interviews of systems engineering experts. These experts were asked to recall specific moments in their past experiences where they had a defining moment that helped shape their maturity of systems engineering expertise. “Aha” moments typically reflect a “lessons learned” from a mistake or failure. Lessons learned were also referenced from *The Little Book of Bad Excuses*²⁷ and *Some ‘Best Practice’ and ‘Worst Practice’ Questions*.²⁸ These mistakes or fallacies were then grouped into the following four categories:

1. Information Gathering/Sharing:
 - a. Not understanding the needs of all of the stakeholders and maintaining communication throughout the life-cycle (sometimes requires living the customer experience)
 - b. Not helping your customer understand the user's real needs (otherwise the system will not be used)
 - c. Believing that there is a single objective view of systems requirements. There are no definitive requirements for a system and they may change during the project development and deployment. The true needs may not even be reflected by the stakeholders, but the stakeholders need to be satisfied to get their support. Individual stakeholders can come and go during a program, and requirements can likewise change due to changes in mission, environment and technology.
 - d. Looking at the data you have rather than the data you need
 - e. Listening to what people say rather than pushing to hear what you need to know
 - f. Believing a single source of information. One must find the right mix of information from a number of sources and operate with incomplete and imperfect knowledge.
 - g. Not reexamining assumptions when conditions have changed
 - h. Forgetting the capabilities of the user/maintainer
 - i. Forgetting the fact that in the case of ambiguous language (and all language is to a certain degree), suppliers interpret it to be a low bar and customers interpret it to be a high bar (each generally interprets it to their advantage)
 - j. Forgetting the obvious when focusing on the more obscure ("it was obvious what the system was supposed to do")
 - k. Optimizing for the known needs rather than the likely needs over the lifecycle
 - l. Not sharing innovations and best practices, etc..., ("you have a farther reach than you think you do")

2. Processes:
 - a. Not getting stakeholder commitment early in the process
 - b. Cutting corners to make milestones rather than making the end date
 - c. Starting design too soon
 - d. Putting off integration and validation until the end
 - e. Neglecting the Total Cost of Ownership (TCO) and focusing on first article
 - f. Relying on process rather than people (or vice a versa)
 - g. Losing sight of the principles behind the process (letter of the law vs. the spirit) or believing that creating the artifacts (even after the fact) is the same as following the process
 - h. Trying to change process or policy by forcing the team to use a new IT system
 - i. Not having a way to report and track problems/issues especially when working between many subsystems/teams.
 - j. Not integrating specialists across the life cycle of the system early in the process

3. Decision Making:
 - a. Putting off pain by not addressing problems early
 - b. Over reacting to near term issues. The result may be the "bullwhip" effect where reacting quickly to a spike in demand can result in long-term over provisioning

- c. Expecting that things will just get better and that slips can be made up later without making fundamental changes early
- d. Mistaking activity for progress
- e. Moving to black and white beliefs rather than living in the actual gray
- f. Being inflexible to change, particularly changes in your own role
- g. Ignoring or not being aware of the consequences of early decisions and the time lag necessary to change them. Early decisions have consequences. While it is possible to change course during the project development, it can be costly. Early decisions matter.
- h. Letting ego get in the way of balanced decision making; staking one's ego on a particular solution.
- i. Not being able to be guided or informed by intuition (holistic Trust your intuition or gut feel, take some risks
- j. Failing to provide proper balance (related to moving to black & white viewpoints). Some questions an SE needs to answer: When is it good enough? Where is ground truth? What is the rationale (why?) used?
- k. Forgetting the needs, capabilities, culture and current processes and practices of the end user in the quest for technical perfection, the "best" solution is not always the best technical solution

4. Conceptual Issues:

- a. Failure to develop a simple mental model. Without this, complications can detail can be overwhelming and the ability to prioritize is greatly compromised. E.g., understand and document the concepts and principles upon which an architecture is based and keep the architecture as simple as possible, but no simpler.
- b. Mistaking a model for reality (not knowing the limitations of the model) (e.g., you can't use 100% of the "available" resources or even close to 100%; you cannot control human behavior nor assume humans will act how you want them to).
- c. Failure to take into account Model inaccuracies in the system – such as physical behavior of components to human behavior – accurately rather than how you want them to act – within the operational context of the system be it controlled or in the natural world.
- d. Overestimating the ease of reuse
- e. Designing something that can't be tested or validated. Need to take the total life-cycle effect into account when making design decisions
- f. Not allocating support and effort for the –ilities, dependability, reliability, serviceability, safety, security, etc.: upfront and trying to work them in at the end after functionality has been addressed. Most effort is usually spent on the exceptional cases in a program.
- g. Undervaluing the value of flexibility, both in the system developing the system and in the system being developed.
- h. Attempting to force collaboration without aligning an incentive system which supports it
- i. Underestimating your own abilities [although it helps] when others believe in you.

Challenges and Landmines

Challenges and landmines represent the various obstacles that the learner might face that stand in the way of success. Challenges are obstacles that are factored in at the initiation of the experience, whereas landmines are dynamic obstacles that appear during the simulation based on the user's profile and relative success during the simulation. The ExpAcc takes as inputs the learner's profile and his/her past history on the ExpAcc, along with the competencies and "Aha's" that have been targeted by the current experience. The ExpAcc then selects from the available set of challenges and landmines, and experience scenarios for use in the experience.

The ExpAcc uses the simulation models and data along with challenge "control" to create an experience scenario which has been optimized to meet a specific set of project goals relating to schedule, cost, capabilities and quality. The scenario also contains an objective utility function of the project metrics that can be used to determine the overall success of the project. The selected challenges and landmines are then used to create deviations in the optimized scenario to reflect the challenges. For example, the required schedule and/or budget can be reduced beyond what is achievable, resources can be misallocated, desired features can be removed, etc. The ExpAcc is thus able to determine the project's level of success if the learner makes no changes to the project. The degree to which the learner is able to increase the score is a measure of success, and if the score falls below this level it is a sign of the counter productive actions of the learner. Landmines are tripped when certain parameters in the experience go beyond a certain set limit. For example, if the experience is not sufficiently challenging for the learner, a new stakeholder can be introduced who has new requirements, or productivity in a team can be reduced due to personnel being pulled to other projects.

At this point the project challenge has been incorporated in the deviations that have been created in the project goals, resource allocation, productivity, etc. as reflected by the project utility function. The ExpAcc thus has the "gold standard" view of the experience. The next step is to determine how the learner might be able to discover these issues and the difficulty level in which they will be revealed. The issues are revealed passively through "documents" provided and actively through interactions with non-player characters (NPCs). The difficulty factor, as determined by the ExpAcc, is set by the level of ambiguity with which this information is presented to the learner. This can be done in a number of ways. One method is to reduce the precision of the information so that it has varying levels of ambiguity. Another means is by presenting conflicting information. In this case, the learner has to learn who or what to believe. Other means of creating ambiguity are through missing stakeholders, inaccurate communication, communication lost in translation due to medium issues (e-communication), culture or context, etc. In any case, the level of ambiguity is set in a self-consistent way. It should be noted that ambiguity could also be used in the construction of land mines.

Based on the learner's ability to discover the challenge issues, they need to take corrective action. As a program systems engineer, the learner is a technical lead for the program. Thus, the corrective actions of the role focus on identification and mitigation of technical problems and risks. Significant communication, facilitation and coordination skills are required, especially for programmatic changes that must be agreed upon by various program personnel and stakeholders and approved by the program manager.

5.0 Recommendations for Systems Engineering Curriculum Development

This section addresses the following recommendations for using lessons learned on the ExpAcc project to enhance systems engineering curriculum in the classroom:

- Applying ExpAcc Concepts (see Section 4.0)
- Assessing and Providing Feedback to the Learner
- Developing Mini-Case Studies
- Leveraging Subject Matter Experts (SME)
- Anticipating Challenges

Applying ExpAcc Concepts (see Section 4.0)

The overall concept of creating an experience with challenges and landmines that leads to the user experiencing an “Aha” moment that, in turn, accelerates the user’s development in a targeted systems engineering competency is not unique to a computer-based environment. These concepts can also be applied to a live classroom simulation. The essential steps of the process are:

1. Select the systems engineering competency or competencies to be targeted. The selection can be determined based on the learner’s needs (see ‘Assessing and Providing Feedback to the Learner’) or a specific set of competencies can be selected based on the learning objectives of the course or the needs of the customer.
2. Define “Aha” moments that support the learner’s ability to achieve the intended lesson and advance in knowledge and ability in the targeted set of competencies. See “Leverage Subject Matter Experts (SMEs) for additional ideas in this area.
3. Define specific challenges, adding landmines depending on the outcomes of the learner’s decision points, that are conducive to creating the right environment or context within which the learner can reach an emotional state that will lead them to experience the intended “Aha” moment.

The success of the classroom simulation will depend on additional considerations, covered in the following sections.

Assessing and Providing Feedback to the Learner

As previously discussed, the ExpAcc was conceived to help users create “scar tissue” (learn from mistakes) in a safe environment –that is to say one in which they can face challenges and make errors without there being any negative ramifications to themselves, others, or work-related outcomes. However, in order to “learn from mistakes” users must also receive clear and actionable feedback as the basis for reflection, subsequent skill practice and behavior change. Therefore, performance assessment and feedback are important aspects of the overall learning process. As depicted in Figure 2, that process can be described in relation to the Experiential Learning Model developed by Kolb, 1984.²⁹ For example, profile building engages users in initial reflection on their related skills, personal qualities and experiences. Such baseline reflection also helps individuals frame personal learning objectives and expectations that then

become benchmarks against which to receive simulation performance feedback and to establish developmental objectives.

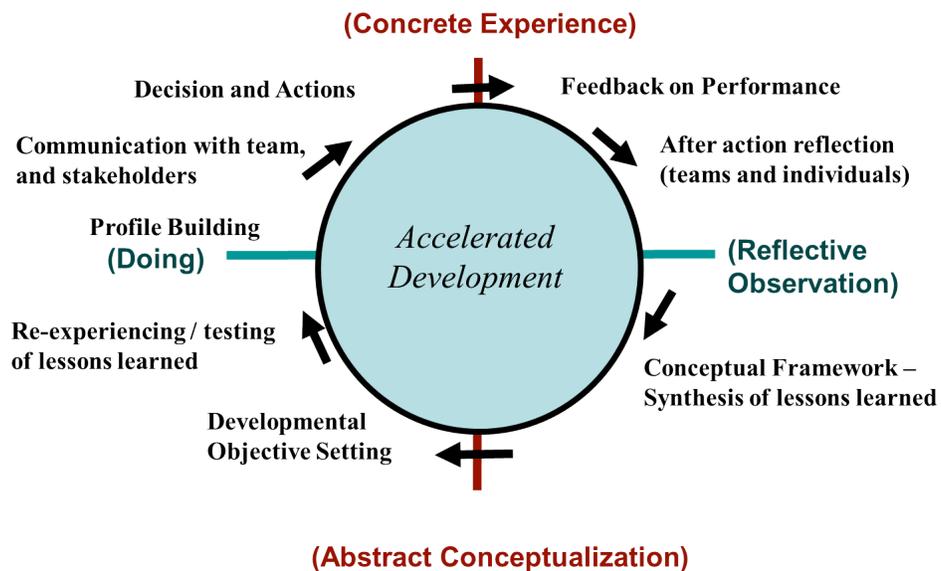


Figure 2. All Phases of Experiential Learning to be Engaged

Profile building is followed by the *concrete experience* aspects of the simulation. This includes communication and interaction with other players, system avatars and scenario content as well as actual user decision making and actions.

Those decisions and actions then become the basis for assessment and feedback, thereby moving the user into the next phase of the experiential learning process. Specifically, *reflective observation* is facilitated by performance feedback and opportunities to consider its implications for future performance in the simulation and on the job. Feedback and reflection then becomes the springboard for helping individuals engage in *abstract conceptualization*. This occurs by providing users with opportunities to integrate and synthesize what they have experienced. Examples include helping users recognize ways in which their performance was or was not consistent within and across simulations; the extent to which simulation performance was aligned with initial self-assessments of skills and abilities; and the ways in which simulation performance may reflect or help to better explain actual job performance.

The setting of developmental objectives derived from synthesis and integration is an important outcome. Therefore, it is critical that accountability for objective setting be built into the overall learning model. This includes work related goal setting and also goals for subsequent simulation engagement. Objective setting ultimately then provides users with new benchmarks against which to re-engage with the simulation in order to test lessons learned or, in terms of experiential learning, to *actively experiment* in the pursuit of mastery and new knowledge.

In support of this framework, prior to beginning their work with the ExpAcc, users are asked to provide information about their personal qualities that are likely to influence how they would approach the scenarios and challenges the accelerator will pose. Examples include background

and interests, educational experiences, self-assessments of competencies, and other potentially relevant qualities such as one's learning and decision-making style. However, rather than having users enter a complete profile all at once they are prompted to provide profile information at the points in time when it is most relevant to the ExpAcc challenge or scenario they are about to undertake. For example, if a particular scenario is designed to challenge a user's skills with respect to problem solving and recovery they are asked to complete the self assessment for that competency prior to beginning that scenario.²⁵

This information serves a number of important functions. First, it provides users with an initial point for self-reflection about their skills, interests and abilities. Second, the information they provide serves as a baseline for performance feedback. For example, competency self-assessments completed as part of a profile can be compared to system-generated ratings on those same skill areas based upon actual performance in the ExpAcc scenarios and challenges. Third, profile information can be used to tailor the type and or difficulty level of the accelerator scenario a user will experience. Finally, it can also be used to help shape the manner in which feedback is provided. For instance, a user whose predominant learning style is visual might be provided with most of his or feedback graphically, while another user with a different learning style might receive most of her feedback in written form.

Developing Mini-Case Studies

Case studies are a classic approach to providing experience-based learning. A case study can be used to illustrate successes, failures and underlying causes of both. Systems engineering has a rich set of potential case studies that can be applied from government and industry. For example, Laufer et al.³⁰ compiled a set of four case studies that focus on systems engineering and management, two from NASA and two from the Air Force. The case studies use a story-based approach, relying on extensive interviews with a number of key individuals in each case to provide multiple, interwoven perspectives on project successes and shortcomings. Here, we take a more structured approach to constructing case studies, by using the concepts of the competency taxonomy, the learner profile, "Aha" moments, and challenges and landmines. The learning objectives for a particular experience are driven by a subset of the competency taxonomy, i.e., the competencies for which skill improvement is desired. For example, one learning objective might be to have the learner demonstrate proficiency in identifying relationships and dependencies in a system – relating to that specific competency under systems thinking.

The learning objectives are set in the framework of an experience, derived from a case study, which is set within the context of a domain. This domain, for example, could be design and development of an unmanned aerial system (UAS), or the integration of sub-systems for a planetary landing vehicle. The domain governs various elements to which the learner is exposed in the experience, as well as their inter-relations. These include such things as processes used in system design and development and in project management, documents and information to which the learner has access, events that can occur, the cast of NPCs with which the learner interacts, and their organization into roles and working groups. For instance, in an experience based on UAS design and development, the process is based on standard acquisition policies and processes, which progress from conceptual design to architecture design, sub-system development, integration and test, then production. Teams work on various aspects; and teaming

structure typically changes as the project progresses. The learner can access design and engineering documents, project schedules, etc., and the learner can also interact with NPCs to gather information on the project that may not be available in documents (e.g., an NPC's opinion of current progress). The experience also contains a set of decisions that the learner can make that may cause problems or may be needed to help ensure project success. For instance, a decision may be to reallocate project resources to an area that seems to have fallen behind, or to order a test, or re-examination of assumptions.

To develop “scar tissue”, various “Aha” moments are designed into the experience. These “Aha” moments are taken from the generic set and are customized to the domain of the particular experience. As an example, consider the generic “Aha” moment involving the need to re-examine assumptions when conditions have changed. A specific contextualization of this occurred in a real project. During a test flight for the second test flight vehicle of the Global Hawk UAS in 1999, the vehicle was destroyed, resulting in a \$45 million payload loss and the loss of the program's only integrated sensor suite.³¹ The system had originally been developed by DARPA and then transferred to the Air Force. Assumptions about the system's frequency management were not re-examined. In particular, the Air Force's frequency management was not compatible with the system, and a flight termination signal was issued that unintentionally resulted in the vehicle's destruction. A learner's decision to re-examine assumptions could have avoided the failure. Such “Aha” moments should align with learning objectives. In this case, a relevant learning objective would be demonstrating proficiency in identifying relationships and dependencies.

In terms of customizing the experience to a particular learner, a set of challenges and landmines are designed into the experience, and they are invoked depending on the learner's skill level as evidenced from the learner profile. An experienced learner might be given more challenges, such as the transfer of a project to different stakeholders or failure of a key sub-system design. Figure 3 illustrates a framework for constructing case study experiences using the ExpAcc framework.

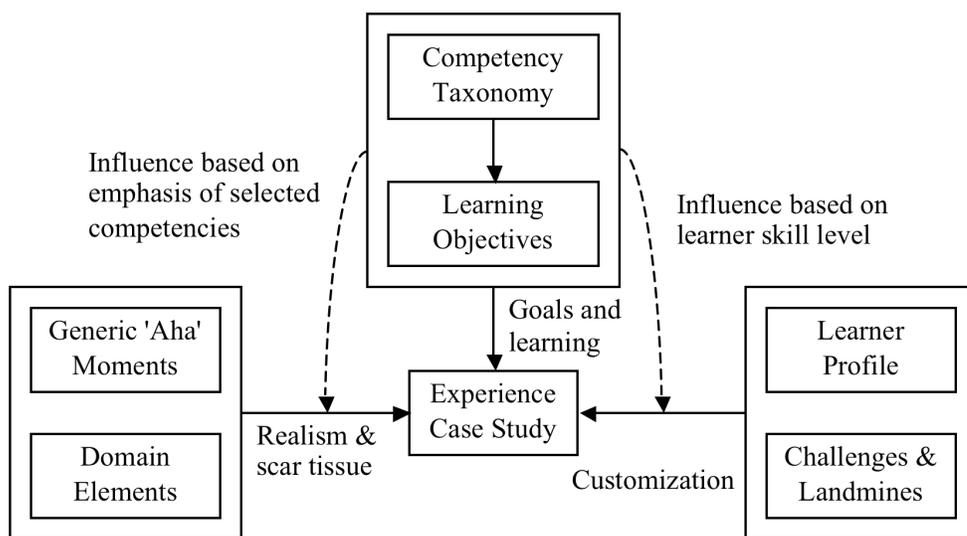


Figure 3. Case study experience framework

Leveraging Subject Matter Experts (SMEs)

A learner interacts with the experience through role-playing, in the context of the mini-case study established for a particular learning scenario. A principal means of enacting this role-play is through verbal exchanges with the other human characters that populate the simulated project environment. These exchanges may be conducted through various communication modes, including face-to-face meetings, telephone or video conference calls, electronic mail, and written documents. We characterize each of these exchanges as a form of inter-character dialogue.

In the multi-player implementation of the experience, these characters may actually be other human learners. In the initial single-player implementation, however, all characters other than the learner are NPCs whose actions are directed under program control. To provide verisimilitude, the dialogues of these NPCs are constructed in consultation with subject matter experts (SMEs) chosen for the environment of each learning experience. For the prototype implementation, these SMEs are experienced in program management and aerospace systems engineering.

To provide a more natural flow of learner-NPC dialogue in the experience, lengthy speeches are avoided in favor of more interactive exchanges. Since natural language processing for these conversations was deemed too difficult to implement, they are instead modeled using a hub-and-spoke dialogue system (see Figure 4) a type of finite state machine where each hub represents the state of a conversation, and each spoke the conversational alternatives in that state. In interacting with an NPC, the learner is presented with the alternative topics of the conversation (the spokes) for each state of the NPC (the hubs). When the learner chooses a particular alternative, the learner avatar speaks an inquiry representing the topic, and the NPC avatar responds appropriately. This results in a transition to another (possibly the same) state, where the learner is again presented with the conversational alternatives.

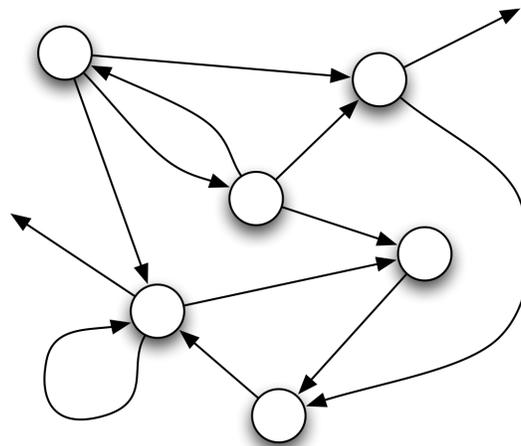


Figure 4. Hub-and-spoke dialogue encoding

This approach provides a dynamic experience for the learner, in that conversations are learner-directed, yet still permits the experience to encode the SME information to be imparted as a set of interrelated conversational states. A particular challenge for the learner in this environment is

therefore to ensure that the proper conversational alternatives are fully explored within the allotted time in the scenario. It may be that particular conversational states impart information that is critical to solving the crises of a particular learning scenario, or that the learner is scored based on a full exploration of the relevant topics, as determined by the SME in the experience development process.

Dialogues are constructed through offline role-playing with the SMEs during the development of a learning experience. The SMEs are first consulted to build a repertoire of topics that the learner must explore to meet the challenges of a particular learning scenario. The content developers then engage the SMEs in conversation over these topics, and explore alternatives conversation threads that the SME has experienced or suggests as relevant. These conversations are recorded, and in post-processing the inquiries and responses are encoded using the hub-and-spoke system described above. Finally, the SMEs validate the encoding and experience goals by engaging their simulated selves in conversation using the dialogue system.

It is recognized that the content development process necessary to implement this approach is complex and time-consuming, and places high demands on the SMEs. A critical component of doing systems engineering well, however, is discovering what information is available from whom, and how to elicit that information in a time-constrained environment. The relatively open-ended dialogue approach allowed by using the hub-and-spoke encoding, coupled with the content richness afforded by engaging SMEs in development of that dialogue, provides a challenge that is expected to be somewhat realistic and engaging for the learner.

Anticipating Challenges

To be successful, the planning for creating a classroom experience will need to factor in potential challenges. Two of the challenges discussed in this section relate to the actual 'feel' of the experience (compelling and authentic) while the third addresses the challenge associated with assessing the actual success of the experience in accelerating learning.

Compelling Experience: The ExpAcc seeks to engage the learner with compelling and realistic contexts. Proponents of video games and simulations for learning point to their ability to provide meaningful and compelling experience as a significant benefit to their use⁶⁻¹⁰. Shaffer¹⁵ and Gee¹⁶ describe how by situating a learning experience within a meaningful context, players are able to experience, practice, and acquire the unique epistemology or literacy associated with that context. This echoes the foundational learning theory of situated cognition and its focus on placing learning into meaningful contexts.¹³ The ExpAcc creates a compelling experience by placing trainees in a meaningful context, giving them control and ability to make impactful choices, and the opportunity to practice important skills in a safe environment with realistic outcomes and feedback.

Realism and Authenticity: Clearly, in any experience-based learning environment, realism and authenticity are critical aspects that underpin success of the environment. Here, realism refers to the degree with which the experience conforms to "real world" and "on the job" situations. Authenticity refers to the extent to which the experience captures key elements of the real world that are critical to learning. As an example, consider the system documents to which a learner

has access during the experience. In a real setting, there are many documents, each ranging up to several hundred pages or more. In the ExpAcc, we do not expect that users would be interested in reading the entire set of documents available to a systems engineer for a real system. Rather, the goal is to provide a realistic format for each critical document, with key information provided, but to omit an excessive amount of non-essential information. Realism and authenticity to date are being addressed by two primary means:

- Review of case study materials available in monographs and reports documenting previous projects to generate material for experience,³¹⁻³⁶
- Consultation with subject matter experts to validate experience materials and suggest improvements.

Assessment of Accelerated Learning: To the extent that the acceleration of behavior change and skill development are major objectives for using the ExpAcc it is important that they be assessed. There are several different types and sources of assessment that can be employed. For example, when properly timed multisource rating (e.g. 360 ratings) can be used to obtain baseline data with respect to the way a user's effectiveness as a systems engineer is perceived on the job. These initial assessments can be then followed up with subsequent evaluation several months later. At the very least, initial self-assessments of relevant skills and abilities can provide important benchmarks for measuring change. Along the same lines it can also help to track users' self-assessments of other important meta-skills and basic social-cognitive capabilities that we know underlie human learning across a diverse set of activities.^{37,38} Examples include assessing the amount of change in a user's self-efficacy³⁹ and self-identity⁴⁰ with respect to their systems engineering proficiency after having gone through the challenges posed in the ExpAcc.

Tracking performance changes within the context of the ExpAcc's challenges themselves is another important approach that should be employed. For example, if in fact the simulator is accelerating the user's development, we would expect to see improvements as they take on subsequent simulation scenarios. Finally, there are important areas of assessment related research that should also be pursued. They include investigating the overall efficacy of the ExpAcc approach. Controlled studies should be undertaken for example, to compare the extent and rate of skill development attained by persons going through the ExpAcc to others not exposed to the simulator and or who have been exposed to other kinds of training and development. We also need to learn more about the extent to which individual differences play a role in if or how individuals develop by exposure to the ExpAcc. Examples of factors to consider include learning style,²⁶ feedback orientation,⁴¹ and goal orientation.⁴²

6.0 Conclusion

The International Council on Systems Engineering (INCOSE) publication *Systems Engineering Vision 2020* predicted that the "Use of technologies such as simulation, visualization, and gaming will lead to innovations in systems engineering education."⁴³ (p. 6) Due to the growing demand for SEs, the retiring of the baby boomer generation and the time it takes to mature as a SE, this vision is critical for addressing the resultant shortfall of experienced SEs. The goal of the ExpAcc is to accelerate the time it takes to mature as a systems engineer by exposing the learner to realistic, emotionally engaging simulated experiences that allow the learner to grow "scar

tissue” earlier in the SE maturity cycle. The intention of the project is demonstrate the feasibility of accelerating SE maturity through the use of technology. A benefit of the research to date is the development of an approach that can potentially be used to accelerate learning through real life simulations in the classroom. Recommendations include: applying the ExpAcc concepts (challenges and landmines, “Aha” moments, and SE competencies), assessing and providing feedback to the learner, developing mini-case studies, leveraging Subject Matter Experts (SMEs), and anticipating challenges. Once the ExpAcc prototype is developed, additional information will be made available on the research and testing results of the pilot.

Acknowledgements

This material is based upon work supported, in whole or in part, by the Systems Engineering Research Center (SERC). SERC is a federally funded University Affiliated Research Center managed by Stevens Institute of Technology. We are grateful to all the members of the Experience Accelerator team including members of participating universities, sponsor members, and subject matter experts.

References:

1. Bagg, T. C., Granata, R. L., Brumfield, M. D., Casey, C. A., & Jamison, D. E. (2003). “Systems engineering education development (SEED) case study.” In *Proceedings of the 13th annual international symposium, INCOSE 2003, crystal city, virginia, USA, june 29 - july 3, 2003*.
2. Charette, R. (2008). “What’s wrong with weapons acquisitions?” *IEEE Spectrum*, November.
3. Ryschkewitsch, M., Schaible, D., & Larson, W. (2009, January 20). “The art and science of systems engineering: Long version.” *NASA*. Retrieved March 1, 2009, from http://www.nasa.gov/pdf/311198main_Art_and_Sci_of_SE_LONG_1_20_09.pdf
4. Dubey, R. (2006) "Study and Analysis of Best Practices for the Development of Systems Engineers at a Multi-National Organization." *Thesis*, MIT, Boston, MA, USA.
5. NDIA SE Division (2010, July). “Top systems engineering issues in department of defense and defense industry (Final 9a-7/15/10).”
6. Aldrich, C. (2004). *Simulations and the future of learning: An innovative (and perhaps revolutionary) approach to e-learning*. San Francisco: Pfeiffer.
7. Federation of American Scientists. (2006). *Harnessing the power of video games for learning [Electronic Version]*. Retrieved June 24, 2007 from <http://www.fas.org/gamesummit/Resources/Summit%20on%20Educational%20Games.pdf>.
8. Foreman, J., Gee, J. P., Herz, J. C., Hinrichs, R., Prensky, M., & Sawyer, B. (2004). *Game-Based learning: How to delight and instruct in the 21st century*. EDUCAUSE Review, 39(5), 50-66.
9. Prensky, M. (2001). *Digital Game-based Learning*. New York: Mcgraw-Hill.
10. Quinn, C. N. (2005). *Engaging learning: Designing e-learning simulation games*. San Francisco: Pfeiffer.
11. Gardner, H. E. (1999). “Multiple Approaches to Understanding.” In C. M. Reigeluth (Ed.), *Instructional-design theories and models: A new paradigm of instructional theory* (pp. 69-89). Mahwah, NJ: Lawrence Erlbaum Associates.
12. Driscoll, M. P. (2005). *Psychology of learning for instruction (3rd ed.)*. Boston: Pearson Education, Inc.
13. Brown, J. S., Collins, A., & Duguid, P. (1989). “Situated cognition and the culture of learning.” *Educational Researcher*, 18(1), 32-42.
14. Herrington, J., & Oliver, R. (1995). “Critical characteristics of situated learning: Implications for the instructional design of multimedia.” In J. Pearce & A. Ellis (Eds.), *Learning with technology* (pp. 235-262). Parkville, Vic: University of Melbourne.

15. Shaffer, D. W. (2006). *How computer games help children learn*. New York: Palgrave Macmillan.
16. Gee, J. P. (2003). *What video games have to teach us about learning and literacy*. New York: Palgrave Macmillan.
17. Ericsson & Charness (1994). "Expert performance: It's structure and acquisition." *American Psychologist*, 49(8), 725-747.
18. Ross, P. E. (2006). "The expert mind." *Scientific American*, 295(2), 46-53.
19. Kasser, J., M. Frank, and Y. Y. Zhao. 2010. "Assessing the competencies of systems engineers." In *Proceedings of the 7th bi-annual european systems engineering conference (eusec), stockholm, sweden, may 23-26, 2010*.
20. Ferris, T. L. J. 2010. "Comparison of systems engineering competency frameworks." In *4th asia-pacific conference on systems engineering (APCOSE 2010), keelung, taiwan, october 4-6, 2010*.
21. Squires, A. 2011. "Investigating the Relationship of Online Pedagogy to Student Perceived Competency Knowledge Development in Systems Engineering Education". *Thesis*, Stevens Institute of Technology, School of Systems and Enterprises.
22. DAU. 2010. "SPRDE-SE/PSE competency model 4/14/10 version." <https://acc.dau.mil/CommunityBrowser.aspx?id=315691&lang=en-US> (accessed May 25, 2010).
23. Gavito, V., Verma, D., Dominick, P., Pennotti, M., Giffin, R., Barrese, T., et al. (2010, December 13). "Technical leadership development program: Final technical report SERC-2010-TR-013." Stevens Institute of Technology, Hoboken, NJ.
24. Squires, A. (2007). "Qualifying Exam: Systems Thinking and K12 Education." *Stevens Institute of Technology*, Hoboken, NJ.
25. Squires, A., Wade, J., Dominick, P., Gelosh, D. (2011) "Building a Competency Taxonomy to Guide Experience Acceleration of Lead Program Systems Engineers", *Proceedings from the Ninth Annual Conference on Systems Engineering Research (CSER)*, Redondo Beach, CA, April 14-16, 2011.
26. Shamir, Boas; Eilam, Galit; "What's your story?" A life-stories approach to authentic leadership development." *The Leadership Quarterly* 16 (2005) 395-417.
27. "The Little Book of Bad Excuses", *Computers & Concepts Associates*, June 1998;
28. "Some 'Best Practice' and 'Worst Practice' Questions", *Airlie Council of the Software Acquisition Best Practices Initiative*
29. Kolb, D. (1984). *Experiential Learning: Experience as the Source of Learning and Development*. Prentice-Hall, Englewood Cliffs, NJ.
30. Laufer, A., Post, T., & Hoffman, E. J. (2005). "Shared voyage: Learning and unlearning from remarkable projects." *National Aeronautics and Space Administration*. Report NASA SP-2005-4111. Washington, DC.
31. Drezner, J. A., & Leonard, R. S. (2002). "Innovative development: Global Hawk and DarkStar - Flight Test in the HAE UAV Program." *RAND*. Report MR-1475-AF. Santa Monica, CA.
32. GAO (2009). *Joint Strike Fighter: "Strong risk management essential as program enters most challenging phase."* *Government Accountability Office*. Report GAO-09-711T. Washington, DC:.
33. GAO (2010). "Navy's ability to overcome challenges facing the Littoral Combat Ship will determine eventual capabilities." *Government Accountability Office*. Report GAO-10-523. Washington, DC:.
34. Leonard, R. S., & Drezner, J. A. (2002). "Innovative development: Global Hawk and DarkStar -HAE UAV ACTD program description and comparative analysis." *RAND*. Report MR-1474-AF. Santa Monica, CA.
35. Taubman, P. (2008). "Costly lesson on how not to build a Navy ship." *New York Times*, April 25.
36. Thirtle, M. R., Johnson, R. V., & Birkler, J. L. (1997). "The Predator ACTD: A case study for transition planning to the formal acquisition process." *RAND*. Report MR-899-OSD. Santa Monica, CA.
37. Bandura, A. (2006), "Toward a psychology of human agency", *Perspectives on Psychological Science*, 1(2), 164-80.
38. Cervone, D. (1991). "The two disciplines of personality psychology." *Psychological Science*, 2, 371-372.
39. Dweck, C.S. & Leggett, E.L. (1988) "A social cognitive approach to motivation and personality." *Psychological Review*, 95, 256 – 273.
40. Avolio, B. & Hannah, S.T. (2008). "Developmental readiness: Accelerating leadership development." *Consulting Psychology Journal: Practice & Research* 60 (4), 331-347
41. London, M. & Smither, J. (2002). "Feedback orientation, feedback culture, and the longitudinal performance management process." *Human Resource Management Review*, 12, 81-100.
42. Vandewalle, D. (1997). "Development and validation of a work domain goal orientation instrument." *Educational and Psychological Measurement*, 57, 995-1015.
43. INCOSE Technical Operations (2007, September). "INCOSE-TP-2004-004-02: Systems engineering vision 2020, version 2.03." *International Council on Systems Engineering*, Seattle, WA.