DOI: 10.20472/IAC.2016.025.027

CHRISTIAN GREUEL
SRI International, United States

JOHN MURRAY
SRI International, United States

CINDY ZIKER
SRI International, United States

LOUISE YARNALL
SRI International, United States

ALEXANDER KERNBAUM
SRI International, United States

INTELLIGENT COACHING SYSTEMS IN HIGHER-ORDER APPLICATIONS: LESSONS FROM AUTOMATED CONTENT CREATION BOTTLENECKS

Abstract:

Intelligent virtual environments hold promise for improving learner-directed instruction in context. These systems trace the progress of learners performing tasks and can insert immediate coaching to focus learner attention, link knowledge to activity, and accelerate the shift from abstract to concrete learning. Such technology has been used to improve self-directed learning of hands-on procedures, but also shows promise for higher-order applied fields, such as engineering.

To realize this vision, research must address the formidable bottlenecks around content creation and build understanding of the types of reusable content libraries relevant to the subject domains. This presentation describes two projects for interactive training that developed prototypes for automated content creation. A third project is presented that illustrates a suite of learning object libraries to support engineering instruction.

The first project, SAVE, uses a 3D browser-based simulation environment not only for hands-on training in equipment maintenance, but also for automating the generation of instructional exercise solutions. SAVE allows a subject matter expert to use the interactive simulation for modeling the correct steps of a procedure, thus providing a rapid way to extract their knowledge. The system collects a trace of the expert's activity, which becomes the reference against which learner activity is compared in automated assessment.

The second project, AR Mentor, delivers augmented reality overlays in head-mounted displays worn by student mechanics while learning to maintain terrestrial vehicles. An automated speech system interacts with the students as they perform equipment adjustments and troubleshoot system faults. To deliver audible step-by-step guidance, a prototype text-to-speech translator was developed to convert steps as written in the technical manual into the voice of a virtual coach.

The third project, SiMPLE, developed a library of engineering computation objects to allow learners to

ISBN 978-80-87927-27-4, IISES

construct electromechanical simulations, and provides an intelligent coaching system to allow novice engineers to iteratively refine their design specifications. When a working simulation is achieved, the system is linked to a 3D printer for physical prototype production.

The first two projects demonstrate methods of using virtual intelligent technologies to accelerate training content production in hands-on domains: expert model tracing and technical manual translation. The third project provides the tools needed to support engineering instruction: object libraries with embedded computations, as well as scripts for design coaching, design testing, and physical prototyping.

Together, these projects illustrate the wide range of available, reusable libraries and the extensive opportunities for automating content creation in many socio-technical fields.

Keywords:

Intelligent coaching systems, Augmented reality, Interactive sociotechnical training, Automated educational content creation

JEL Classification: C63, I21

Introduction

Procedural skills are an increasingly pervasive requirement in today's world, in areas ranging from IT system administration to complex data analyses, from automotive equipment repair to intricate medical diagnosis. The acquisition of procedural skills requires learning by doing—learners gain knowledge by trying to solve challenge problems, exploring the usage and limitations of tools and techniques, getting feedback on oversights and mistakes, and requesting assistance in the face of impasses and confusion. Intelligent virtual environments (VEs) hold promise for improving learner-directed instruction in these contexts. Such systems trace the progress of learners as they perform training tasks, and can insert immediate coaching or provide performance evaluation to focus learner attention, link knowledge to activity, and accelerate the shifts between abstract and concrete learning.

VE technology is widely used to improve self-directed learning of hands-on manual procedures, but it also shows appreciable promise for the use of modeling tools in a diverse range of higher-order applied fields, such as design engineering, policy analytics, and econometrics. To realize this vision, research must address the formidable bottlenecks around content creation and explore the types of reusable content libraries relevant to the subject domains. In this paper, we describe two interactive training projects that developed prototypes for automated content creation. A third project illustrates a suite of learning object libraries to support engineering instruction.

The first project, Semantically-enabled Automated Assessment in Virtual Environments (SAVE), uses a 3D browser-based simulation environment for hands-on training in equipment maintenance, supplemented by automated generation of instructional exercise solutions. SAVE allows a subject matter expert (SME) to use interactive simulations for modeling the correct steps applied to given procedural tasks and provide a rapid way to extract their knowledge. The system logs an SME's activity, which becomes the reference model against which learner activity is compared in automated assessment. The second project, AR-Mentor, delivers augmented reality (AR) overlays in head-mounted displays worn by student technicians while they learn vehicle maintenance. An automated speech system interacts with the learners as they perform equipment adjustments and troubleshoot electrical faults. To deliver audible step-by-step guidance, a prototype text-to-speech translator was developed to convert steps as written in the technical manual into the voice of a virtual coach. The third project, Simulation for Manufacturing and Prototyping with a Learning Environment (SiMPLE), developed tools to allow learners to construct electromechanical simulations, providing an intelligent coaching system to enable them to iteratively refine their design specifications. These tools include object libraries with embedded engineering computations and suites of scripts for design coaching, design testing, and physical prototyping once a working simulation is achieved.

The first two projects demonstrate practical methods of using virtual intelligent technologies to accelerate content production in hands-on domains: expert model tracing and technical instruction manual translation. The third project provides the tools needed to support complex technical instruction and translation to physical domains. Together, these projects illustrate the variety of reusable system libraries for training and educational platforms, demonstrate a range of opportunities for automating content creation, and raise awareness on techniques for built-in evaluations and assessments. These usages of intelligent virtual environments and simulations offer key insights that apply to practical training systems in numerous technical and analytic disciplines.

Semantically-enabled Automated Assessment in Virtual Environments

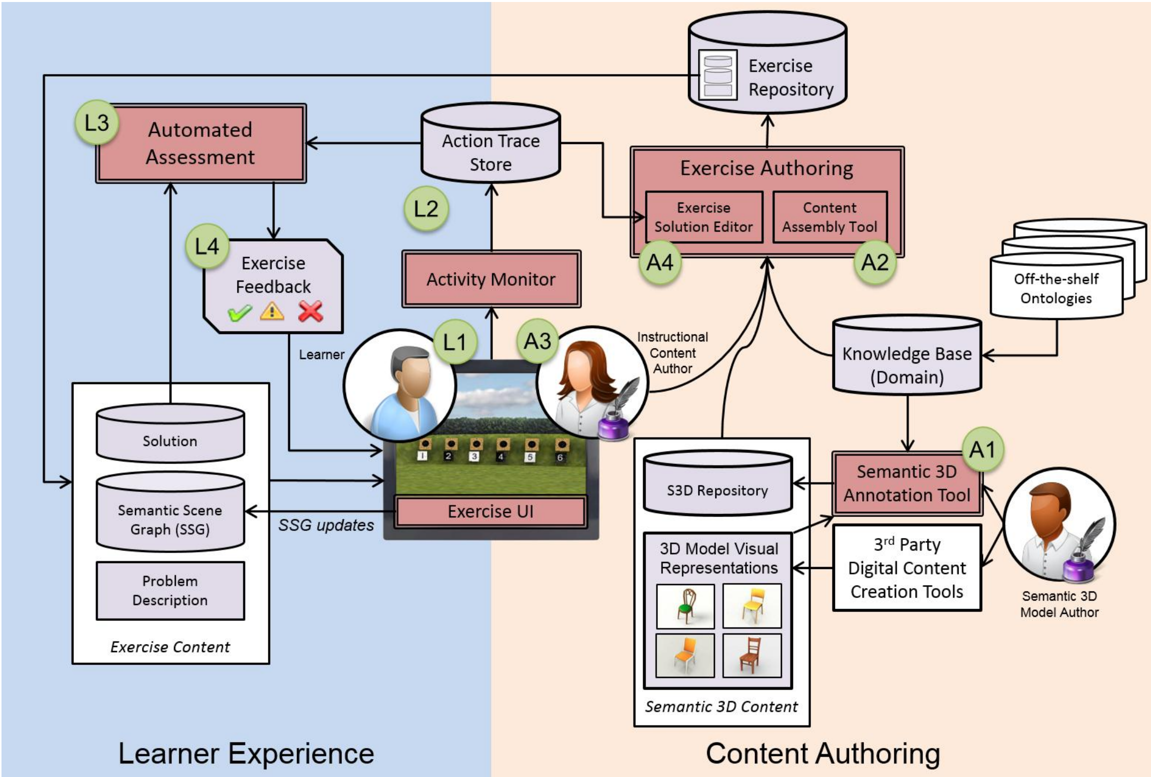
Virtual environments (VEs) provide an appealing vehicle for acquiring procedural skills, particularly in domains where real-world training incurs significant time, expense, or risk. Taking full advantage of VEs, however, requires reliable and consistent mechanisms to assess learner performance. Current assessment requires direct observation by an instructor. Creating mechanisms to enable meaningful automated assessment can both reduce the cost of using VEs for training and open the door to self-directed learning systems, so users can acquire procedural skills at their own pace and on their own time.

The overall objective of the SAVE project was to develop a prototype assessment framework that observes a learner operating within an instrumented VE, evaluates their performance, and provides helpful feedback to improve their skills (Greuel, Myers, et al., 2016). In contrast to intelligent tutoring tools that address purely “algorithmic” skills with a single or small number of acceptable responses, SAVE addresses more open-ended procedural skills that may have a range of acceptable solutions with significant variation among them. Assessment in SAVE is facilitated by content authoring tools that enable the efficient specification by SMEs of training exercises and solutions to those exercises.

SAVE supports two classes of end users: learners who are assessed while performing exercises in the VE, and content authors who are responsible for developing exercise materials. Figure 1 provides a brief summary of the SAVE architecture, which is organized around these two user types. The training exercise user interface (EUI) provides both a visualization of a learning task and the interactive mechanism for solving the problem (L1). The learner actions are logged in a semantically grounded trace (L2); the automated assessment module (L3) compares these actions to the solution model for the problem, and generates learner-friendly feedback (L4) for display in the EUI.

The content authoring tools are used by two different types of authors, working in collaboration. The semantic 3D model author uses an annotation tool to define semantic overlays (A1) to generate a characterization of 3D objects and their components. With a content assembly tool (CAT), an instructional content author then uses these annotated objects to compose a specific virtual context for a training exercise (A2). A key element of our approach to exercise authoring is that the author first defines the procedural structure of the solution by demonstrating how it should be done directly in the EUI (A3); a companion solution editor (A4) then enables the author to specify annotations to the solution that define allowed variations from the initial demonstration.

Figure 1: The SAVE framework supports both learners and content authors



Source: Greuel, Myers, et al., 2016

The initial prototype of the SAVE framework was developed in part to support preliminary evaluation and user studies (Myers & Gervasio, 2016). The framework is extensible to support the addition of, and reasoning about, dynamic state information linked to objects populating the semantic VE. Future plans include a provision for behavior specification within the 3D model authoring process, assistance tools that guide instructional content authors in solution editing process, and an improved core solution matching capability. Further, additional means are under consideration to build upon the basic capabilities of the SAVE prototype, including (1) the application of a design-for-assessment approach to provide an iterative experimental protocol for adapting evidence-centered design resources to the system and (2) multimodal analysis framework to provide data-based insights into learner strategies and behaviors (Greuel, Murray, & Yadav, 2016).

Action and Solution Models

Action models in SAVE are initial representations of potential actions and events, which provide the semantic basis for use in content authoring and learner performance assessment. Two-level action models allow for reasoning about user actions at both the level of individual steps in coordinate-space and aggregated actions in a semantically-grounded space. An abstraction mechanism converts basic EUI keyboard and mouse actions into higher-level, meaningful actions for assessment. A solution model in SAVE is composed of one or more generalized action traces, each consisting of a sequence of steps and annotations that specify the allowed variations. A step is a parameterized action, a class of actions, or a set of options, each of which is composed of a partially ordered set of steps. Annotations defined over steps include action ordering and grouping constraints; annotations defined over parameters include parameter type, value, and equality constraints; and annotations defined over state capture requirements on the application state or on object properties that cannot be determined from actions themselves. Solution models thus implicitly define sets of specific solution instances, each of which is considered a valid solution.

Automated Assessment and Pattern Matching

The automated assessment module in SAVE determines a mapping from a learner response to the predefined exercise solution model. This alignment problem is formulated as approximate graph matching, using graph edit distance to rate the quality of the mappings. Graph edit distance measures the cumulative cost of graph editing operations (e.g., deletions, insertions) needed to transform the learner response into an instance consistent with the solution model. The intuition is that the lowest-cost alignment corresponds to the specific solution instance the learner is most likely attempting.

To perform assessment through approximate graph matching in SAVE, the solution model is represented as one or more graphs, each representing a family of possible solutions. Within the graphs, actions and their parameters are nodes, parameter roles are links, and conditions required by the solution are constraints on one or more nodes. The learner response is represented similarly as a response graph. Alignment involves finding the lowest-cost mapping between the response and a solution graph, with costs incurred for missing mappings and violated constraints.

The SAVE approach provides an automated assessment of a learner’s performance, providing contextual feedback to help improve their skills and enhance their understanding. In contrast to intelligent tutoring tools that assess “algorithmic” skills with single acceptable responses, SAVE addresses open-ended procedural skills with a range of acceptable solutions. The automated assessment is facilitated by content authoring tools that enable instructors to specify training exercises and solutions to those exercises.

Technical Training using Augmented Reality Mentoring

Technical training is often needed to conduct safe and efficient repairs and maintenance on complex mechanical equipment. However, such training can be very expensive, since it typically consists of one-on-one demonstrations and painstaking trial-and-error review of dense technical manuals. The AR-Mentor system offers a more independent and selfpaced approach to such hands-on training (Kumar et al., 2014). The system uses two types of technology: see-through AR overlays, which permit learning while gazing directly at the work space rather than glancing back at instructions, and automated interactive dialog, which enables the learner to pace their instruction as needed. Instruction is delivered through a head-mounted display with an earphone and microphone.

With AR-Mentor, the learner receives on-demand voice instruction and views several types of visual overlays: annotated technical diagrams of complex equipment and tools, 3D animations to demonstrate how to manipulate tools and components, animated icons that direct the learner's gaze, and live-action videos of mechanics conducting procedures. The learner can ask questions, skip steps, or request steps to be repeated.

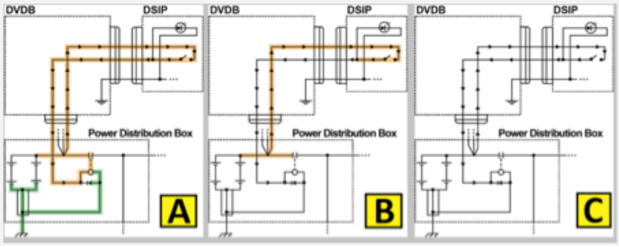
A research team tested the AR-Mentor prototype in a technician instruction context during a 12-week, 512-hour, and 80-lesson maintenance training session for ground transportation vehicles. The prototype was tested with two of these lessons, one focused on learning detailed procedural content (*Lesson A*) and one focused on alternate troubleshooting procedures that involved decision making based on logic and reasoning (*Lesson B*). Lesson A involved 33 steps that took an experienced technician about 40 minutes to perform. The procedure required learning how to use a tool called a *bubble level* that contains multiple levels, each placed at subtly different angles. In addition, the procedure required the technician to move to four different locations on the vehicle. For the Lesson B, trainees were learning to troubleshoot electrical circuits — a form of logical-deductive reasoning based on locating the cause of the fault through systematic testing. To support this more complex form of learning, AR-Mentor blended step-by-step illustrated and verbal instructions with embedded formative assessments of *next-step reasoning*. AR-Mentor displayed schematic diagrams that were annotated to represent the state of the circuit (on, off) and to point out where the mechanics should perform diagnostic tests of circuits (e.g., using a multimeter). In the training setting, instructors taught by programming a training system to simulate four different electrical continuity bugs in the vehicle's power distribution system. For the purposes of the study, three bugs were used for training and one bug for performance assessment. System performance assessment involved administering assessments of learning and observers coding the quality of the learning process (e.g., errors, help-seeking, time to complete) in two contrasting training conditions: (1) pairs of trainees using AR-Mentor and, (2) guidance provided by an individual instructor (see Results of Learning below).

Learning Model

According to learning science theory, AR-Mentor may not only make learning easier because it reduces the distractions associated with shifting eye gaze from instructions to the work space, it also supports self-directed learning by delivering three types of learning modalities in parallel: (1) hands-on practice, (2) visual demonstration, and (3) verbal explanation. This presentation activates three parallel neurological processing tracks: the temporal-spatial, visual, and logical-verbal. According to parallel processing theory, when multiple neurological systems are activated, transfer of knowledge to long-term memory is enhanced (Mayer & Moreno, 2002; Paivio, 1971). In addition, AR-Mentor was observed to enhance two social learning processes: mentoring and peer collaboration. In observations of the AR-Mentor in context, trainees asked fewer questions of instructors, and instructors did not need to repeat the information as frequently. Higher degrees of learner confidence and agency were also observed and confirmed in interviews.

In practice, the system provided two forms of tutoring: First, as an *automated technical manual* that walked a mechanic through step-by-step procedures with interactive voice instructions (e.g., the learner states “Computer, done” after a step is completed and requests “Computer, next step” when ready to move on) and instructive graphics with text projected over the work environment. This step-by-step approach was developed and refined over two rounds of pilot testing. Second, the system functioned as a *coaching tool* that taught troubleshooting skills through a mix of step-by-step dialog and specially created, embedded formative assessments. These assessments prompted the trainee to reflect on next-step reasoning in the process before the system provided direct instruction on that step (see example in Figure 2). The coaching tool approach was piloted once.

Figure 2: Sample next step reasoning assessment

Audio	To figure out where to start checking the circuit, you need to start closest to the component where you have power. Remember when you flipped the master power switch and heard the click? Think about what component is getting power based on that click. Select the correct schematic representing the status of the circuit.
Graphics	Schematic A? Schematic B? Schematic C? 

Source: Yarnall et al., 2015

Content Creation

Content creation for the procedures required hundreds of hours to interview expert instructors, record video of the correct steps, program libraries of animation sequences and voice dialog scripts, and calibrate the 3D arrows and graphic overlays to the gaze of the learner. This process represents a bottleneck that limits the applicability of such technology in work contexts where mechanics need to learn several dozen procedures and are required to update knowledge flexibly as equipment changes. To accelerate the process, the AR-Mentor team developed a system concept to permit instructors and SMEs to create lessons by taking demonstration pictures and video recordings, entering them into a reusable library, and using a graphical editor to build a lesson.

For generation of automated dialog instructions, a method was specified for translating the text of a technical manual into automated voice dialog. This involved parsing the text to identify sections associated with schemas they established for work tasks and step-by-step dialog. The work task schema had five components: (1) determining a specific "work package" in a technical manual, (2) checking tools and parts relevant to that work package, (3) establishing the equipment pre-conditions, (4) providing relevant safety warnings, and (5) delivering the procedural steps. The dialog schema focused on five "primitives" of interactive self-learning dialog: (1) going to a specific step, (2) repeating a step, (3) obtaining the location of a component, (4) obtaining the definition of a component, and (5) obtaining an explanation of a step.

Results of Learning

The initial round of testing of the automated technical manual approach revealed basic efficacy, but identified learning problems around the complex bubble level procedures. Analysis linked the problem to an overly dense presentation of visual and dialog information of the bubble level content. These representations were simplified and delivered at a slower pace. The second round of testing showed comparable results in the learning process between the AR-Mentor and instructor conditions (e.g., errors, help-seeking) with much reduced burden on the instructor (see Table 1). In addition, the simplification of bubble level information led to significantly fewer errors than Round 1 on those steps (Round 1 M = 2.5 errors; Round 2 M = 0.63 errors, $t(13) = 2.5$, $p < .03$) (small sample analysis supported by de Winter, 2013). A test was developed and validated of procedural knowledge, and comparable results were obtained between the AR-Mentor technology and traditional instruction (Yarnall et al., 2015).

Table 1: Comparison of novice error, help-seeking, instructor guidance, and time per learning conditions

Learning Condition	Total Errors Mean	Total Help-Seeking Mean	Total Instructor Guidance Mean	Mean Total Time: hh:mm:ss
AR-Mentor (n = 8)*	1.75	5.63	1.75	1:23:00
Instructor+Manual (*n* = 7)	2.00	5.86	14.71	1:11:00

Note that *n* represents learners completing all 33 steps by hand; not all did due to testing time constraints.
Source: Yarnall et al., 2015

In the case of alternate troubleshooting, the one round of testing indicated that the learning process involved roughly comparable amounts of time (14 minutes for instructor condition; 19 minutes for AR-Mentor), relatively low error rates (0.63 per bug on average for the instructor condition; 1.75 per bug on average for the AR-Mentor condition), but with 20 times as much instructor guidance required in the instructor condition than the AR-Mentor condition. Learners expressed positive perceptions of the AR-Mentor system. The final learning assessments comparing the two study conditions were discounted because of lack of baseline equivalence between them. The trainees in the instructor condition had received a full day of training in advance of the observational study and the AR-Mentor trainees had received no training. However, to provide an indication of AR-Mentor's efficacy, with an average of just 19 minutes of instruction, the 4 AR-Mentor learners averaged 44% correct in the performance assessment, and displayed adequate recollection of the procedures for using tools and recognition of components. By comparison, the two learners who had one day of training prior to participating in the 14-minute practice sessions averaged 100% correct on the performance assessment.

Future enhancements for AR-Mentor are already being developed, such as enabling the detection of uncertainty in learner's' voice responses and using learners' feedback about the quality of each online lesson to inform lesson updates and improvements.

Simulating Manufacturing and Prototyping with Learning Environments

The SiMPLE system uses a scalable massively-open online course (MOOC) interface and includes Gazebo, a 3D robotics simulation software package that supports rapid prototyping and iterative model enhancements (Koenig & Howard, 2004). The learning goals include developing proficiency using 3D simulation and other technologies (e.g. 3D printers, laser cutters, and robot kits) while promoting understanding of basic core systems. SiMPLE includes a series of five progressive, interactive online modules and design tasks, designed to teach students how to troubleshoot, adapt, and modify complex systems to meet new demands (see Table 2).

Table 2: SiMPLE course module description

Module Title	Course Description
SIM 101: Introduction to Simulation	Explains the elements of SiMPLE and how simulation fits into the overall engineering design process
SIM 102: Introduction to Modeling	Explores more features of Gazebo and review basic electrical concepts, as well as the construction of a simple electric circuit
SIM 103: Introduction to System Design	Provides both mechanical simulations and physical experiments related to the impact of weight distribution on a simple vehicle
SIM 104: Design Challenges	Learners employ acquired knowledge and skills in engaging competitions designed to exemplify the use of both simulation and physical prototyping
SIM 201: Customizing a Model	Explains how Gazebo interfaces with other tools such as laser cutters to provide a broad range of flexibility for developing physical models.

Source: SRI International, 2016

A key feature of the SiMPLE course materials is the use of multiple representations to accelerate learning (Figure 3). These representations include: a 3D world view to enable visualization of model dynamics and interactions within the simulated world environment, a schematic view that allows for easy comparisons between disparate systems, a model editor view that shows the kinematics of the model, and a physical representation that is created using a robot kit. A graphing utility tool provides visual representations to enhance learner diagnosis of design flaws by plotting simulation properties over time; this tool allows users to quickly optimize simulations and make quantitative comparisons.

Gazebo enables learners to collaborate and iterate designs using a novel Component Modeling Language that allows users to create system-level functional models through an intuitive drag-and-drop graphical user interface, and to iterate designs by testing them in a 3D simulation environment. The model editor allows learners to build and simulate their models. Using this tool, learners and field personnel can explore how components function, test their designs, and modify complex systems. Once the design has gone through several iterations within Gazebo simulation environment, the user is then able to export data files that are compatible with the other prototyping technologies (e.g. software for laser cutters and 3D printers).

The MOOC course materials and software include features designed to support learning of physical electromechanical systems through a learning companion tool that provides guidance based on user performance. Embedded formative assessments or “knowledge checks” that provide auto-generated feedback are included in two of the five modules.

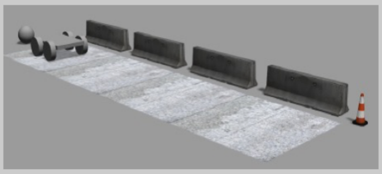
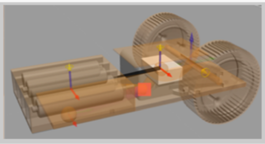
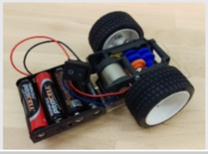
A small pilot study with adults from a variety of backgrounds (i.e. four practitioners, one undergraduate student, and two university faculty) showed evidence of the potential for advancing the intended learning goals. Observations revealed a high degree of engagement and collaboration among participants. All participants successfully completed the coursework, embedded assessments, and design challenges. Results from a pre- and post-assessment of thirteen items related to course content revealed improvements in learner performance, after completion of the course (see Figure 4).

Future efforts that will advance the work of the SiMPLE project include leveraging the Gazebo back-end to collect additional data on user behaviors; enhancing the learning companion tool; integrating with a cloud-based modeling and simulation environment to enable rapid deployment; and developing additional open-ended advanced courses that can extend learning in many professional settings.

Conclusions

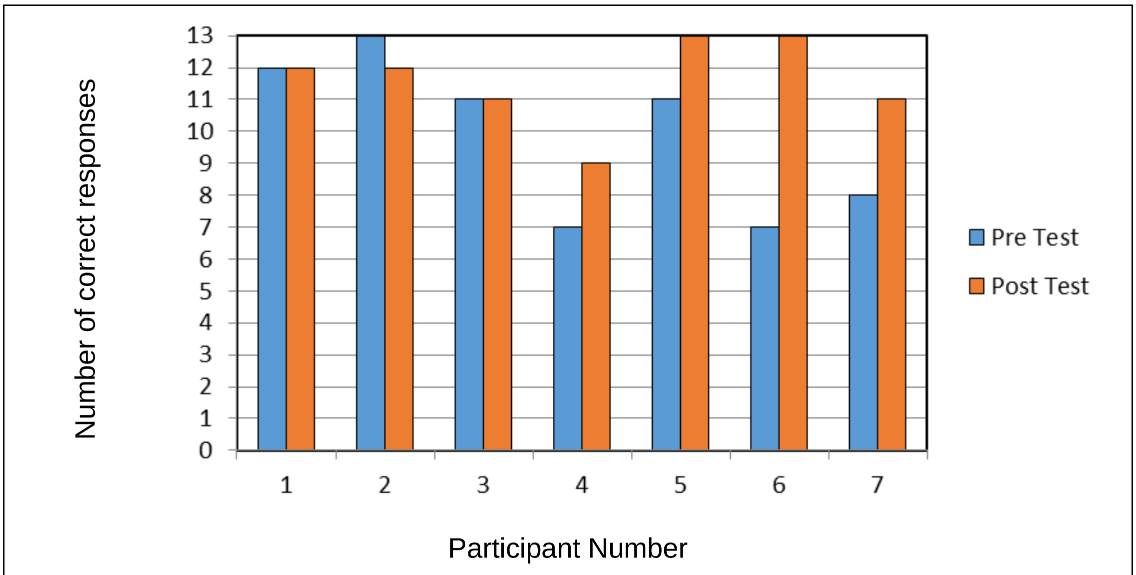
In this paper, we discussed several applications of VE technology to help improve self-directed learning of manual procedures in technical occupations. We believe that these early-stage applications offer valuable insights that can be applied broadly to hands-on training systems in numerous technical and analytic disciplines. We focused in particular on innovative techniques for automated content creation, intelligent coaching, and learner performance assessment. There are promising opportunities for introducing these techniques in any profession where complex simulations and modeling tools are used. Examples of potential domains include economic policy analytics, electromechanical systems engineering, medical emergency response training, or clean energy workforce development.

Figure 3: SiMPLE links multiple representations for fast learning

Representation Name	Image	Purpose
3D World View		• Enable visualization of model dynamics and interactions with the simulated world environment • Run the simulation in real time
Schematic View		• Present an abstract graph view of the links, components, and joint types • Allow for decomposition into subsystems • Allow for easy comparison between disparate systems
Model Editor View		• Show the kinematics of the model • Allow editing of link and joint properties
Physical System		• Experiment with the physical parts to test adequacy of simulated system • Develop hands-on manipulation skills
Graphing Utility		• Track variables over time • Optimize performance • Discover complex relationships

Source: SRI International, 2016

Figure 4: Assessment results of SiMPLE course content



Source: SRI International, 2016

Acknowledgements

This material is based upon work supported by the United States Government under Contract Nos. W911QY-14-C-0023, W91WAW-12-C-0063, and W911NF-14-2-0062. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the U.S. Government.

The authors would like to thank the following for their contributions to the works described in this paper: Karen Myers, Melinda Gervasio, Grit Denker, Daniel Elenius, Chris Jones, Scott Krehbiel, John Pywtorak, Rukman Senanayake, Michael Wessel, Rakesh Kumar, Supun Samarasekera, Girish Acharya, Glenn Murray, Michael Wolverton, Zhiwei Zhu, Vlad Branzoi, Nicholas Vitovitch, Jim Carpenter, Sara Vasquez, Maria-IsabelCarnasciali, Srujal Patel, Anna Werner, Ian Chen, Karen Koenig, Nate Koenig and the Open Source Robotics Foundation.

References

DE WINTER, J. C. (2013) Using the Student's t-test with extremely small sample sizes. *Practical Assessment, Research & Evaluation*, 18(10), 1-12.

GREUEL, C., MURRAY, J., & YADAV, M. (2016) Semantically Enabled Virtual Environments for Training Assessment. *In Proc. of 6th European Immersive Education Summit (EiED 2016)*. Padua, Italy.

GREUEL, C, MYERS, K, DENKER, G. & GERVASIO, M. (2016) Assessment and Content Authoring in Semantic Virtual Environments. *In Proc. of Annual Meeting of the Interservice/Industry Training, Simulation and Education Conference (I/ITSEC 2016)*. Orlando, FL. (to appear)

KOENIG, N. & HOWARD, A. (2004) Design and Use Paradigms for Gazebo, an Open-Source Multi-Robot Simulator. *In Proc. of 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2149-2154.

KUMAR, R., SAMARASEKERA, S., ACHARYA, G., YARNALL, L., et al. (2014) Augmented reality virtual personal assistant for training, maintenance, and repair. *In Proc. of Annual Meeting of the Interservice/Industry Training, Simulation and Education Conference (I/ITSEC 2014)*. Orlando, FL.

MAYER, R. E., & MORENO, R. (2002) Aids to computer-based multimedia learning. Learning and instruction, 12(1), 107-119.

MYERS K., & GERVASIO, M. (2016) Solution authoring via demonstration and annotation. *In Proc. of 16th International Conference on Advanced Learning Technologies (ICALT 2016)*. Austin, TX.

PAIVIO, A. (1971) Imagery and language. In S. J. Segal (ed.). *Imagery: Current Cognitive Approaches*. New York, NY: Academic Press.

SRI INTERNATIONAL. (2016) *Final Presentation: SiMPLE*. Defense Advanced Research Projects Agency, Manufacturing Experimentation and Outreach Two (MENTOR2) Principle Investigator meeting, Washington, DC. July 14.

YARNALL, L., VASQUEZ, S., WERNER, A., KUMAR, R., et al. (2015) Human performance in content design for interactive augmented reality systems. *In Proc. of Annual Meeting of the Interservice/Industry Training, Simulation and Education Conference (I/ITSEC 2015)*. Orlando, FL.