Behavior Authoring and Run-time Management of Computer Agents for a Virtual Operating Room Training Environment

Yiannis E. Papelis, Ph.D.
Menion Croll, Hector Garcia
ypapelis@odu.edu, mcroll@odu.edu, hgarcia@odu.edu
Virginia Modeling Analysis & Simulation Center

Mark W. Scerbo, Ph.D.
Rebecca Kennedy
mscerbo@odu.edu, Becca.kennedy92@gmail.com
Dept. of Psychology

Old Dominion University

ABSTRACT

Developing effective virtual environment scenarios that involve human participants is a challenging task because of the difficulty associated with anticipating and responding to all possible reactions of the human participant as the scenario unfolds. Further complicating the task in immersive simulations is the need to accommodate multiple interaction modalities that go beyond direct keyboard and mouse input, examples of which include gestures, use of domain-specific props, and voice recognition. In this paper, we present an approach to modeling an immersive virtual environment aimed at training surgical procedures, the Virtual Operating Room (VOR). In the VOR, trainees interact with a surgical team comprised of real and virtual team members in a standard OR, incorporating real and virtual equipment. Scenarios in the VOR are described using a concurrent state machine methodology that supports non-linear scenario specifications with manageable complexity, even for heavily multi-branch scenarios. The main execution engine utilizes a flexible architecture that allows integration of external control signals that can affect scenario evolution. The paper describes the architecture and provides an example of a scenario addressing laparoscopic cholecystectomy moderated in real time through user voice recognition, instrument manipulation and hardware-based performance assessment. The voice recognition system utilizes a semantic interpretation grammar that allows detection of semantic responses even when spoken using different sentence patterns. A realistic physical simulation of an instrumented abdominal cavity is used to measure task-related performance. The voice recognition system and instrumented abdomen inform the virtual characters who can provide task-specific and summative feedback to the trainee.

ABOUT THE AUTHORS

Yiannis Papelis is a Research Professor at the Virginia Modeling, Analysis & Simulation Center at ODU. His research interests include autonomous unmanned systems, NextGen, semi-autonomous behavior modeling, and immersive virtual reality. Dr. Papelis received a BS degree from Southern Illinois University, an MS degree from Purdue, and a Ph.D. degree from University of Iowa, all in electrical & computer engineering

Mark W. Scerbo, Ph.D., is Professor of Human Factors Psychology at Old Dominion University and an Adjunct Professor of Health Professions at Eastern Virginia Medical School. His research addresses human performance assessment, user interaction with medical simulation technology, and the development of new medical simulation models and technology.

Rebecca Kennedy is a doctoral student of Human Factors Psychology at Old Dominion University. Her research interests include user interaction with medical simulation technology and virtual environments as well as improving the usability of products and interfaces.

Menion Croll is a VMASC senior project scientist. His research interests include visualization and serious gaming. He received a BS from Virginia Tech, and an MS from Old Dominion University, both in Computer Science.

Hector Garcia is a VMASC senior project scientist. Areas of expertise include Visualization, Virtual Environments and Virtual Reality, integrating state of the art visualization systems with modeling and simulation applications. He has developed simulations for medical training such as the Virtual Operating Room, the Virtual Pathology Stethoscope and the Wound Debridement Simulator.

Behavior Authoring and Run-time Management of Computer Agents for a Virtual Operating Room Training Environment

Yiannis E. Papelis, Ph.D.
Menion Croll, Hector Garcia
ypapelis@odu.edu, mcroll@odu.edu,
Virginia Modeling Analysis & Simulation Center

Mark W. Scerbo, Ph.D
Rebecca Kennedy
mscerbo@odu.edu, Becca.kennedy92@gmail.com
Dept. of Psychology

Old Dominion University

INTRODUCTION

The term scenario as used in immersive training applications typically refers to a sequence of events designed to test a trainee's knowledge, skill and readiness in a given situation. Training scenarios operationalized within a computer-generated virtual environment are widely used in medical applications and can involve one or more trainees and one or more virtual human entities that fill the required roles, augmented with a variety of actual and/or simulated instruments and related elements. In situations where the training task involves use of a physical device, the interaction is simplified as an actual or simulated device can be instrumented – an example is a flight simulator used to train pilots where the trainee interacts with physical devices captured in an instrumented cockpit. Difficulty increases when the trainee actions cannot be captured as easily or when verbal communication is an integral part of the task. An example of the latter is medical applications in which the trainee must communicate verbally with other scenario participants. In such cases, the scenario need not only include the list of events but also a list of expected verbal interactions among the scenario participants.

One of the challenges when developing such immersive simulations is that the traditional definition of a scenario does not capture all the elements required to develop a computer-based testing system. A typical scenario specification includes the description of a series of events and the expected behavior of computer generated actors as well as the role of the trainee. The events are linear, although non-linear variations can also be described albeit as pre-scripted options that get exercised at key event points. Effectively, the notion of a training scenario closely resembles the meaning of the word, scenario, as used to choreograph/block theater or movie productions. Actors are guided along a series of events that appear interactive and even chaotic, yet are precisely planned. The problem with this approach is that it does not match at all the reality of a trainee participating in a scenario whose purpose is to assess their skill or knowledge. The core issue is that the person being tested will not necessarily behave in a specific way – to the contrary, the trainees are not following a set script, instead they are following a prescription for training and must improvise as best they can to situations that unfold in the virtual environment. This problem is exacerbated with novices, who are more likely than experts to behave in unexpected ways. Once a trainee deviates from the original scenario, the responses of the remaining actors must also adapt; at the end, it is unlikely that the final sequence of events will be exactly what was planned in the original scenario – and unlike a movie set in which the director can command a re-take, a failed scenario in a training environment is costly and ultimately does not achieve its goal.

The traditional approach to addressing this problem is for the scenario developer to enumerate all possible branching points and create sub-scenarios for each one – the system monitors the scenario evolution and depending on the response of the trainee, different sub-scenarios get activated at each branching point. The responses can be verbal, invoking triggers based on voice recognition, or physical, invoking triggers based on hardware sensing. This approach can work but as the sophistication of a training situation increases, key shortcomings become a limiting factor. When scenarios have multiple correct outcomes or multiple ways to reach the same endpoint, the complexity of a scenario increases exponentially with the number of allowable interactions. Furthermore, when utilizing voice recognition as a means to advance a scenario, variations of how individuals can express the same concept prevent word-for-word recognition as a viable approach to sequencing a scenario.

In this paper we present solutions to the aforementioned problems, as applied to an immersive medical training application. We address the explosion of the specification complexity by utilizing an alternative formalism for expressing a scenario that allows the designer to express multiple alternatives in a compact and easy to visualize manner. The methodology utilizes a hierarchy to group larger scenarios into modules, and concurrency to allow independent expression of multiple scenario streams that may or may not affect each other. We enhanced existing techniques for recognizing sentences based on their semantic meaning as opposed to rigidly described word structure to capture variability in expressing similar concepts and allow such verbal utterances to be used as triggers for affecting scenario evolution. This technology has been integrated into a virtual environment supporting scenario execution and evaluation in single (and in the future multi-) trainee applications in the context of a Virtual Operating Room (VOR).

This paper describes the VOR and the underlying techniques for addressing two shortcomings of existing systems, namely drastic increase in scenario complexity for capturing highly variable scenarios and use of sentence recognition to allow natural interaction between trainees and the virtual environment.

The VIRTUAL OPERATING ROOM

The Virtual Operating Room (VOR) was developed to provide surgical teams with a fully immersive virtual environment to train surgical teams the way they operate (Baydogan, Belfore, Scerbo, & Saurav, 2009; Scerbo et al., 2006, 2007). The VOR was designed, in part, to provide a rich environment for studying contextual factors that contribute to errors, such as inappropriate equipment design, poor judgment and decision-making among, poor team interactions, or the potential effects of changes in organizational policies Moray (1994).
The VOR is outfitted with both real and virtual equipment, integrates other medical simulators, and allows trainees to interact with a surgical team comprised of real and/or virtual team members. The team members are comprised of students, residents, and virtual agents. It enables problem-solving scenarios to be created and executed both on a technical and social level. Within the operating room, it is critical that members of the surgical team work well together in order to provide the safest and most efficient care for the patient. The ability of individuals to learn how to work with people from different professional specialties, cultures, levels of experience, and personalities is a challenging task for which there is little formal training. Figure 1 illustrates a rendering of the VOR's environment.

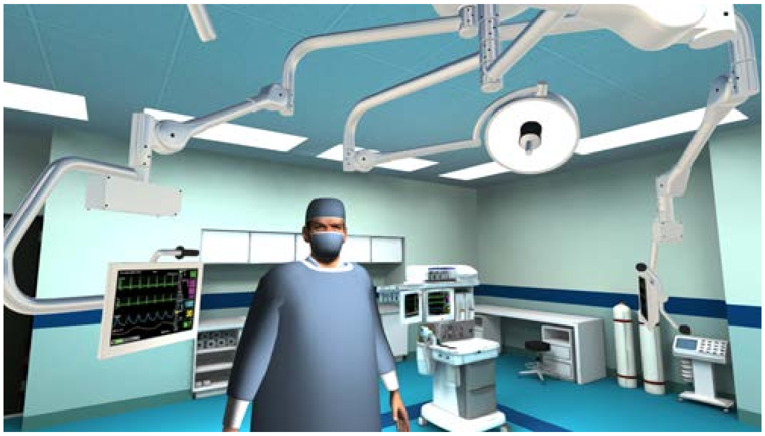

Figure 1 – Screenshot of Virtual Operating Room Environment

Scenario Specification Methodology

The scenario specification approach is based on traditional state machines but with several extensions designed to support non-linear traversal of expected interactions between the participant and the system. A scenario is comprised by states, scenes, transitions, links, and activity/predicate functions.

State

A state represents the information associated with the scenario at a given point in time. States contain three pieces of program code (functions): the entry function, the exit function and the current function. In addition, states are associated with incoming and outgoing transitions. The execution semantics of states are straightforward. A state is activated when any of its incoming transitions triggers. When a state is activated, its entry function executes. As long as the state remains active, its current function executes. While a state is active, when any of the outgoing transitions triggers, the state is deactivated. When a state is deactivated, its exit function executes.

Transitions

A transition is used to link objects; these objects can be scenes or states. Each transition has an origin and destination object, and a predicate function. When the origin of the transition is an active object, then the transitions' predicate function executes and if it returns true, then the source object is deactivated and destination object is activated.

Scene

A scene extends to the notion of a state by adding hierarchy and additional semantics that simplify the authoring of non-linear scenarios. Similar to a state, a scene has an entry, exit and current function, but in addition, a scene can contain other states, and has entry and exit ports and links that connect its ports to the internal states. A scene activates when any of its input ports activate. A scene deactivates when any of its output ports activate. In addition to this structure, a scene is associated with performance metrics that are calculated while it the scene is executing.
A scene is represented by a blue box, with slightly curved corners. Input ports are depicted at the top of the scene and exit ports are depicted at the bottom of the scene. States are depicted as green circles and links are depicted in dashed red. An example is shown in Figure 2.

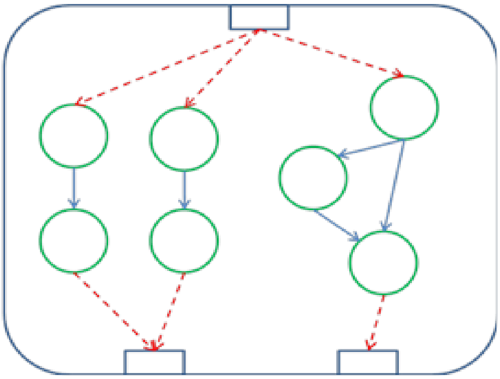

Figure 2 - Scene Illustration.

A scene becomes active when any of its input ports become active. Once a port is active, links designate which states internal to the scene will gain control. Similarly, links between states and the output ports dictate how execution of internal states affects the completion of the scene.

Input links (Forks)

Input links (or forks) connect entry ports to internal states. Unlike transitions that have a predicate function to select which transition fires, all fork links originating at an input fork activate simultaneously, effectively creating multiple active paths within the internal states. In the example from Figure 2, there are 3 sets of disjoint state machines in the scene and upon activation of the input port, they will all become active. Fork transitions may only be used with disjoint sub state machines. This prevents states from being activated multiple times.

Output Links (Joins)

Output links (or Joins) connect internal states to exit ports. The scene terminates when any of its exit ports activate, but for an exit port to activate, all output links must have states that are active. Effectively, Joins merge parallel execution streams that were initiated with Forks.

Activity and Predicate Functions

States, scenes and transitions depend on a variety of functions in order to operationalize a scenario. In order to facilitate high level scenario authoring and avoid low level programming, a set of activities has been defined that allows the user to create activity and predicate functions without having to develop code while authoring the scenario. Instead, the user can simply select activities among a pre-specified set of available functions; if desired however, a user can indeed develop more complex specifications to be associated with the execution of a state. Examples of available activities include:

- Animation: invoke an animation for any of the virtual actors
- Scoring: update the performance score of a trainee
- SpeechOut: generate speech
- SpeechReco: recognize a sentence
- CommOut: send a message to external hardware
- CommIn: receive message from external hardware

The CommOut and CommIn actions are designed to streamline communication with external devices that can either provide feedback to the trainee (i.e., a physical instrument can be made to show a specific indication) or capture trainee actions (i.e., an external device can detect if the trainee ligated the wrong vein.

Sentence Recognition

The trainees communicate with the virtual team members using wireless headset microphones. The speech recognition application was created using the Dragon Naturally Speaking Client SDK version 12.5. Voice recognition is intended to be a natural interaction method, but relying solely on system recognition of key words might limit the flexibility in detecting varying user utterances. To support more natural trainee interactions, the voice recognition system uses a semantic interpretation of spoken utterances guided by key words. The semantic interpretation grammar enables the system to detect the meaning of trainee responses even if the wording and sentence structure of the utterances vary (Jay, 2002). To implement semantic interpretation, we build on the Speech Recognition Grammar Specification (W3C 2004), a standard published by the World Wide Web Consortium. A grammar specified in SRGS can be converted into the well-known Finite State Automata (FSA) formalism, which in turn can be exercised to determine if a sentence matches a rule in the grammar. In an FSA, a series of states is linked with transitions – each transition is traversed when the specific word is recognized. Flexibility is increased by allowing a transition to be a "null" which allows transition without any recognition. For example, consider the FSA shown in Figure 3, which is meant to recognize a greeting. The following are a subset of the sentences that can be recognized by the grammar:

Hi there Bill
Hi
Hello
Hello Bill
Hi Bill

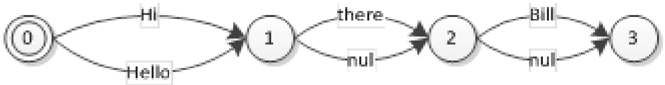

Figure 3 – Example of Sentence Specification

The formalism was extended by adding an arc of type ANY, meaning it will match any word, as opposed to a NULL arc that does not need a word to transition. The following shows an example which uses ANY.

In this example, after reaching state (2), the sentence can end right away (2→4 transition) or after addition of any word, or after addition of any two words. The following are a subset of the sentences that can be recognized by the grammar:

Hello
Hi there John
Hello dear
Hello dear Jane

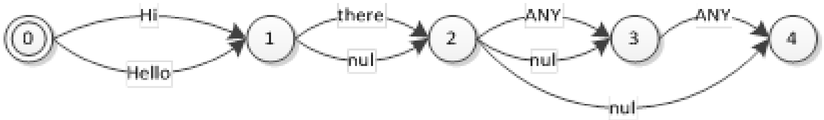

Figure 4 – Example of Sentence Specification Using 'ANY' arcs

To create the actual semantic interpretation grammar, we first created a list of possible trainee utterances. A hierarchical task analysis for a laparoscopic cholecystectomy was used to determine which statements are expected to be uttered in an optimal procedure. To identify utterances that were previously unexpected or unrecognized by the VOR, video recordings of surgical residents interacting with the VOR were viewed, transcribed, and analyzed.

For every identified trainee speech event, statements were classified according to four semantic categories: declarative, imperative direct, imperative indirect, and interrogative. Groups of sentences in these four categories describe: 1) communicating status information, 2) commands, 3) commands phrased as questions, and 4) asking about the status of information. Individual speech events were then further classified in terms of a specific semantic event (e.g., asking for a trocar, asking for vitals) and each event was then categorized according to the surgical team member responsible for responding to the trainee's statement, question, or request.

Several sentences structures and wording variations were then identified and diagrammed for each semantic event. For these diagrams, key words were identified as utterance components like actions, items, confirmation of steps, patient status information, or problem diagnosis information which corresponds to semantic categories. In this way, different sentence structures (e.g., "Please hand me a trocar," "Can I have a trocar?") specified by key words (e.g., "trocar") are labeled as implying the same semantic interpretation.

The Surgical Simulator Hardware

The cholecystectomy procedure is performed on a LapTrainer system by Simulab, Inc. (Seattle, WA). This is a commercial simulator that uses plastic and rubber models of the abdominal cavity including the stomach, liver, and gall bladder. A replaceable gall bladder is attached to a retracted infundibulum with Velcro and can be ligated with genuine laparoscopic instruments.

A variety of custom feedback devices have been developed to provide awareness of trainee actions during the procedures. The gall bladder has been instrumented with hair-thin invisible conductors that can sense when the proper ligature has been applied. A motion capture device is also installed inside the body cavity in order to recognize when the trainee is inactive or has moved away from the area of interest.

Surgical Procedure

At present, the VOR scenarios address a fundamental surgical procedure: laparoscopic cholecystectomy (gall bladder removal). Cognitive task analyses were used to establish the roles, responsibilities, and activities for each member of the surgical team: attending surgeon, operating surgeon, anesthetist, scrub technician, and circulating nurse. (In the present scenario, virtual agents were created for only the attending surgeon, anesthetist, and circulating nurse.) A timeline was created to delineate the activities for each step of the procedure for each team member. The initial description represented the procedure performed under ideal conditions. Next, critical complications were identified and weighted according to human error identification methods (Stanton et al. 1986). This workflow description was used to create a framework for coordinating movements and dialogue among the virtual agents.

CONCLUSION

Developing effective virtual environment scenarios that involve human participants is difficult because humans are not predictable. Consequently, it is necessary to anticipate and be prepared to respond to many possible behaviors as a scenario unfolds. In this paper, we described our approach to overcoming these challenges. Scenarios are described using a concurrent state machine methodology that supports non-linear scenario specifications. The main execution engine utilizes a flexible architecture that allows integration of external control signals that can affect scenario evolution. The scenario contains scenes that can include one or more states with inputs and outputs to the scene mediated through dedicated ports. Transitions and links enable connections between/within states or scenes. In addition, activity and predicate functions manage actions within active states and govern transitions.

We applied this modeling approach to the VOR, an immersive virtual environment aimed at training surgical teams and procedures. A scenario was developed for laparoscopic cholecystectomy moderated in real time through user voice recognition, instrument manipulation and hardware-based performance assessment. Because communication is a fundamental component of surgical team performance, special emphasis was placed on the voice recognition portion of the system. A semantic interpretation grammar was utilized to allow detection of utterances spoken using different sentence patterns. In addition, the simulated body cavity was instrumented with sensors to recognize actions and measure task-related performance. We believe this approach will enable the development and execution of more sophisticated and complex scenarios and ultimately provide a much richer training experience for users of the VOR

ACKNOWLEDGEMENTS

The authors would like to acknowledge the National Center for Collaboration in Medical Modeling and Simulation, and the Virginia Modeling Analysis & Simulation Center for supporting the project.

REFERENCES

Baydogan, E., Belfore, L. A., Scerbo, M.W., & Saurav, M. (2009). Virtual operating room team training via computer-based agents. International Journal of Intelligent Control and Systems, 14, 115-122.

Jay, T.B. (2002). The psychology of language. New York: Pearson.

Moray, N. (1994). Error reduction as a systems problem. In M. S. Bogner (Ed.), Human error in medicine (pp. 67-91). Hillsdale, NJ: Erlbaum.

Scerbo, M.W., Belfore, L. A., Garcia, H. M., & Weireter, L. J., Jackson, M., Nalu, A., & Baydogan, E. (2006). The virtual operating room. Proceedings of the Interservice/Industry Training, Simulation and Education Conference, Paper No. 2711, (pp. 1-9). Arlington, VA: National Training and Simulation Association.

Scerbo, M.W., Belfore, L.A., Garcia, H.M., Weireter, L.J., Jackson, M.W., Nalu, A., Baydogan, E., Bliss, J.P., & Seevinck, J. (2007). A Virtual operating room for context-relevant training. Proceedings of the Human Factors & Ergonomics Society 51st Annual Meeting (pp. 507-511). Santa Monica, CA: Human Factors & Ergonomics Society.

Stanton, N.A., Salmon, P.M., Walker, G.H., Baber, C., & Jenkins, D. P. (2005). Human factors methods: A practical guide for engineering and design. Burlington, VT: Ashgate.

W3C 2004, Speech Recognition Grammer specification Version 1.0, Retrieved March 2014 from:
www.w3.org/TR/speech-grammar.