From Simulations to Automated Tutoring

Dr. Sowmya RAMACHANDRAN[a] and Dr. Barbara SORENSEN[b]
[a]Stottler Henke Associates, Inc, San Mateo, CA, USA
[b]AFRL/HEA, Mesa, AZ, USA

Abstract: Training medical personnel to maintain readiness for medical emergencies and combat-related operations is a critical problem. Distance learning solutions are required for enabling effective training while minimizing time away from the important on-the-job duties of providing quality medical care. Simulation-based training can significantly benefit learners by providing opportunities for hands on training. A simulation by itself, however, is not sufficient to enable learning. It must be accompanied by opportunities for reflection and a chance for learners to try their skills under different conditions. This means a simulation-based training course should include several scenarios. The high cost of developing and administering training scenarios renders this infeasible.

We have developed a simulation-based training framework called SimCore that incorporates intelligent, automated assessment and coaching in support of self-paced learning. This reduces the need for human facilitators. A key feature of this framework is an authoring tool that supports rapid scenario development and customization and is designed for use by subject matter experts and course developers. This brings down the cost of scenario development. The system has been designed to interface easily with third-party simulators, with minimal effort. It also includes a Flash-based simulator that can be played on a web-browser.

A beta version of SimCore is currently being distributed for evaluation.

Keywords: Simulation-based training, Intelligent Tutoring, Scenario Authoring Tool

1. Introduction

Simulation-based training can significantly benefit learners by providing them the opportunity to practice and learn from applying their skills and knowledge to realistic training scenarios [1, 2]. Traditionally, simulation-based training has meant live simulations. The logistics and cost of such simulation constrain how often they can be presented. Typically such simulations deliver one or two scenarios. With computer-based simulations, it is possible to deliver cost-effective training with the added benefit that trainees can play them anywhere, anytime. In order to support such self-paced learning, simulations must be accompanied by performance assessment and feedback. As the healthcare community starts to invest significantly in simulation-based training, it is running against the limitation of requiring hands-on instructor-led facilitation to make it effective. This is prohibitively expensive. Automated coaching and feedback reduces the need for instructor facilitation, thus making simulation-based training cost-effective and feasible.

Report Documentation Page

Form Approved
OMB No. 0704-0188

1. REPORT DATE **JUL 2007**	2. REPORT TYPE	3. DATES COVERED **00-00-2007 to 00-00-2007**

4. TITLE AND SUBTITLE **From Simulations to Automated Tutoring**	5a. CONTRACT NUMBER 5b. GRANT NUMBER 5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER 5e. TASK NUMBER 5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **Stottler Henke Associates, Inc,1650 S Amphlett Blvd Ste 300,San Mateo,CA,94402**	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR’S ACRONYM(S) 11. SPONSOR/MONITOR’S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT

Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES

Human Interface and the Management of Information. Interacting in Information Environments Symposium on Human Interface 2007, Held as Part of HCI International 2007, Beijing, China, July 22-27, 2007, Part II. Volume 4558 of the series Lecture Notes in Computer Science pp 466-474.

14. ABSTRACT

Training medical personnel to maintain readiness for medical emergencies and combat-related operations is a critical problem. Distance learning solutions are required for enabling effective training while minimizing time away from the important on-the-job duties of providing quality medical care. Simulation-based training can significantly benefit learners by providing opportunities for hands on training. A simulation by itself, however, is not sufficient to enable learning. It must be accompanied by opportunities for reflection and a chance for learners to try their skills under different conditions. This means a simulation-based training course should include several scenarios. The high cost of developing and administering training scenarios renders this infeasible. We have developed a simulation-based training framework called SimCore that incorporates intelligent, automated assessment and coaching in support of self-paced learning. This reduces the need for human facilitators. A key feature of this framework is an authoring tool that supports rapid scenario development and customization and is designed for use by subject matter experts and course developers. This brings down the cost of scenario development. The system has been designed to interface easily with third-party simulators, with minimal effort. It also includes a Flash-based simulator that can be played on a web-browser. A beta version of SimCore is currently being distributed for evaluation.

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT **unclassified**	b. ABSTRACT **unclassified**	c. THIS PAGE **unclassified**	**Same as Report (SAR)**	**6**	

Standard Form 298 (Rev. 8-98)
Prescribed by ANSI Std Z39-18

According to Kolb [3], experiential learning is a cyclic process where a student works with a concrete situation, reflects on his experience, creates abstractions of the knowledge gained from the experience, and finally tries his new knowledge on other related situations. Simulations provide the concrete experience. Without reflection and abstraction, the experiential learning cycle would be incomplete. Typically instructors work with students to help complete the learning cycle but this makes the cost of simulation-based training considerably higher. Additionally, providing a personalized learning experience requires one-on-one instruction which can be prohibitively costly. Techniques from the field of Artificial Intelligence and Intelligent Tutoring systems can be applied to automate one-on-one personalized instruction including automated performance assessment, coaching and review. This will close the loop on the experiential learning cycle while freeing simulations from the requirement for dedicated instructors.

We are currently developing technologies for applying the concept of Intelligent Tutoring Systems (ITS) to healthcare simulations. ITSs are software tutors that are designed to provide one-on-one tutoring, much like humans [4]. As a first step in this direction, we are developing SimCore (**Sim**ulate, **Co**ach, **Re**view), a tool that adds intelligent performance assessment and coaching facilities to training simulations. Using SimCore, simulation developers can convert their simulations into automated, intelligent tutors. Freed from the need for hands-on instructor-led support, simulation systems can be used to deliver not one but a variety of scenarios, giving students the opportunity to learn by applying their newly acquired skills under varying conditions. SimCore allows for complex free-play simulations. It can be used to actively guide a student towards performing an action or a procedure, even as the simulation allows them the freedom to explore. Authoring tools are typically needed to make Intelligent Tutoring Systems a viable option [5]. SimCore includes a scenario authoring tool that can be used to rapidly customize existing scenarios or to create new ones.

2. Architecture

SimCore employs a client-server architecture with two server-side components (Figure 1): the authoring tool and the runtime server. The authoring tool enables the specification of training scenarios in abstract, high-level terms. The runtime server drives the simulation, initiates events, responds to and evaluates student actions, and provides after-action reports. The system has been designed to interface with third party simulators with minimal customization of the runtime engine or the authoring tool. A default Flash-based simulator is included with the SimCore package.

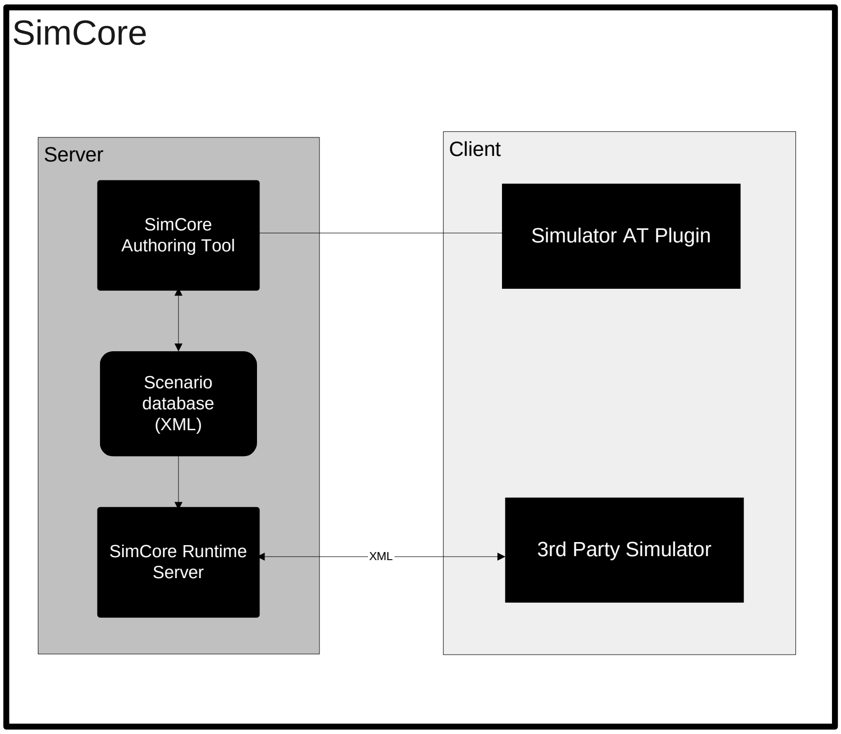

Figure 1; SimCore client-server architecture with two server-side components

The runtime server connects to a simulator via a simulator API. The simulator API must be customized to work with each specific Simulator. Communication between the runtime server and the simulator is done using XML. In general, the simulator is responsible for maintaining the user interface for one or multiple players. The runtime server is responsible for maintaining the state of the simulated world.

3. Scenario Model

SimCore is based on a generalized, domain-independent scenario model. In the runtime server, the actions, props, etc. are represented by a set of Java Objects. These building blocks are created by the authoring tool, stored in a series of XML files, and read in by the runtime server. A scenario is composed of locations, props, NPCs (non-playing characters), and events. Locations represent the various different locations in a scenario. For example, a side of the street which is the scene of an accident can be a location. An ambulance may be a second location. Props are scenario objects in locations and can have actions performed on them. The car involved in an accident is an example of a prop. NPCs are virtual human characters in the scenario. The person hurt in the accident, as well as his companions in the car would be modeled as NPCs. Props and NPCs are associated with attributes that represent their states at any given time. For an NPC, for instance, the modeled attributes may include his/her vital signs, age, medications, allergies. The scenario model includes actions that be performed on the NPCs and Props.

Each action has a unique command name, a textual representation to display to players, a series of possible textual replies upon performing the action, a series of sub-areas on the target where the action can be performed, a set of preconditions as to when this action can be performed, and a set of effects that are executed after the action is performed. For example, “Check Blood Pressure” may be an action with the precondition that the blood pressure monitor should be available for performing this action. The action definition would also include rules specifying the simulators response when the player performs this action. A scenario also includes events. Sometimes it is necessary to change the scenario state when the player does not initiate an action. Events are the way to accomplish this. Events are a set of effects that will be performed if a set of preconditions are met. They also contain a trigger set that, if equal to true, will cause this event's preconditions to no longer be checked and the event will be made inactive. An example is a box (prop) that is supposed to explode in 30 seconds if it has not been opened. An event is created that has a precondition that checks to see if 30 seconds has passed. If so, run the effect that causes the box to explode. The inactive section of the event checks to see if the box has been opened. If so, then the event's preconditions will no longer be checked, and the box won't explode because of this event. Authoring a scenario entails instantiating the scenario model. The Runtime Server executes this model to deliver a training scenario.

4. Automated Performance Assessment and Coaching

SimCore includes an automatic performance assessment component that also guides students towards an optimal solution path. Each scenario can be associated with a solution template using the SimCore authoring tool. A solution template is a generalized procedure or protocol that is required to complete the scenario successfully. The solution template is not visible to the player; it is used behind-the-scenes to assess student performance. By a generalized procedure, we mean that the procedure can include unordered sets of actions and conditional actions. The solution template defines the actions the student should perform. The scenario author can also specify specific error rules that represent actions the student should not perform. During each scenario, the simulator software sends notification messages to the SimCore engine that describe each student action and report the values of each simulation state variable. SimCore evaluates each student action by comparing it with the solution template and the error rules contained within the scenario definition. This information is passed back to the simulation which then selects an appropriate feedback. In addition, this information is included in the after-action review. The solution template is also used by SimCore to provide just-in-time hints which are associated with the solution template during authoring.

5. Authoring tool

The authoring tool is a very important component of the SimCore framework. It enables the rapid development of training scenarios. The authoring tool has visual editors for defining the scenario model described earlier. In addition, there is a section in the authoring tool for specifying the solution templates and error rules. Hints and feedback to help coach can also be specified using these editors. Figure 2 shows the Props editor where scenario author can create Props, define their attributes, and the actions that can be performed on them.

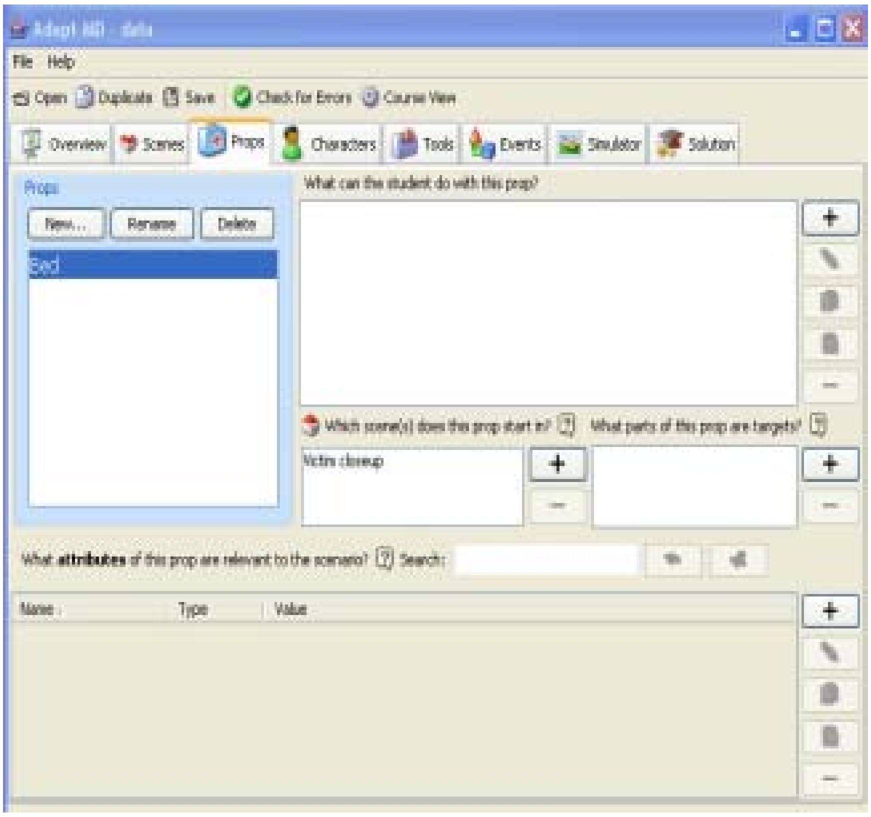

Figure 2: SimCore Authoring Tool

6. Evaluation

We developed the SimCore framework in three spirals. The software at the end of each spiral was demonstrated to experienced EMS trainers. Their suggestions have strongly influenced the design of the system. We are currently making the product available to beta testers and we will be collecting feedback from these users.

7. Related Work

A few intelligent tutoring systems have been developed for healthcare domains [6, 7]. In most such systems, the tutoring is tightly coupled with the simulation or the problem-solving interface. SimCore is based on a domain-independent scenario model and can thus be used for a broad range of domains. It is designed to be plugged into third-party simulators with minimal effort. [5] presents an in-depth discussion of authoring tools for Intelligent Tutoring Systems. The trade-off between power and usability is a central dimension that characterizes authoring tools. Our objective is to develop a tool that is usable by healthcare training experts who have little programming experience. Hence our first version of SimCore trades power for authoring simplicity. Future versions of the system will make the framework increasingly powerful while involving users in the development process to ensure that usability does not get sacrificed.

8. Future Work

SimCore beta is now available. SimCore is re-usable framework for developing low-cost, customizable simulations and training games for training EMS and other healthcare professionals. Using SimCore, subject matter experts and course developers can rapidly create simulations to test students' ability to apply their skills and decision-making capability in realistic scenarios. SimCore provides the facility to incorporate automated performance assessment and feedback, enabling organizations to create rapidly powerful, skill-application oriented self-paced training. A Flash simulator that is included as a part of the SimCore package makes the system usable by organizations that do not have a third-party simulator at their disposal.

In the near future, our primary objective to collect feedback from beta users on the system, understand how well it meets their needs, and gaps that must be addressed by future versions. Efficiency related improvements are also in the near-term agenda.

Our long-term vision for the product includes improved usability, enhanced student modeling, adaptive coaching, and reflective after-action review with Socratic dialogs.

References

[1] Schank, R. (1995), What We Learn When We Learn by Doing, *Technical Report no. 60*, Institute of Learning Sciences, Illinois.

[2] Johnson, W. B. and Norton, J.E. (1992). Modeling student performance in diagnostic tasks: A decade of evolution. In *Cognitive Approaches to Automated Instruction*. Eds. Regian, J.W and Shute, V.J. Lawrence Erlbaum Associates. 1992.

[3] Kolb, D. A. (1984) *Experiential Learning*, Englewood Cliffs, NJ.: Prentice Hall.

[4] Forbus , K. D., and Feltovich, P. J. (2001). *Smart Machines in Education*. AAAI Press.

[5] Murray, T., Blessing, S., and Ainsworth, S. (2003). *Authoring Tools for Advanced Technology Learning Environments: Toward Cost-Effective Adaptive, Interactive, and Intelligent Educational Software*. Springer.

[6] Shaw, E., Ganeshan, R., Johnson, W.L., and Millar, D (1999). Building a Case for Agent-Assisted Learning as a Catalyst for Curriculum Reform in Medical Education, In *Proceedings of the Int'l Conf. on Artificial Intelligence in Education*, July, 1999.

[7] Kizakevich, P.N., M. L. McCartney, D. B. Nissman, K. Starko, and N. Ty Smith (1998). Virtual Medical Trainer: Patient Assessment and Trauma Care Simulator. *Medicine Meets Virtual Reality - Art, Science, Technology: Healthcare (R)evolution*, J. D. Westwood, H.M. Hoffman, D. Stredney, and S.J. Weghorst, eds.,pp. 309-315, IOS Press and Ohmsha, Amsterdam.