An Architecture to Support Integrated Manikin-Based Simulations

James F. Leathrum, Jr.
Roland R. Mielke
Michel Audette
Frederick McKenzie
Dept. of Modeling, Simulation and Visualization Engineering
Old Dominion University
{jleathru, rmielke, maudette, rmckenzi}@odu.edu

Robert K. Armstrong
Geoffrey T. Miller
Andrew E. Cross
Sentara Center for Simulation and Immersive Learning
Eastern Virginia Medical School
{armstrrk, millergt, crossae}@evms.edu

Mark W. Scerbo
Dept. of Psychology
Old Dominion University
mscerbo@odu.edu

ABSTRACT
A radically new approach to the design and implementation of high-technology, computer-driven manikin-based simulation environments for medical training is proposed. The approach utilizes a novel, component-based manikin hardware architecture currently under research for the U.S. military. A new distributed software architecture is proposed in this paper that facilitates interoperability among manikin components and provides the potential to easily integrate other real or virtual devices normally present in a real medical facility.

Author Keywords
Manikin-based training, medical training environment, simulation architecture, system interoperability.

1. INTRODUCTION

The U.S. military has recognized the need for the development of new and novel approaches for realizing the next generation of medical manikins to be used in support of diverse military medical training curricula. The desired manikin must be scalable, from a rugged basic unit for use in the field to a vastly more capable unit for use in hospital training environments. The manikin must be flexible, having end-user modifiable configuration options and easily customizable functionality. It must be compatible with a wide array of peripherals and/or extensions provided by numerous different sources. Ideally, the manikin would use an open standard/open source development approach. And the supporting software architecture must be open and extensible; that is, it must be capable of accommodating new training techniques and manikin technology that does not exist yet but that may be developed in the near future. This is a substantially new and different approach to manikin design and development compared to the current state of the medical simulation industry:

- Current systems have relatively fixed configuration options, often requiring that a new manikin be built or procured for each new desired capability.
- Current "standards" for software, hardware, and system function are vendor-specific. Even within the same vendor, manikin models and accessories are not guaranteed to be interoperable.
- Current manikin systems are not built to be compatible with future innovations.
- Current manikin systems are not designed to function as an integrated component of a complete learning environment.

Numerous studies have illustrated the benefits of manikin-based training and the use of virtual environments in medical training [1]-[8]. In support of improved manikin-based training, the U.S. Army currently is funding efforts for the development of an Advanced Modular Manikin [9]. This research effort focuses on development of a physical modular manikin where different manikin components can be interconnected to create various medical scenarios such as tension pneumothorax or traumatic femoral bone fracture. This modular approach requires techniques for the development of realistic manikin components and, more importantly, standards for the interconnection of these components, to include methods for physical connection of components, the sharing of fluids, interconnection of airways, and the exchange of data. Missing in the effort though is the development of a software architecture enabling the integration of the individual manikin components into an integrated learning environment to include external, virtual devices. This architecture clearly must support current and future innovations in manikin development. In addition, however, the architecture also should enable manikin control by a sophisticated human physiology engine such as BioGears [10], HumMod [11], or HumMod-Golem [12], the inclusion of external medical devices, both real and virtual, the inclusion of methods for learner assessment, and built-in automated prescriptive performance feedback. The focus of this paper is to

SCSC 2015, July 26-29, 2015, Chicago, IL, USA

propose a software architecture to address these requirements.

The paper is organized in six sections. In Section 2, a high-level description of the system is presented. Included is a statement of important system requirements. In Section 3, a user model is described. The role of instructors, learners, and administrators are considered. The proposed system architecture is presented in Section 4. This discussion includes an explanation of the purpose and functionality of the main architecture components. In Section 5, the suggested architecture for other devices and external components that might be integrated with the main system is defined. These devices might include real or simulated medical equipment or external software systems for data logging for feedback (formative and summative) and after-action review. Conclusions are presented in Section 6.

2. SYSTEM DESCRIPTION

In support of the new generation of high-technology manikins, a new distributed, component-based software architecture is proposed for system control and data management. The heart of this system consists of three main software components: a scenario controller/ configuration manager component, a physiology engine component, and an interface manager component that facilitates communication. The three main software components all must be present for system operation and they interconnect to additional system components via the interface manager. The physiology engine software component facilitates the use of compatible existing physiology engine designs or new designs as they are developed. All other system components are assumed to possess local computational capability and each is designed to interoperate with all other system components.

The proposed distributed software architecture has significant advantages compared to present day black-box, proprietary system designs. First, the architecture provides manufacturers of manikin components with an open, well-defined interface to guide their designs. Even if manufactures wish to protect intellectual property, they are able to hide their proprietary code within an appropriate interface wrapper.

Second, the modular design facilitates the introduction of additional components, not usually associated directly with human-patient simulator systems, which provide a more realistic training environment. The manikin might be placed in a three- or four-walled immersive environment, where a realistic training environment is established by projecting appropriate background images. Virtual patient monitors displaying physiologic parameters also might be projected on the wall display. Even interactive virtual medical equipment simulators, such as an infusion pump, can be implemented on wireless tablets and used in place of real equipment that often is too expensive, unavailable for training, or too expansive in numbers and variations to be practical and feasible for training centers to purchase and manage. Each of these environment-enhancing simulations can be realized as an interoperable component in the modular software architecture. Examples of the use of simulated components having the capability to support manikin-based medical training are described in [13].

Third, the modular software design facilitates and even encourages the use of other software systems that might support and/or further enhance the desired training outcomes. Software companies might provide tools for scenario design. Training companies might develop advanced after action review tools for use with the system. Companies that make educational products might develop learning management systems for managing the overall simulation training environment. The system architecture design encourages these vendors by providing open access to the design and a clearly defined interface for component integration.

Finally, the proposed system design motivates and encourages development of interface standards for all hardware components and system software components. Such standards would facilitate realization of medical training environments that incorporate components from different manufactures and vendors. In the long run, the component architecture concept will be successful only if these interface standards are developed and then accepted by a broad audience of program participants.

3. USER MODEL

The overall system functionality is defined by the requirements specifications. The user interface represents *how* the functionality is controlled by the user. The system must support different classes of users (i.e., learners, instructors, and simulation technicians/operators), each having different objectives. Thus, the functionality may differ for different classes of users, but the interface should be common across all users. The system also must support two primary modes of operation, training and assessment. During training, formative feedback is provided during the execution of a scenario to facilitate learning. During assessment, summative feedback is provided after scenario execution and utilizes logged scenario events and embedded scoring rubrics to support after-action review of performance.

Three classes of users are modeled:

- *Instructors* – Instructors develop the scenario and then control the execution of the scenario. During scenario creation, they interact with a scenario administrator to define learning outcomes, objectives and performance indicators, including desired learner behaviors. During training, predefined scenarios are selected or modified as necessary to meet desired learning outcomes. The scenario is run, scoring performance, to provide specific and individualized performance improvement and feedback/debriefing to the learner.

- *Learners* – Learners interact with the manikin, medical devices, and various monitors present in the learning environment. The learner makes appropriate diagnoses, performs clinical procedures, and executes decision-making, for which they are assessed. On completion, the learners participate in a focused debriefing session with the instructor and/or simulator using the after action review system. Here they receive feedback concerning their performance and suggestions for improvement.
- *Simulation Technicians/Operators* – Simulation technicians/operators manage the technical aspects of system operation. While instructors provide subject matter expertise in the design of scenarios, technicians/operators are responsible for implementing this information into a coherent simulation. This involves developing the scenario in a database, setting up the physical system (manikin, devices, etc.), and managing the actual execution of the scenario. Their overall function is to remove the technical requirements from the instructor/faculty.

Instructors must be able to work with scenarios to meet pedagogical requirements and therefore need to manage the scenarios, the learners, and the resulting data. Learners may use the system to manage their own training, need to access to specific training scenarios, and need to access their training performance results. Further, the system must support training modes having full feedback and summative assessment modes with limited feedback to learners. Lastly, simulation technicians/operators need to manage all of the content in the system.

4. SYSTEM ARCHITECTURE

The system architecture provides a framework for the independent development and interoperability of individual simulator/manikin components and software subsystems. It defines required subsystems as well as optional subsystems to support a learning environment based on a specific scenario. The architecture establishes a basis for developing standards for component communication, software interfaces, and database requirements. Together, these standards constitute a system-level standard. The architecture provides the flexibility for integration of new technologies, whether manikin, communication, software, or computer technology. Finally the architecture supports extensibility, the capability to include future enhanced simulator/manikin components, new virtual medical devices, new after-action review capabilities, and advanced learning management systems.

The architecture identifies the main system software components and defines how these components interact to establish desired system functionality. In addition, the architecture facilitates design of the primary user interfaces. The basic architecture is illustrated in Figure 1 with user interactions color coded. The architecture is a distributed system and borrows many concepts from the High Level Architecture (HLA) for distributed simulation defined under IEEE Standard 1516 [14].

The primary architectural components are defined in the following.

- *Scenario Manager* – The scenario manager handles the development and execution of specific simulator learning scenarios. The development involves the creation of the learning module and the configuration of the physical architecture required to support the scenario. This information is stored in a scenario database and used to initially configure the system and then to control the execution of the scenario.
- *Physiology Engine* – The physiology engine models the physiological state of the patient that is represented by the simulator.
- *Simulator Components* – The architecture supports the development of individual components including torso, legs, arms, head, etc. These components can be interconnect to form a complete manikin or used individually for part-task training. In addition, the modular implementation of the simulator supports hierarchical definition of subcomponents, such as the lungs and stomach, which would be considered components of the torso.
- *Environmental Components* – The architecture is extensible and can include external environmental components that support required realism for a given training environment. Examples include monitoring devices such as cardiac monitors, capnography, or remote sensing devices such as ultrasonography. These devices can interface with the architecture to acquire simulated data and then can provide data back based on the state of the simulator. The architecture also supports integration in an immersive virtual environment as a component within the architecture.
- *Communication Interface* – The communication interface provides a means by which architecture components can communicate with each other during the execution of a scenario. All manikin sensor data and physiology data are communicated over this interface. The interface can be supported by wired or wireless hardware communication, wireless being preferred but wired supported where the environment is not suitable for wireless. Specification of a specific hardware implementation is avoided to allow for future technology upgrades; rather, data standards and communication protocols are utilized to enable interoperability.
- *Interface Manager* – The interface manager is the glue that establishes the interoperability across the array of components in the architecture. Device functionality is provided to:

- Manage the communication between components over the communication interface.
- Manage communication for the initialization of the system.
- Perform time management, ensuring components are advancing time at the same rate.

- *Configuration Manager* – The configuration manager handles the initialization of the system based on the scenario data stored within the database. It ensures that appropriate components, with appropriate functionality, are communicating with the system. This initialization includes identifying that the physiology engine and simulator components are selected properly to support the requirements of the scenario. For example, if training for a pneumothorax is specified, the correct torso component must be employed in the simulator definition.

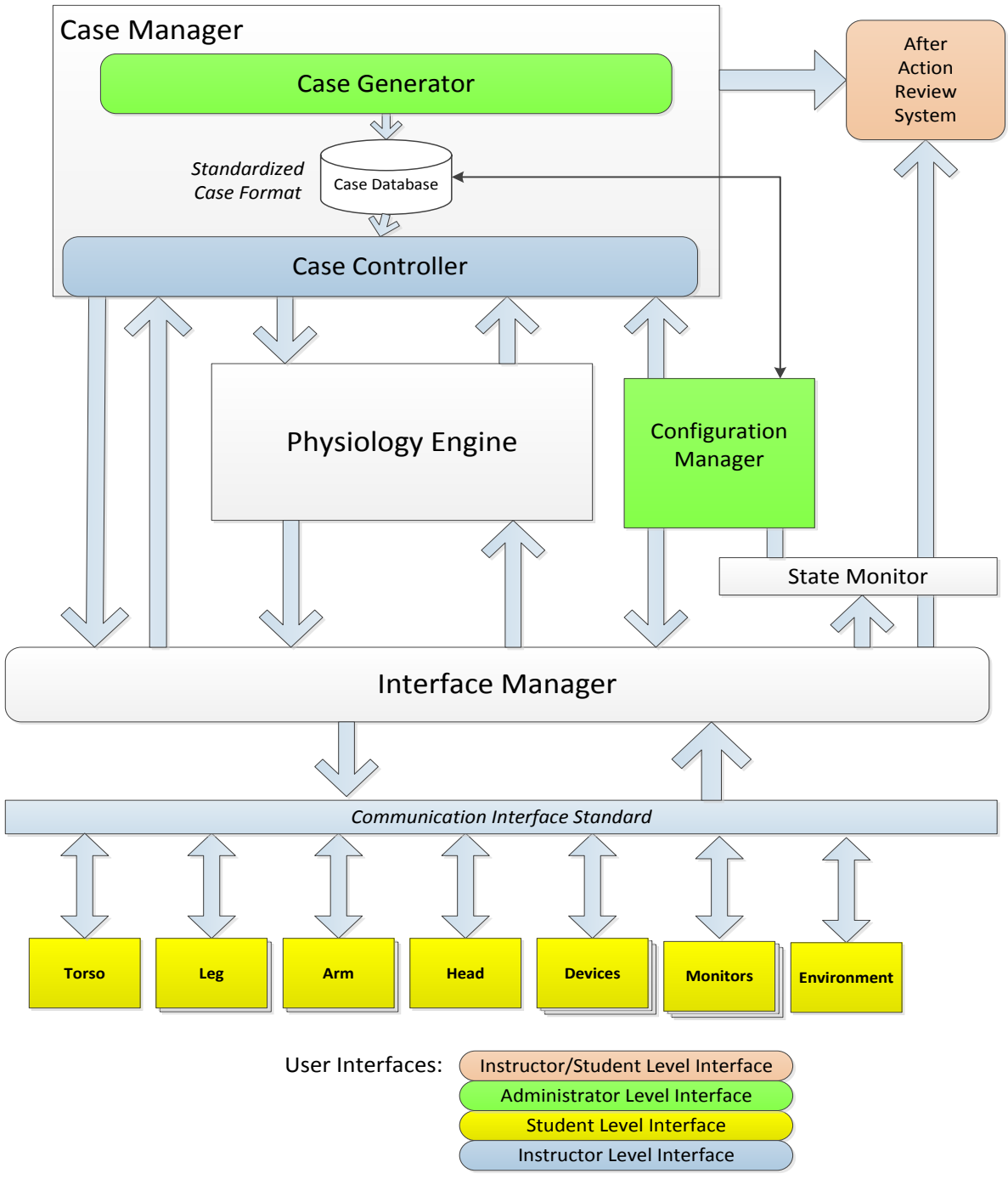

Figure 1. System Architecture.

- *State Monitor* – During the execution of a scenario, the system must monitor and record information to support the feedback, after-action review, and evaluation of the learning outcomes, objectives, and performance indicators. The state monitor identifies the relevant information being communicated during scenario execution, and then records that information with appropriate time stamps.
- *After Action Review System* – Using data acquired by the state monitor, the after-action review system allows the instructor and learner to review the scenario execution for learning assessment and feedback.

The components are organized as computing platforms that interact with each other. Each individual simulator and environment component must have sufficient computational power to not only manage sensor and display capabilities, but also to communicate within the architecture. The individual computing platforms can be partitioned in numerous fashions based on developer choices.

5. DEVICE ARCHITECTURE

The system architecture provides the infrastructure to interconnect external devices. External devices are devices attached through the communication interface. These devices include individual manikin components (torso, legs, arms, head, etc.), real and/or virtual medical devices (stethoscopes, ultrasound equipment, etc.), medical monitors (cardiac monitors, pulse oximeters, anesthesia systems, etc.), and instructor interfaces. In this section, a notional hardware architecture is presented for future development of devices and the requirements for standards are highlighted.

5.1. Notional Hardware Architecture for External Devices

From a system architecture perspective, the primary requirement for external devices is that they be able to communicate through the communication interface. Application Programming Interfaces will be developed to facilitate communication, but that functionality could be developed internally to the device. While the internal hardware architecture of these devices is left to the manufacturer, and is not imposed by the architecture or standards, it is helpful to develop a notional hardware architecture for the purpose of explaining how devices will interface to the main system.

Figure 2 presents a notional hardware architecture for a manikin subsystem. Each manikin subsystem is developed to operate as a stand-alone simulator component or as part of a complete human-patient simulator system. Each subsystem is envisioned to utilize a microcontroller to manage the interface to the communication interface and to implement a local physiology model that is controlled through updates from the main physiology engine. Based on state changes occurring locally due to learner activity, the microcontroller updates the physiology engine state. In addition, the physiology engine could update the local

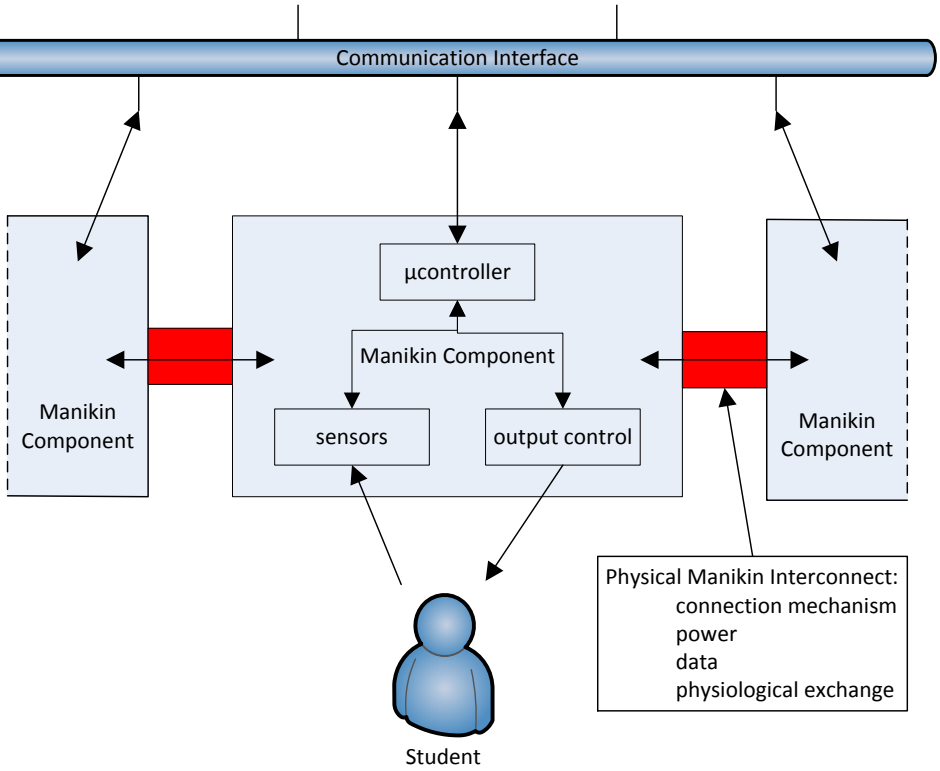

Figure 2. Notional Manikin Hardware Architecture.

physiology state, such as changing the pulse rate of the manikin. The microcontroller also manages the device interface with the learner. Device interfaces might include control of physiology indicators observable by the learner (pulse, breathing, etc.) or by external medical devices (for example outputs to allow proximity sensing for medical device placement). Finally, the microcontroller also manages sensor data that indicate learner intervention with the manikin.

Medical devices and monitors are envisioned to be subsets of the simulator hardware architecture. Medical devices primarily act as sensors to detect proper use of the device, and then to generate appropriate output to the manikin. Monitors accept physiology information and then display this information in the appropriate format for use by the learner. The medical devices, to include virtual devices (IV pumps, etc.) and virtual reality trainers (laparoscopic, endoscopic, intravascular, etc.) would require a similar architecture to the manikin, requiring microcontroller support to communicate with the rest of the system.

5.2. Standards Requirements in Support of the Architecture

Definition of a system architecture and a notional device architecture are required in order to be able to develop new simulator standards. Standards in turn facilitate the development of system components by different manufacturers. Standards development is required for scenarios, communication, software interfaces, and hardware interfaces. These standards are described in the following.

- *Scenario Model*: The scenario model defines the specific requirements of individual scenarios. Scenario models are maintained in the scenario database. An individual scenario model defines:
  - *Learning Model* – The learning outcomes, objectives and performance indicators for the scenario. This includes the data requirements for feedback and after action review.
  - *Physiological Scenario* – The required characteristics of the physiology engine are defined. This supports the selection of an appropriate physiology engine to support the scenario under study.
  - *System Components* – The simulator and external environment components are enumerated and their roles are defined.
  - *Data Model* – The data communicated between components are defined; these definitions include data semantics and types. Then for each data element, the data producer and data consumers are identified.
- *Communication Interface Standard* – This standard defines the data types and formats for communication between physical components in the system. Communication protocols support the data movement through the system. Existing standards under consideration include DICOM [15] and IEEE 11073 [16].
- *Software Component Interfaces* – Standard Application Programming Interfaces (API) are defined for each software component. The API establishes how software components interact with the system. The APIs enable independent software development of components and the integration of existing software. An example API standard that enables software interoperability is POSIX (IEEE Standard 1003 [17]) supporting operating system interoperability. APIs are defined for the Scenario Manager, Physiology Engine, Configuration Manager, Interface Manager, Manikin Components, Environment Components, State Monitor, and After-Action Review System.
- *Manikin Hardware Interfaces* – Hardware interfaces for manikin components are required in addition to software interfaces. These interfaces include physical interconnect, power sharing and storage, physiology data interfaces, and potential physical data interconnects.

6. CONCLUSION

The architecture presented in this paper enables a leap in human-patient simulator technology by supporting the modularization of individual simulator components, the inclusion of external medical device simulators, and the development of a feedback and after-action review and assessment component. The architecture is scalable and encourages commercial development through standardized interfaces.

An Old Dominion University senior capstone design team, working in conjunction with the Eastern Virginia Medical School, has demonstrated an initial proof of concept of the communication between external virtual medical devices and a Laerdal SimMan 3G manikin system [18], [19]. This prototype system allows a learner to inject a drug through a virtual infusion pump developed on an Android tablet, communicate that action to the manikin system, and then to observe the change in the manikin's physiology on a virtual patient monitor that also is developed on a tablet.

ACKNOWLEDGEMENTS

The authors would like to thank the reviewers for their insightful comments and feedback. The reviewers (with links to their comments) are:

- Kent Denmark: https://groups.google.com/forum/#!topic/public-scientific-reviews/v7guJsVbQU4
- Eric Bauman: https://groups.google.com/forum/#!topic/public-scientific-reviews/TGgiBf0Cyxs
- Mark Ottensmeyer: https://groups.google.com/forum/#!topic/public-scientific-reviews/q9GDauQylD4

- Jacob Barhak: https://groups.google.com/forum/#!topic/public-scientific-reviews/VPaDcQNiR80

REFERENCES

1. M. Aebersold, D. Tschannen, and M. Bathish, "Innovative Simulation Strategies in Education," Nursing Research and Practice, Vol. 2012, 2012 (available online at: http://dx.doi.org/10.1155/2012/765212).
2. S. Lapkin, T. Levett-Jones, H. Bellchambers, and R. Fernandez, "Effectiveness of patient simulation mannequins in teaching clinical reasoning skills to undergraduate nursing students: a systematic review," *Clinical Simulation in Nursing*, vol. 6, no. 6, pp. e207–e222, 2010.
3. B. Harder, "Use of simulation in teaching and learning in health sciences: a systematic review," *Journal of Nursing Education*, vol. 49, no. 1, pp. 23–28, 2010.
4. D. Wayne, A. Didwania, J. Feinglass, M. J. Fudala, J. H. Barsuk, and W. McGaghie, "Simulation-based education improves quality of care during cardiac arrest team responses at an academic teaching hospital: a case-control study," *Chest*, vol. 133, no. 1, pp. 56–61, 2008.
5. M. Boulos, L. Hetherington, and S. Wheeler, "Second Life: an overview of the potential of 3-D virtual worlds in medical and health education," *Health Information and Libraries Journal*, vol. 24, no. 4, pp. 233–245, 2007.
6. E. Conradi, S. Kavia, D. Burden et al., "Virtual patients in a virtual world: training paramedic students for practice," *Medical Teacher*, vol. 31, no. 8, pp. 713–720, 2009.
7. P. Youngblood, P. M. Harter, S. Srivastava, S. Moffett, W. L. Heinrichs, and P. Dev, "Design, development, and evaluation of an online virtual emergency department for training trauma teams," *Simulation in Healthcare*, vol. 3, no. 3, pp. 146–153, 2008.
8. S. Galloway, "Simulation Techniques to Bridge the Gap Between Novice and Competent Healthcare Professionals," *The Online Journal of Issues in Nursing*, vol. 14, no. 2, May 2009.
9. *Advanced Modular Manikin*, U.S. Army Medical Research Acquisition Activity, Department of the Army, Solicitation Number: W81XWH12R0032.
10. BioGears (2015), Available online at: https://www.biogearsengine.com/.
11. R. L. Hester, A. J. Brown, L. Husband, et. al., "HumMod: A Modeling Environment for the Simulation of Integrative Human Physiology," *Frontiers in Physiology*, vol. 2, no. 12, 2011.
12. J. Kofránek, J, Mateják, M, Privitzer, P, Tribula, et. al. "HumMod-Golem Edition: large scale model of integrative physiology for virtual patient simulators," *Proceedings of World Congress in Computer Science 2013 (WORLDCOMP'13), International Conference on Modeling, Simulation and Visualisation Methods* (MSV'13), pp. 182-188,
13. M. Samosky, J. Thornburg, A. Karkhanis, et al, "Enhancing Medical Device Training with Hybrid Physical-Virtual Simulators: Smart Peripherals for Virtual Devices," Studies in Health Technology and Informatics, 184, 377-379, 2012.
14. *IEEE Standard for Modeling and Simulation (M&S) High Level Architecture (HLA) – Framework and Rules*, IEEE Standard 1516, 2010.
15. *Digital Imaging and Communications in Medicine*, The DICOM Standard 2015a, 2015.
16. *IEEE Standard for Health Informatics – Point-of-Care Medical Device Communication*, IEEE Standard 11073.
17. *IEEE Standard for Information Technology – Portable Operating System Interface (POSIX®)*, IEEE Standard 1003, 1996.
18. K. Rusnak, M. Lewis, F. Ashour, J. Mellot, and M. Conley, "Medical Virtual Integrated Training Environment (VITE)," *ModSim World 2015*, Virginia Beach, VA, 2015.
19. F. Ashour, K. Rusnak, M. Conley, M. Lewis, and J. Mellott, "Implementation of a Simulation-Based Medical Training Environment," *Modeling, Simulation, and Visualization Student Capstone Conference*, Suffolk, VA, 2015.