Simulation-Based Evaluation of Medical Workstations Designed for Human Space Exploration

Jacob Richardson, Brian Dilinila, Thomas Tracey, Minh Dong, John Paul Asija, Michael Poteat, Jake Webster
Old Dominion University
Norfolk, Virginia
jrich087@odu.edu, bdili002@odu.edu, ttrac005@odu.edu, mdong004@odu.edu, jasij001@odu.edu, mpote001@odu.edu, jwebs001@odu.edu

ABSTRACT

The Exploration Medical Capability (ExMC) element at NASA Langley Research Center desires a process for evaluating and comparing designs for medical workstations capable of supporting long-duration human space exploration. The authors, as part of their senior capstone design experience, are jointly charged with both designing an evaluation process and implementing a prototype of that process as a proof of concept. A key component of the evaluation process is a simulation having the capability to assign quantitative values to dynamic performance measures. Dynamic performance measures are evaluated by observing the conduct of an activity or task over a period of time. Examples of dynamic performance measures include the time required to complete a task, or the volume/shape of the spatial envelope required to perform a task. The design team has determined that using a discrete event simulation, in which avatars are monitored while performing representative medical tasks, is the best approach for evaluating dynamic performance measures. The focus of this paper is the process of defining those metrics, as well as the approach to evaluating the dynamic metrics of the corresponding simulation.

ABOUT THE AUTHORS

All authors of this paper are seniors at Old Dominion University (ODU) graduating May 2018 in the undergraduate program for Modeling, Simulation, and Visualization Engineering.
Jacob Richardson is currently an intern at Jefferson Applied Research funded by the DOE as well as a tutor for the Modeling and Simulation Engineering department at ODU.
Brian Dilinila is currently an intern under Combat Direction Systems Activity Dam Neck representing NAVSEA at and providing assistance to Joint Staff J7.
Thomas Tracey is both a teaching assistant for Modeling and Simulation courses at ODU and a research assistant working on a graphical user interface used for assisting in the education of discrete event system simulations.
Minh Dong is interested in game development and is currently working on a personal video game project.
John Paul Asija is a research assistant in the Darden College of Education at ODU funded by the NSF GEODE grant.
Michael Poteat is a research assistant at Old Dominion University working on computational recognition and segmentation of secondary structure elements embedded within a larger protein image.
Jake Webster has interests in game and mobile development.

Simulation-Based Evaluation of Medical Workstations Designed for Human Space Exploration

Jacob Richardson, Brian Dilinila, Thomas Tracey, Minh Dong, John Paul Asija, Michael Poteat, Jake Webster
Old Dominion University
Norfolk, Virginia
jrich087@odu.edu, bdili002@odu.edu, ttrac005@odu.edu, mdong004@odu.edu, jasij001@odu.edu, mpote001@odu.edu, jwebs001@odu.edu

INTRODUCTION

As of 2018, many advanced challenges in astronautics have been achieved. This includes the challenge of sending numerous space exploration vehicles to Mars. However, the National Aeronautics and Space Administration views this as only the beginning. One of the many near-horizon goals of NASA is that of advancing deep space exploration, as well as sending humans to Mars. Advancements in deep space exploration have in turn resulted in a new and innovative space capsule design, which will facilitate future voyages. On the other hand, these advancements raise many questions. One of these questions involves the logistics involved in maintaining crew health and safety. NASA aims to further develop its ability to accommodate medical needs and provide preventive measures aboard the deep space capsule in future manned voyages.

As the ability to send astronauts on deep space voyages becomes more of a reality, the ability to keep astronauts safe and healthy becomes more of a priority. The effects of zero gravity on the human body for extended durations is still an area in which many questions are unanswered. With this being the case, NASA must make sure that the medical capability is in place to serve the astronauts should the time arise. One of the current challenges that NASA is facing is how to make a differentiation between two medical workstations so that comparisons and differences can be highlighted when comparing designs. A series of medical procedures performed in each medical workstation would provide a foundation for determining how the astronauts move around within the confined space, how they access the supplies within the workstation, and the ability to carry out a given procedure.

The Capstone Team at ODU has taken the given problem and broken it down into a single procedure that can be replicated across multiple designs. The Capstone Team has completed this process in order to generate data NASA can utilize to assist in differentiating between designs. First, it is important to note that any given procedure can be broken down into a series of atomic tasks which can then be measured through the use of performance measures. Two different types of performance measures will be utilized in the design of the problem solution.

Static performance measures will be generated through the leveraging of knowledge from subject matter experts (SMEs). Through the use of simulation, multiple runs on a series of tasks provides the capability of being able to evaluate the dynamic performance measures associated with that set of tasks. In order to capture this data, markers will be used to denote a start and stop point within the simulation which will be directly associated with a dynamic performance measure. The Capstone Team has taken the challenge at hand and has designed a solution which will assist in the evaluation of two medical workstations. Figure 1 includes a high-level design of the solution the Capstone Team will implement.

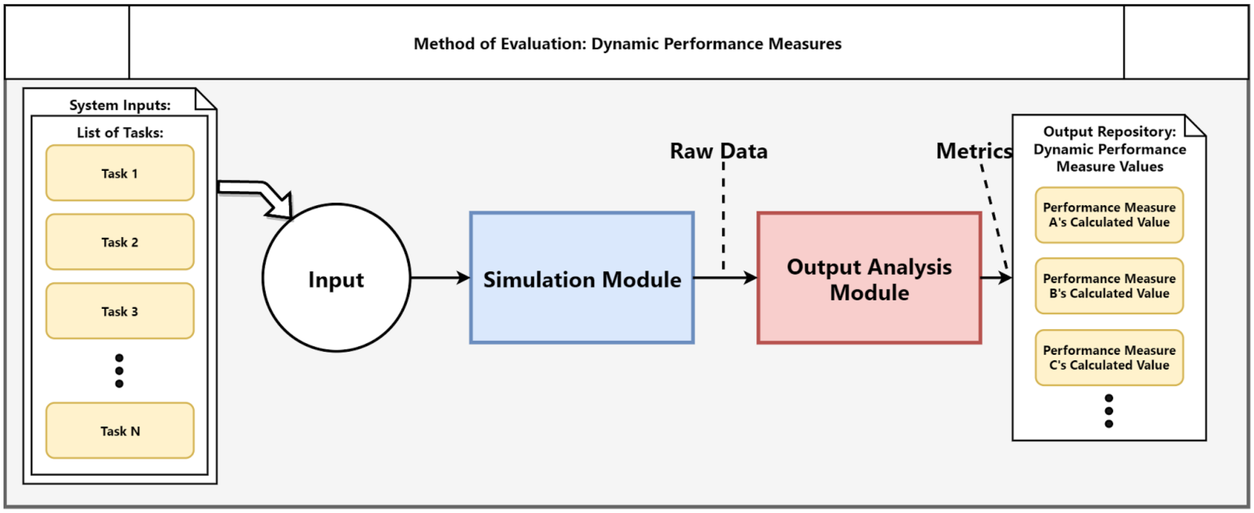

Figure 1. High-Level View of Three Main Components of the System.

This paper addresses the Capstone Team's solution for NASA: a high-level design of the solution, as well as a discussion of how the design components will function towards meeting the overarching goal of supporting NASA ExMC in regards to the evaluation of medical workstations. It discusses each component of the solution including the inputs to the system, the core simulation module as well as the output repository which facilitates the computation and analysis of the medical workstation. The following sections discuss our high-level system architecture as well as the technical approach to our solution and, specifically, the system's use, the development of performance measures as well as the development process of performance measures. The following sections also discuss the two different types of performance measures and the association of performance metrics, as well as the evaluation and visualization of the performance metrics.

SYSTEM ARCHITECTURE

With regard to the system architecture of The Capstone Team's solution, the simulation tool has been broken down into three major components: the input module, simulation module, and the output analysis module. The input to our system will be a set of parameters provided by NASA, which include the design of the medical workstation and the set of medical tools and their locations. The simulation module will take the information that was provided as an input, and - through the use of performance measures - generate output which will be used to create performance metrics. Performance measures have been separated into two different categories: static and dynamic, and the use of simulation allows for the Capstone Team to evaluate dynamic performance measures. The process for evaluating static performance measures as well as dynamic performance measures will be explained in further detail in the sections to come. The final module within our system architecture, the output analysis module, will take the raw data from the simulation module and generate performance metrics. Then, through the use of these performance metrics, the output analysis module will generate a visualization of the resulting data.

METHOD OF EVALUATING DYNAMIC PERFORMANCE MEASURES

Identifying Performance Measures

Figure 2 provides the process of developing performance measures. The process begins with developing an understanding of the system. Once completed, performance measures can be brainstormed to highlight areas where differences in workstation designs may exist. After completing a list of performance measures, a panel of SMEs must convene to discuss the validity of each performance measure. If the list is not approved, then it must be modified. Else, the process can continue so information can be gathered for the performance measures. The values provided by the performance measures can help NASA determine differences in designs or determine if changes to the measures are necessary.

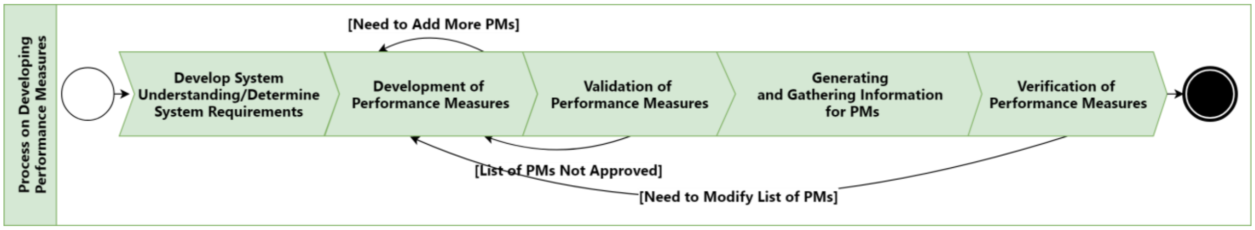

Figure 2. Process of Developing Performance Measures.

NASA-LaRC is providing a representative scenario for treating cardiac arrest. By breaking down the scenario into "tasks" (i.e. goals to be reached for the procedure), and further down as actions, the scenario constitutes a process which falls under the modeling perspective of a process flowchart. This approach is popular in developing system models that fall under the discrete event simulation paradigm. The use of process-based discrete event simulation exist in tools such as Arena (Choi and Kang, 2013), where processes consist of process blocks that have sequential flow to other blocks. This is utilized when it is necessary to observe and understand how different processes interact over time. This allows for an improved method to identify events within the system.

These actions can be transformed into logic blocks to build together the process flowchart model. Events start upon the activation of processes and end when their corresponding processes end. Times to complete tasks can vary by introducing stochastic values for time. These times can be defined by the end user of the simulation. The end user can as well introduce probabilities for certain conditions in the scenario.

By building the flowchart, we can identify areas where workstation designs may be different. Information such as time spent moving through the workstation, spending time away from the patient, etc. may be considered as information necessary for NASA's evaluations. Therefore, markers may be placed within the process flowchart model to state where information can be recorded. This information may then be sent as raw output data to be used for calculations for the dynamic performance measures' metrics.

From Figure 3, the scenario has been divided into five tasks: 1) Moving the Patient to the Workstation, 2) Monitoring the Patient's Heart Rate, 3) Checking for Throat Blockage, 4) Checking for Fluid in Lungs, and 5) Applying AED and Conducting CPR. When all tasks have been completed the scenario has been completed.

Figure 3. High-Level Perspective of Process Flowchart Model.

Task Five (Figure 4) involves applying the AED on the patient to analyze the patient's heart rhythm. Unlike the previous two tasks, tool modularity is present. The AED has multiple uses such as analyzing the patient's heart rate along with applying shock. This is an example of the concept that tasks are associated with tools.

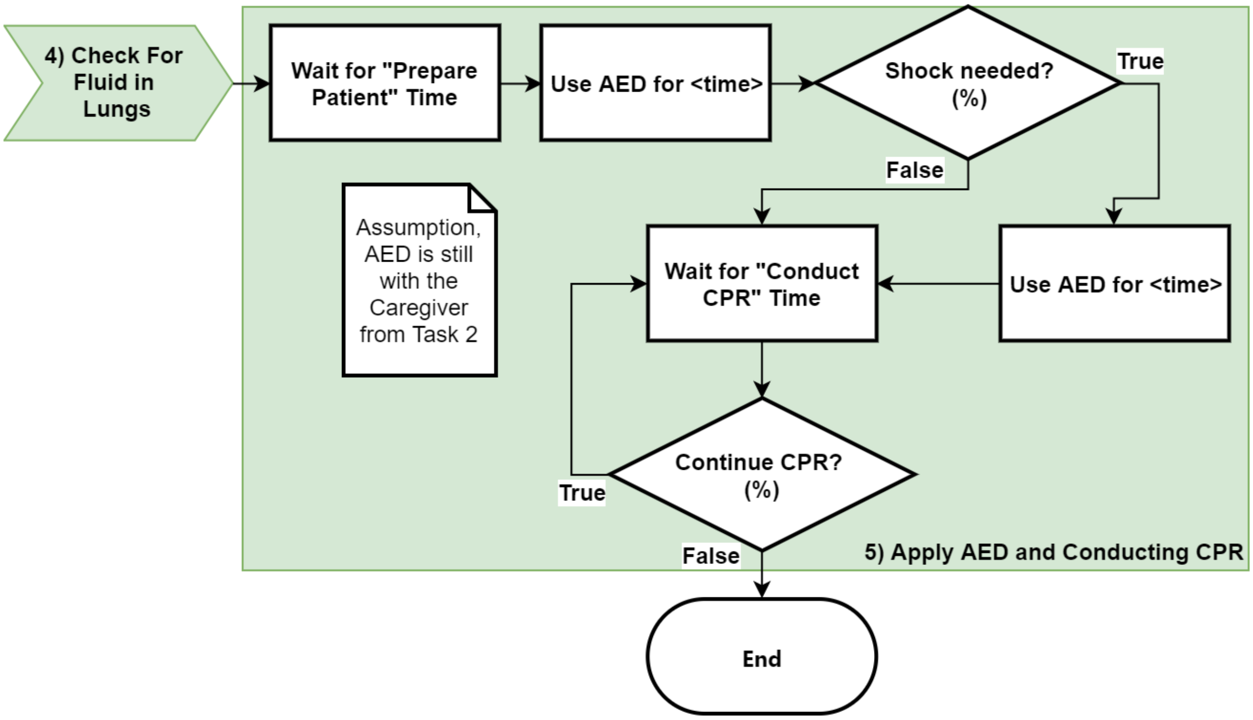

Figure 4. Process Flowchart for Task 5: Completing CPR.

Evaluating Static Performance Measures

Static performance measures focus on the features of medical workstations without the need of the observation of system behavior over time. These measurements delve into areas such as the dimensions of the workstation and habitat, the initial quantity of resources, etc. Due to the interdisciplinary nature of this project, the method of evaluating static performance measures requires the convening and consultation of SMEs. Static performance measures are provided to this panel of experts, whom are responsible for providing values for each measure. These experts can produce the values by methods such as looking up values from tables, conducting calculations, or even providing educated guesses.

Evaluating Dynamic Performance Measures

As opposed to static performance measures, dynamic performance measures' values are collected through the ongoing behavior of the system under study. In order to observe this behavior, a simulation is necessary. Example dynamic performance measures include time performed for tasks and constraint violations (e.g. time, spatial). Dynamic performance measures are not an input to the system. However, they are associated with representative tasks derived from sample scenarios. These tasks are among the number of inputs to the simulation. The tasks are carried out within the simulation module which produces raw data in the form of time-stamped events as output. Time-stamped events are records of changes of the system state. Examples of events are "Start Task" and "End Task" which correspond to time duration of tasks. This output acts as the necessary input to the output analysis module which calculates metrics. These metrics are the values associated with dynamic performance measures. Some dynamic performance measures may have more than one metric. The metrics are then saved within an output file and stored separately from the static performance measure values.

SIMULATION MODULE

The simulation module that produces the necessary data for calculating metrics consists of three components: a script, the simulation source code, which contains the process blocks used for building the process flowchart, and the output repository. The script is in human-readable format and is defined by the user through the corresponding input module. This script contains the initial system state, which includes information on the tools and resources available, the position and orientation of the actors, and a sequential list of actions. These actions are sent to the simulation to build a process block and execute it in order to change the state of the system. The commands' parameters include information such as an actor executing a process, the destination point that an actor must move to, and what tool to use. After execution, the system state is sent to the output repository in the form of time stamped events.

OUTPUT ANALYSIS MODULE GRAPHICAL USER INTERFACE

The data that comes out of the simulation will be then be output as time-stamped events. Because of this, the output analysis module will need to have the capability of parsing the data, as well as the markers that were put in place in order to generate meaningful dynamic performance measures. The output analysis module will not only utilize time-stamped events as an input but the static performance measures as well. Through the extraction of data from the markers and time-stamped events, along with the combination of static performance measures, the output analysis module will possess the capability of calculating meaningful performance metrics. These performance metrics will be used to provide the user with feedback regarding different aspects of the medical workstation. Figure 5 is an example of how data would be extracted by using markers. The yellow stars denoted the starting point at which the simulation begins to record data. A corresponding ending marker would also be present that denotes the position at which the simulation should stop recording in order to generate a given metric.

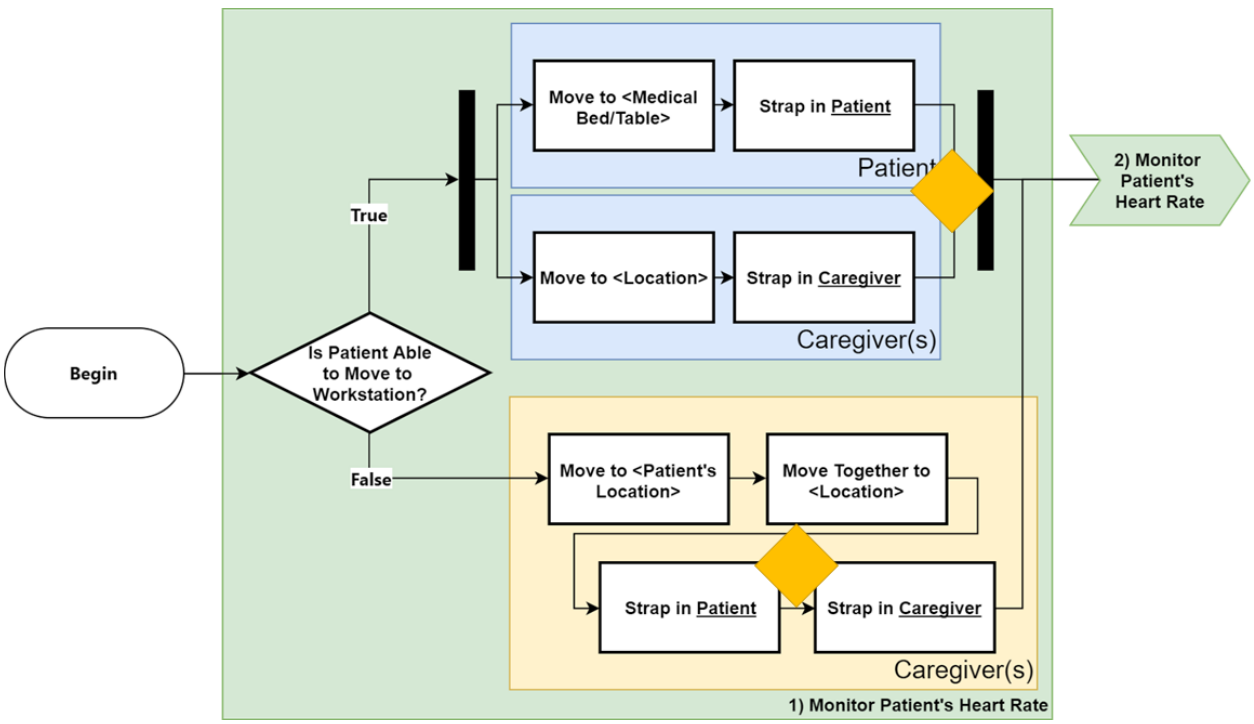

Figure 5. Flow Chart of the Starting Points to begin Metric Data Collection.

The user will then be able to easily manipulate the data through the use of a graphical user interface. It is from this GUI that visualizations of the data will be generated. For example, if the user was interested in the total amount of time that a series of tasks took to complete, the breakdown of each metric that encompasses total time can be seen and compared between workstation designs in Figure 6 as shown below.

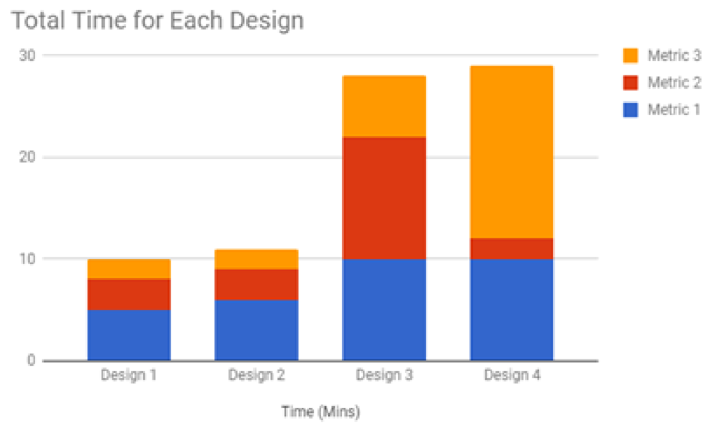

Figure 6. Graph of Metrics Encompassing the Total Time for a Series of Tasks.

CONCLUSION

The Capstone Team understands that NASA's engineering process requires support for evaluating medical workstation designs. While NASA has the capability of reaching out to SMEs to assist in evaluating designs, a study of the system's behavior over time is necessary to gather information about dynamic performance measures. Therefore, the Capstone Team has provided a process and solution to allow NASA to improve their ability to evaluate the workstation design. The model provided is flexible – for it allows NASA to build representative scenarios with processes that can represent internal tasks. NASA can then pinpoint critical information that is needed for metrics belonging to dynamic performance measures.

ACKNOWLEDGEMENTS

The Capstone Team would like to thank Dr. Jim Leathrum and Dr. Roland Mielke from ODU's Department of Modeling, Simulation, and Visualization Engineering for their guidance and support. The team would also like to thank Dr. Kara Latorella of ExMC at NASA-LaRC for providing this opportunity as well as serving as a point of contact and as a subject matter expert.

REFERENCES

Choi, B. K., & Kang, D. (2013). *Modeling and Simulation of Discrete-Event Systems*, Hoboken, NJ: John Wiley & Sons.