A MODULAR AND EXTENSIBLE ARCHITECTURE INTEGRATING SENSORS, DYNAMIC DISPLAYS OF ANATOMY AND PHYSIOLOGY, AND AUTOMATED INSTRUCTION FOR INNOVATIONS IN CLINICAL EDUCATION

by

Douglas Allen Nelson Jr.

B.S.E., Bioengineering, University of Pittsburgh, 2009

B.S., Applied Mathematics, University of Pittsburgh, 2009

Submitted to the Graduate Faculty of

Swanson School of Engineering in partial fulfillment

of the requirements for the degree of

Doctor of Philosophy

University of Pittsburgh

2017

UNIVERSITY OF PITTSBURGH

SWANSON SCHOOL OF ENGINEERING

This dissertation was presented

by

Douglas Allen Nelson Jr.

It was defended on

December 15, 2016

and approved by

Sanjeev G. Shroff, PhD, Distinguished Professor and Gerald McGinnis Chair, Department of Bioengineering, Swanson School of Engineering, University of Pittsburgh

Patrick J. Loughlin, PhD, Professor, Department of Bioengineering, Swanson School of Engineering, University of Pittsburgh

John M. O'Donnell, DrPH, RN, CRNA, MSN, Professor and Chair, Department of Nurse Anesthesia, School of Nursing, University of Pittsburgh

Dissertation Director: Joseph T. Samosky, PhD, Assistant Professor, Department of Bioengineering, Swanson School of Engineering, University of Pittsburgh

A MODULAR AND EXTENSIBLE ARCHITECTURE INTEGRATING SENSORS, DYNAMIC DISPLAYS OF ANATOMY AND PHYSIOLOGY, AND AUTOMATED INSTRUCTION FOR INNOVATIONS IN CLINICAL EDUCATION

Douglas Allen Nelson Jr., PhD

University of Pittsburgh, 2017

Adoption of simulation in healthcare education has increased tremendously over the past two decades. However, the resources necessary to perform simulation are immense. Simulators are large capital investments and require specialized training for both instructors and simulation support staff to develop curriculum using the simulator and to use the simulator to train students. Simulators require staff to run the simulator, and instructors must always be present to guide and assess student performance. Current simulators do not support self-learning by students. Thus, the expensive simulators sit idle most of the day. Furthermore, simulators are minimally customizable, resulting in programs often being required to purchase simulators that have more functions and features than needed or that cannot be upgraded as needs change.

This dissertation presents the development of BodyExplorer, a system designed to address limitations of current simulators by reducing the resources required to support simulation in healthcare education, enabling self-use by students, and providing an architecture to support modular and extensible simulator development and upgrades. This dissertation discusses BodyExplorer's initial prototype design, integration, and verification, as well as development of the modular architecture for integrating sensors, dynamic displays of anatomy and physiology, and automated instruction.

Novel sensor systems were integrated to measure user actions while performing (simulated) medication administration, cricoid pressure application, and endotracheal intubation on a simulation mannequin. Dynamic displays of anatomy and physiology, showing animations of breathing lungs or a beating heart, for example, were developed and integrated into BodyExplorer. Projected augmented reality is used to show users underlying anatomy and physiology on the surface of the mannequin, allowing users to see the internal consequences of their actions. An interface for supporting self-use and showing additional views of anatomy was incorporated using a mobile display. Using the projected images, mobile display, and audio output, a virtual instructor was developed to provide automated instructions to users based upon real-time sensor measurements.

Development of BodyExplorer was performed iteratively and included feedback from end-users throughout development using user-centered design principles. The mixed-methods results from three usability testing sessions with end-users at two academic institutions will be presented, along with the rationale for design decisions that derived from the results. Built upon feedback received during usability testing, the results from two scenarios of automated instruction will be provided by demonstrating examples of learning to apply cricoid pressure and learning to administer (simulated) medications to control heart rate. Discussion will also be provided regarding how the automated instruction techniques can be extended to provide training in other healthcare applications.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

NOMENCLATURE

Throughout this document, the following abbreviations will be used:

AR … Augmented Reality

AI … Automated Instruction

CASE … Comprehensive Anesthesia Simulation Environment

CELTS … Computer-Enhanced Laparoscopic Training System

CGI … Computer-Generated Imagery

CI … Confidence Interval

CDC … Centers for Disease Control and Prevention

CPR … Cardiopulmonary Resuscitation

DLP … Digital Light Processing

ECG or EKG … Electrocardiogram

FBM … Full-Body Mannequin

FFPD … Free Form Projected Display

FSM … Finite State Machine

FSR … Force-Sensitive Resistor

GAS … Gather, Analyze, and Summarize

HMD … Head-Mounted Display

HPS … Human Patient Simulator

IOM … Institute of Medicine

IV … Intravenous

IR … Infrared

IRB … Institutional Review Board

LCD … Liquid Crystal Display

LED … Light-Emitting Diode

MIS … Minimally Invasive Surgery

MRI … Magnetic Resonance Imaging

PC … Personal Computer

PCB … Printed Circuit Board

PI … Principal Investigator

PTT … Partial-Task Trainer

RFID … Radio-Frequency Identification

RSI … Rapid Sequence Induction

SBME … Simulation Based Medical Education

SBS … Screen-Based System

SD … Standard Deviation

SME … Subject Matter Expert

SRNA … Student Registered Nurse Anesthetist

TUI … Tangible User Interface

UCD … User-Centered Design

USB … Universal Serial Bus

PREFACE

It takes a village…to raise a BodyExplorer. I am forever grateful to all who have contributed at various stages along the way, dating back to bioengineering senior design projects completed just over 8 years ago (individual contributions will be named later in this dissertation).

It was about this same time 8 years ago that I approached Dr. Joseph Samosky, then my senior design project mentor, about continuing his mentorship as I pursued a PhD. Joe, thank you for saying "yes" 8 years ago and for your continued support on my academic journey. Through your guidance and in the lab, I have learned much more than research methodology and systems design, and I anticipate those additional "pearls" will be incredibly valuable as I move forward in my career and beyond.

It was about this same time 4 years ago that Dr. Sanjeev Shroff, Dr. Patrick Loughlin, and Dr. John O'Donnell said "yes" to providing their guidance on my committee. Drs. Shroff and Loughlin, it has been my honor to come through the program as an undergrad, soaking up all the knowledge you had to share, to having collegial debates with you on system design, research methodology, and data analysis. John, thank you for adding your clinical and educational expertise to the team. Without your support, the support of the Pitt School of Nursing, and the support of your students, curriculum design and usability testing would not have been possible.

Usability testing was strengthened through outside collaboration with colleagues from the Robert Morris University School of Nursing. I am thankful for the support and collegiality from Dr. Suzie Kardong-Edgren and Jan Barber throughout usability testing at RMU.

I am profoundly grateful for the financial support provided to me and towards advancing this work from the Pitt Departments of Bioengineering, Anesthesiology, and Nurse Anesthesia, the Swanson School of Engineering and the School of Nursing, a TATRC Grant from the US

Army, a Coulter Translational Research Award, a Clinical & Translational Science Institute Award, and Pitt's Innovation Institute.

I would like to thank my dear friend and fellow PhD student, Salim Olia, for his unending support, including critiquing early design ideas and research methodology, assisting with transport of BodyExplorer for usability testing, replacing me for system demos when I was unavailable, being my writing accountability partner, and for providing multiple reviews of this dissertation. I greatly appreciate your friendship, in addition to our home improvement therapy sessions!

To my parents, Doug and Ann Nelson, thank you for your never-ending love and support—even when my schooling seemed to never end! You have taught me many things that I could not have learned directly through school alone; how to love, how to treat others, how to live, and how to lead.

To my wife, Courtney Kase, your commitment and support throughout this journey has been unwavering. Though it sounds cliché, I truly could not have completed this dissertation without you. Your emotional support was uplifting when I was rundown. You have supported us financially while I finished the degree. Your guidance on mixed-methods research helped provide me with direction for video analysis. And I swear, this is the last time I'll ask you to read my dissertation…

1.0 INTRODUCTION

The Institute of Medicine's (IOM) 1999 report, *To Err is Human*, suggested that the prevalence of medical errors which lead to patient death, and the costs of such errors to society was a significant and alarming public health issue. A more recent report by James in 2013 published even higher mortality estimates. James points out that if these rates are correct, medical error is the third leading cause of death in the United States. While preventing future medical errors altogether will require a multifaceted, interdisciplinary approach, and possibly even cultural changes in healthcare, simulation as a part of healthcare education shows promise towards reducing medical errors. Flight simulation for aviation is often looked upon as a model for healthcare simulation. While there are many lessons to learn from flight simulation's success, healthcare simulation, in general, is much more complex because we are still learning many new things about the human body and best practices for care. As the methods for treating patients with newer medical technologies and procedures advances at a rapid rate, so too must the methods and technologies that we use to educate healthcare providers.

1.1 THE IMPERATIVE FOR HEALTHCARE SIMULATION

The IOM report in 1999 estimated that about 100,000 people in the US die each year due to preventable medical errors [1]. This report also estimated that these preventable medical errors

result in costs between $17 and $29 billion annually, with about half of these costs being directly associated with additional healthcare needs necessitated by medical errors. One flaw with the IOM report was that it was based on retrospective data and thus criticized as potentially inaccurate. A more recent estimate by James (2013) of mortality associated with preventable medical errors suggests that the true range (derived from a variety of sources) is 200,000 and 400,000 annually in the US [2]. Comparing these new estimates with 2012 data reported from the Center for Disease Control and Prevention (CDC), preventable medical error ranks only below heart disease (599,711) and cancer (582,623), making preventable medical error the third leading cause of death in the US.

One year before the IOM report was published, an article was published describing the *Imperative for Medical Simulation* [3]. In the article, Dawson and Kaufman describe the success of flight simulation in aviation and the need for such success in medical education. Pilots can learn how to fly a plane in a simulator before flying a 75-ton aircraft and without putting their own life, the lives of passengers, and the expensive aircraft at risk. The authors argue that the master-apprentice learning model in medicine is inefficient, and that the use of patients as the primary teaching material for developing clinical skills involves significant patient risk and is inefficient with respect to allocation of time and money. They posit that the medical education field could learn from and build upon the efficiencies in cost and safety that simulation in aviation has achieved. Furthermore, the authors argued that simulation can be used to certify and credential physicians to standardize and objectively quantify the procedural skills of physicians as part of the credentialing process.

Adding to Dawson and Kaufman's argument for simulation in healthcare, Ziv et al wrote a paper suggesting that simulation-based medical education (SBME) is superior as a training

method and also represents an **ethical imperative** for faculty and educational administrators [4]. These authors highlighted the dilemma of using real patients as teaching material for future clinicians in training. They concluded that training clinicians must include experience with live patients balanced with a competing responsibility to provide efficient, high-quality, and safe care for those patients. Ziv et al recommend that simulation be used as part of the first training experiences which allows clinicians-in-training to be more prepared and experienced when they progress to treating live patients [4].

1.2 THE IMPERATIVE FOR INNOVATIONS IN HEALTHCARE SIMULATION

Modern-day healthcare simulation has roots in the early 1900s when healthcare providers began training with individual-task trainers, known today as partial-task trainers (PTT), for inserting needles into fake arms. In the mid-1900s, PTTs for learning cardiopulmonary resuscitation (CPR) was a precursor for development of full-bodied mannequins (FBM)—full body versions of plastic human bodies [5, 6]. Today, with the advent of minimally-invasive surgeries (MIS) and robotic surgeries, specialized simulators have been designed to address training for those procedures. A more in-depth history of FBMs, in addition to the current state of the art within simulation education technologies, will be provided in Sections 2.1 and 2.2.

Historically, simulation technologies have been reactive to the emergence of new medical technologies, procedures and clinical best-practices. The Society of American Gastrointestinal and Endoscopic Surgeons (SAGES) and others within the MIS community worked for 20 years to create and validate simulation technologies as a cognitive-psychomotor competency examination mandatory for gastroenterologists' licensure after MIS entered the surgical suite [7]. Pioneers like

Dr. Richard Satava, Dr. David Gaba, and many others, paved the way for other simulation technologies to become validated training tools at a time when simulation was still struggling for global acceptance as a standard within healthcare education [8, 9]. Now, simulation has become a required component for licensure in areas of MIS through programs such as the Fundamentals of Laparoscopic Surgery and the Fundamentals of Endoscopic Surgery [10, 11]. Similarly, the American Board of Anesthesiology has recognized simulation as an integral part of their Maintenance of Certification in Anesthesiology Program (MOCA) [12].

Over the past decade, studies across multiple healthcare disciplines and specialties have shown support for simulation as a viable supplement or replacement, in some cases, for traditional classroom or clinical activities. The National League of Nursing (NLN) sponsored a 3-year longitudinal study of simulation in pre-licensure nursing curricula where simulation experiences replaced 25% and 50% of traditional clinical experiences [13]. The results supported the hypothesis that simulation could, in some instances, replace traditional clinical experiences while still successfully achieving educational outcomes. These results, in addition to the increasing demand for clinical education sites and clinical facilitators, have led to some states, including Pennsylvania, to permit some of the mandated clinical hours for pre-licensure nursing students to be completed with simulation.

As simulation education methods become more pervasive, the challenges around implementing simulation become even more apparent. Similar to new advancements in healthcare technologies and procedures requiring new training methods for use of those technologies and procedures, advancements in training technologies, too, require the educators to learn how to use the new technologies. Operating simulation technology, specifically computer-controlled FBMs, requires advanced training. Instructors and operating technicians must learn how to program the

FBM to respond appropriately (and in a consistent manner) during student interactions. Companies that sell the FBMs also often sell service and training support alongside the technology for thousands of dollars on top of the initial FBM investment. As the method has penetrated to all areas of healthcare education, required training programs have emerged. Due to the required training, institutions will often have a team of simulation experts consisting of both faculty and staff to support the educational efforts. As new faculty and staff join the existing teams of simulation specialists, significant ongoing monetary and time resource investments are required.

Additionally, computer-controlled FBMs can cost between $30,000 and $300,000. These FBMs require specially designed suites with rooms for simulated patient care of the FBM and separate areas where trained operators remotely control the mannequin during a scenario. Initial startup costs for a simulation suite are estimated to be around $1M, with fixed annual costs over $250,000 [14].

1.3 THESIS STATEMENT

Through iterative user-centered design, this thesis demonstrates methods for providing automated instruction using a modular architecture that has the potential to support self-learning and also reduce the resources required to support simulation in healthcare education.

1.4 CONTRIBUTIONS

I have worked with Dr. Samosky in the Simulation and Medical Technology Research and Development Center (SMTRDC) at the University of Pittsburgh for 8 years, beginning with my undergraduate senior design project in 2008. Throughout my time in the SMTRDC, Dr. Samosky and I have collaborated on many projects to innovate in the field of healthcare simulation and smart medical device design. Throughout my work on those projects, I have had the opportunity to meet with various clinical providers in university schools and teaching hospitals, both locally and across the country, to discuss and understand needs and gaps that our technology could address and fill. These discussions have guided my development efforts presented in this dissertation.

Building upon my experiences in the SMTRDC which encompass healthcare simulation and initial collaborative work designing prototype systems that will be discussed later in this dissertation, I pursued the design of BodyExplorer as the core of my dissertation. I contributed to the development and evolution of BodyExplorer since the earliest prototyping. Through my dissertation work, I have focused on many aspects of the development, including software design and programming, algorithm design, hardware design and fabrication, system verification in bench-top testing, and system validation via end-users' evaluation.

During development of all of the initial prototypes, I contributed to the software design and programming, initially in LabVIEW and recently in C# and Unity. During development of the prototype for identifying simulated medications, I designed the algorithm that identified the simulated medications based upon measured conductivity, while colleague Brandon Mikulis designed the initial electronic circuit for measuring the conductivity. During development of the prototype for BodyWindows after initial work by Dr. Samosky, a student design team including Emma Baillargeon, Andrew Brown, Amy Chaya, Leah Enders, and Ali Sukits, and expanded work

by Russel Bregman, Bo Wang, and Robert Weaver, I integrated the prototype for identifying simulated medications with the prototype for opening anatomy windows on the mannequin, which allowed injected medications to drive an animation of a beating heart based upon whether the injected medication was a stimulant or inhibitor. In the most recent work related to system redesign, I modified the electronic circuit for measuring the conductivity and redesigned the conductivity sensor itself.

I was an advisor to Elwin Lee on development of the initial bench-top cricoid pressure application prototype, and I also provided software development support. I worked with colleague Andrew Hosmer and other Art Institute of Pittsburgh interns to create the physical model of the cricoid and thyroid cartilage and the silicone neck skin that was integrated into the mannequin's neck. I designed and fabricated all the electronics for detecting touch and applied cricoid pressure when the system was incorporated into the mannequin.

I worked again with Andrew Hosmer to design the mounting apparatus for the Hall-effect sensors used in detecting location and depth of endotracheal intubation. I designed and fabricated all of the electronics for detecting location and depth of endotracheal intubation when the original prototype from 2008 was integrated with the other prototypes. I also modified the detection algorithm and electronics to accommodate detection with only 4 Hall-effect sensors on the trachea instead of 7 sensors previously.

When efforts began to integrate the prototypes for medication administration, cricoid pressure application, endotracheal intubation, and anatomy windowing, I reworked all of the sensor and data acquisition hardware to fit inside of the mannequin. I designed multiple iterations of printed circuit boards, assembled the boards, and integrated all of the electronics into the

mannequin. I also wrote the software in LabVIEW for accommodating the integration of all of the prototypes.

To support continued and efficient software development, and for reasons discussed in this dissertation, I chose the Unity development environment and C# programming language to replace LabVIEW. I spearheaded the efforts to transition the software from LabVIEW into Unity and C#, and was fortunate to have 8 months' support from two programming interns, Ryan Rightmer and Hunter Nebbia, to assist with the transition. I worked with Andrew Hosmer and digital artist, Jeremy Tabor, to prepare the 3D anatomy models for integration into Unity. I also worked with Jeremy to iteratively create animations of breathing lungs and beating heart that were programmatically controllable by C# scripts that I programmed in Unity. Learning animation through these experiences, I integrated Jeremy's model of an endotracheal tube with animated cuff inflation to pass through the 3D model of the trachea, under programmatic control in Unity.

After system integration, I designed, conducted, and analyzed the results of the verification experiments for the cricoid pressure and medication administration sensor systems. For the verification of the cricoid pressure sensor system, I designed a loading platform assembly to apply repeatable loads to the mannequin's neck during testing.

During system validation with end-users, I designed and conducted the usability testing sessions which included: communication with multiple universities' IRB offices and receiving Human Subjects Protection application approvals; training research assistants and data coders; conducting the testing sessions and data analyses using quantitative and mixed methods design; and modifying the system based on findings. During system validation, I had the opportunity to train and work with research assistants from the University of Pittsburgh School of Nursing and School of Arts and Sciences, including Dane Brinley, Nicholas Grant, Alison Henry, Troy Lunt,

Alyson Mesisca, Amanda Roll, Dana Zelder, Connor O'Donnell, Maeve O'Donnell, Jennifer Ruminski, Kathleen Hamel, Ryan Dubowey, and Elizabeth Reiner.

1.5 OVERVIEW

The following chapters of this dissertation discuss the processes and outcomes of BodyExplorer design and development, from initial prototype design and integration of subsystems and learning modules through usability testing with representative end-users, concluding with demonstrations of the automated instruction capabilities.

2.0 BACKGROUND AND SIGNIFICANCE

This chapter presents the background and significance for simulation in healthcare. It also provides a description of the state of the art in commercial simulator technology and research-prototype systems. Finally, the chapter provides the rationale for BodyExplorer development.

2.1 A BRIEF HISTORY ON FULL-BODY MANNEQUIN DEVELOPMENT

Scholarly works presenting the history of simulation in healthcare have previously been published, [6, 15-17]. In this section, I will present the history as it relates to innovations in mannequin-based simulators and how that history relates to BodyExplorer developments.

Modern-day simulation technologies originated in the 1960s with the development of a training mannequin for CPR—the Resusci-Anne. The Resusci-Anne was developed in Norway by a toy manufacturer, Asmund Laerdal. The development was inspired, in part, by work from Dr. Peter Safar, who was a pioneer of resuscitation and began the University of Pittsburgh's Anesthesiology Department. While the Resusci-Anne allowed training for CPR, it was a passive system that did not have a method for recording a user's performance, computerized control, or performance feedback. [18]

In 1966, Dr. Judson Denson and Dr. Stephen Abrahamson from the University of Southern California built the first computer controlled mannequin, SimOne [19]. SimOne's development was funded by the United States Department of Education over three years for $272,130 [20]—about $2M in 2014 dollars after adjusting for inflation from 1965 [21]. SimOne (**Figure 1**) was in

many ways ahead of its time; the mannequin had the shape of a human body, it had realistic chest movement to simulate breathing, it had constricting and dilating pupils in response to light and could blink its eyelids, it was controlled by a computer, and it could respond automatically to up to 10 simulated medications being injected into an intravenous (IV) port [19, 22].

Figure 1. SimOne
(adapted by permission from BMJ Publishing Group Limited. [15])

SimOne, despite its early successes at showing educational impact and gaining national public recognition [19, 23], was not accepted by the medical field as a viable means for training clinicians.

SimOne was too expensive to implement, and medical educators were resistant to modifying the apprenticeship model to include training with a simulation mannequin [15].

Over the next few decades, simulators were developed, initially, at academic institutions before they would be commercialized. The Cardiology Patient Simulator, or Harvey, (**Figure 2**) was developed by Dr. Michael Gordon at the University of Miami in the 1970's [24, 25]. Harvey provides highly realistic normal and pathologic heart and lung sounds as well as other physical examination representations. Harvey has been iteratively refined and remains a commercially available product.

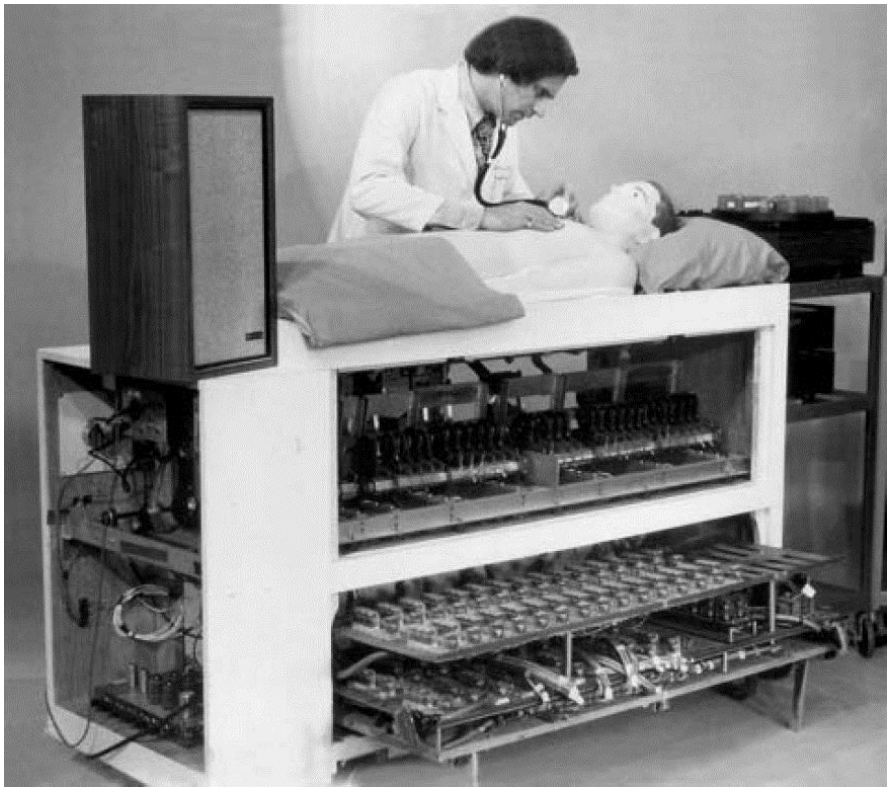


Figure 2. Cardiology Patient Simulator, aka Harvey
(adapted by permission from BMJ Publishing Group Limited. [15])

The Gainesville Anesthesia Simulator was a computer-controlled, full-body mannequin developed under the direction of Dr. Michael Good and Dr. J S Gravestein at the University of Florida in the 1980's [26]. The Gainesville Anesthesia Simulator (**Figure 3**) was originally designed to allow anesthesia residents to develop clinical skills in a safe environment, while also allowing more advanced clinicians to practice difficult anesthesia cases [16]. The simulator was later known as the Loral Simulator and was then purchased by the Medical Education Technologies, Incorporated (METI) which marketed the later iterations as the METI Human Patient Simulator (HPS). Purchased by CAE Healthcare, this simulator is still commercially available today as the CAE/METI HPS [6]. The HPS simulator has all the previously mentioned features of SimOne, in addition to other features, including physiological and pharmacological models, and an add-on package for recognizing administered anesthesia gases.

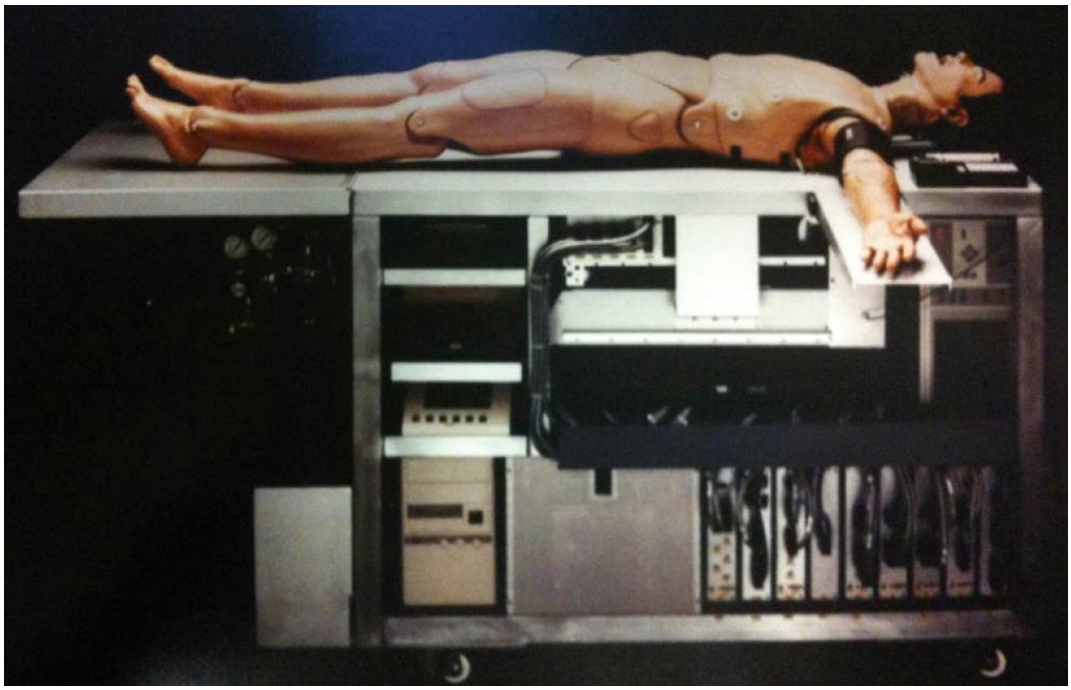


Figure 3. Gainesville Anesthesia Simulator
(republished with permission conveyed through Copyright Clearance Center, Inc. from [6])

The Comprehensive Anesthesia Simulation Environment (CASE) was developed under the direction of Dr. David Gaba at Stanford University in the 1980's at the same time that the Gainesville Anesthesia Simulator was being developed [27]. The CASE simulator (**Figure 4**) incorporated numerous PTTs available at the time and added some custom hardware and software to design a simulator for recreating the anesthesia environment in the operating room. The CASE simulator also supported team training between the anesthesiologist, surgeon, and nurses. The system went through several commercialization efforts and became known as the CAE/Eagle/MedSim Simulator. It was purchased by an Israeli ultrasound simulator company and commercialization was eventually abandoned.

Figure 4. Comprehensive Anesthesia Simulation Environment
(republished with permission conveyed through Copyright Clearance Center, Inc. from [6]
© 1986 David M. Gaba, MD)

In the mid-1990s, Laerdal advanced their initial work on Resuci_Anne with support from researchers at the University of Pittsburgh on airway design and developed SimMan—"a substantially cheaper…higher fidelity mannequin simulator [15]." The SimMan 3G (current, 3rd generation, **Figure 5**)—replicates all of the functions of the early SimOne in a FBM, as well as functions afforded by widespread use of computers such as wireless control from the simulator operating room or a wireless, handheld tablet. However, SimMan 3G still requires control from a human operator and cannot provide instructions to those using the system without an instructor.

Figure 5. SimMan 3G

There are a few interesting points to glean from the development of mannequin simulators. Much of the development work for mannequin simulators derived from work that began at academic institutions. Each of the highlighted early simulators were developed in isolation without collaboration from other colleagues developing in the same areas at the time, and without leveraging work that was previously done in the field. Newer FBMs have leveraged early work

from FBM development with some advancement to the technology, but newer FBMs still are dependent on significant human intervention to operate and provide instruction.

Since the initial development of FBMs, there has been an explosion of development in other technologies for simulation. Those developments will be highlighted in the following section.

2.2 THE STATE OF THE ART

Commercially available technologies for SBME span a spectrum of complexity and physical functionality [28, 29]. These technologies are typically classified into three main categories: 1) Screen-based systems (SBSs), 2) Partial-task trainers (PTTs), and 3) Full-body mannequins (FBMs). The following sections will describe technologies in each of these categories while outlining specific advantages and disadvantages.

2.2.1 Screen-Based Systems (SBSs)

SBSs are used to teach a variety of cognitive skills and topics to students [30]. Traditionally, SBSs utilize the mouse and keyboard for student interaction (**Figure 6** and **Figure 7**). Newer SBSs introduce a more hands-on approach to student interaction utilizing touch-screen systems for virtual dissection, such as the Anatomage Table [31], and procedural training scenarios, such as those found in ArchieMD's Medrills [32]. SBSs have many benefits to student training, including:

1. they reside in a computer and can thus be widely deployed,

2. students can interact with human anatomy or physiology in a safe way without worrying about safety precautions and regulations associated with working on cadavers or real patients,
3. students can manipulate images on-screen to gain different perspectives, and
4. all students can learn with the same experiences because learning modules can be standardized regardless of the time and location when a student used the SBS.

While SBSs offer these benefits, they often lack the richness of hands-on training in which actual medical tools are used by the student. Furthermore, they are often utilized for static demonstration and tutorial rather than a dynamic simulation that can be freely manipulated by the student.

Figure 6. Anatomy & Physiology Revealed (photo courtesy of McGraw-Hill Higher Education)

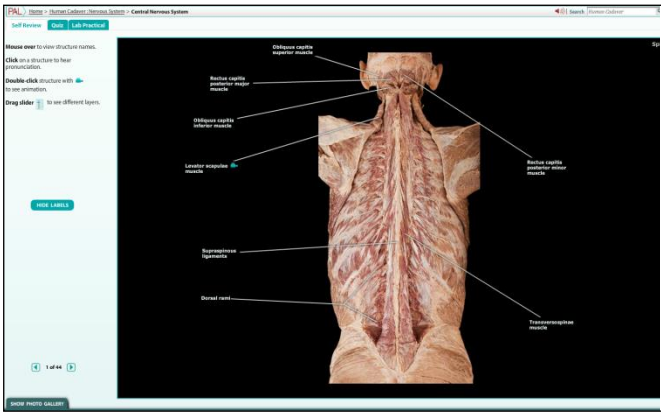


Figure 7. Mastering Anatomy & Physiology (photo courtesy of Pearson Higher Education)

2.2.2 Partial-Task Trainers (PTTs)

PTTs provide hands-on psychomotor training for specific medical interventions in isolation; some examples include catheter insertion (**Figure 8**), suturing (**Figure 9**), tracheal intubation (**Figure**

10), or ultrasound training (**Figure 11**) [33]. Some PTTs have embedded sensors for measuring student interactions with the system, while others do not have sensors and rely solely upon instructors to guide student interactions and judge proficiency.

Figure 8. Central Venous Cannulation
(photo courtesy of Kyoto Kagaku America, Inc.)

Figure 9. Subcuticular Suturing Trainer
(photo courtesy of Simulab Corporation)

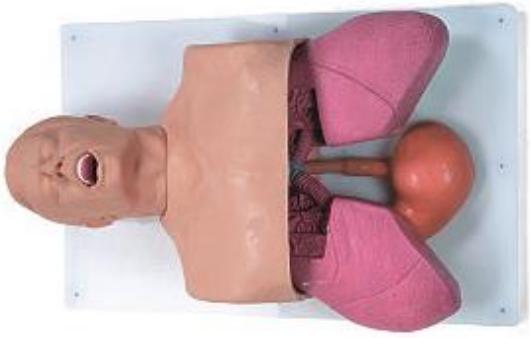


Figure 10. Airway Management Trainer
(photo courtesy of Simulaids, Inc.)

Figure 11. Vimedix Ultrasound Trainer
(photo courtesy of CAE Healthcare, Inc.
© CAE Healthcare, 2017)

Some PTTs for MIS incorporate strategies from SBSs to show anatomical structures, utilize sensors for measuring student interactions, and encompass an intelligent instructional algorithm for providing students with guided feedback in response to student actions (**Figure 12** and **Figure**

13) [34]. These advanced PTTs for MIS are advancing towards automated training systems where students can learn on their own while still receiving guided feedback.

Figure 12. Laparoscopic Simulator, LAP Mentor (photo courtesy of Simbionix, Inc.)

Figure 13. Arthroscopic Simulator, ARTHRO Mentor (photo courtesy of Simbionix, Inc.)

These systems can measure student actions with the interface, but the simulation is constrained. In an open-ended scenario, such as open, invasive surgery or other medical procedures outside of MIS such as rapid sequence induction of anesthesia, the method for automatically measuring student interactions within PTT simulation becomes difficult. Furthermore, PTTs are not able to be combined for advanced, multi-task training scenarios.

2.2.3 Full-Body Mannequins (FBMs)

FBMs have the potential to provide a platform for simple skill training to integrated, multi-task training [33]. FBMs can be passive for training of patient positioning or manipulation, or can be

computer-controlled for more advanced skills training. FBMs are amenable to individual student training, or training a team of students working together. While using computer-controlled FBMs, students can be presented with patient vital signs, perform CPR or intubation, administer an electrical shock to the patient using a defibrillator, or even administer medications (**Figure 14** and **Figure 15**). Some specialized FMBs can even give simulated birth to a baby (**Figure 16**). A technician can use the connected computer to update the FBM's vitals or reactions in response to observed student interactions with the FBM.

Figure 14. SimMan 3G (Copyright © Laerdal Medical (**http://www.laerdal.com**). All rights reserved. Used with permission

Figure 15. CAESAR (photo courtesy of CAE Healthcare, Inc. © CAE Healthcare, 2017)

Figure 16. Noelle (photo courtesy of Gaumard Scientific Company, Inc.)

While FBMs can be controlled by a computer, there are limitations to that control. FBMs are limited in their ability to automatically and instantaneously respond to student interactions. Some FBM control software allows the mannequin to have predefined trends for updating vital signs in response to student interactions, but these trends typically must be initiated by an instructor or technician once the instructor or technician observe the specific student interaction to begin the trend. FBMs are also limited in that they are not able to show internal anatomy or physiology that provides students with insight into underlying physiologic function, aside from traditional monitoring methods such as electrocardiography (ECG or EKG) or oxygen saturation (SpO_2). Showing students the internal anatomic and physiologic consequences of their external actions

during training, in addition to the traditional monitoring methods, may provide students with better mental models of physiologic function that could translate to improved bedside care.

2.2.4 Limitations to Innovation

As Bill Buxton, renowned user experience designer and researcher, states, "Everything is best for something and worst for something else [35]." Each of these categories of simulators has their own specialized role within SBME. However, integration of tools across categories could allow increased learning potential in clinical education. A PTT's intubation trainer could be a more accurate model than what is presented within a FBM. It could be advantageous for learning to be able to swap out the FBM's intubation model with the PTT's intubation model. Yet, this integration is not feasible when hardware and software interfaces are not standardized across simulation technologies. We may look to standards developed for computers, such as the universal serial bus (USB), as an example for what we might expect from future technologies within SBME offering plug-n-play functionality across and between simulation technologies.

The ability for innovators to expand functionality of existing simulators by adding new software or hardware is limited. Companies often shield the underlying proprietary software or hardware from end-user modification. If a researcher develops an improved model for recognizing the injection of simulated medications, that improved model cannot be integrated and used within a preexisting FBM by the researcher due to these proprietary barriers.

Existing FBMs provide some opportunity for customization. Instructors are able to create new training scenarios using simulator-specific software for FBMs. There is even a third-party market for buying and selling preprogrammed training scenarios for FBMs [36]. However, adding

the costs for FBM, simulator-specific software licenses, and scenario-use licenses ($40-$180 per individual scenario [36]) is a barrier for widespread adoption [37, 38].

To address these limitations, a low-cost, modular system that allows open access to hardware and/or software modifications will promote rapid innovations in SBME, widespread adoption, and expand the possibilities for interdisciplinary training. Without such a system, necessary innovations to simulation technology will not be able to advance rapidly enough to meet the needs of healthcare educators.

2.3 POSITIVE OUTCOMES OF SIMULATION-BASED MEDICAL EDUCATION

Simulation has been reported to be an effective means for training clinicians in multiple domains [39-42]. Simulation interventions are more effective than no intervention at all [43], more effective when compared to non-simulation interventions such as didactic lectures [44], and have been cross-compared in order to develop best practices within SBME [45]. Simulation has even been reported to be an effective replacement option for up to 50% of actual clinical experiences during undergraduate nursing education [13]. Some studies have reported positive patient outcomes associated with simulation-based training interventions, predominantly within the domains of airway management, gastrointestinal endoscopy, and central venous catheter insertions [39].

Not all studies have shown positive outcomes within SBME. When comparing problem- or case-based learning sessions and mannequin based simulation sessions, the problem- and case-based learning sessions were as effective, if not more effective, than the mannequin based sessions [46, 47]. These results show that merely adding any type of simulation into a curriculum is not necessarily beneficial. It is necessary to understand what teaching tools to use and when to use

them, which agrees with Buxton's primary axiom, and the conclusions by Maran and Glavin [28, 35] which state that choosing the proper tool to use for the given setting is vital to educational success with that tool. Every tool may be good for teaching something, but bad for teaching something else.

2.4 PRIOR ART: TANGIBLE USER INTERFACES

Tangible user interfaces (TUIs) were originally defined by Ishii and Ullmer in 1997 as "augment[ing] the real physical world by coupling digital information to everyday physical objects and environments." [48] The authors were inspired by others during that decade who had visions of ubiquitous computing [49] and computer augmented environments [50]. A primary distinction between traditional computing and the ideas presented by Ishii, Ullmer, Weiser, and Wellner, et al during the 1990s was that computer media should not be confined to a screen on a desktop; rather, the media should emerge from the screen and seamlessly blend into the real physical world and the user's environment. Moreover, input to control the computer media should also not be confined to the primary inputs of the mouse and keyboard; instead, the inputs should support every day, tangible objects to be used as inputs, such as physical blocks, paper, toy cars, or other instruments.

Examples of TUIs exist today within healthcare simulation technologies, mostly within the category of PTTs. Examples of TUIs in healthcare simulation were described previously in **Section 2.2.2**, and include the Vimedix Ultrasound Trainer [51] and the Arthroscopic Simulator [52]. Other research prototypes of TUIs within healthcare simulation will be described in the next section.

2.5 RATIONALE FOR BODYEXPLORER DEVELOPMENT

2.5.1 Augmented Reality in Healthcare Simulation

BodyExplorer is a TUI that also falls within the category of Augmented Reality (AR). AR is a mixing of computer-generated imagery of objects (i.e., virtual objects) within the real-world environment. In a survey of AR, three characteristics were used to define AR systems regardless of what technology the system utilized to implement AR. The characteristics state that an AR system:

1. Combines real and virtual.
2. Is interactive in real time.
3. Is registered in three dimensions.[53]

Prior AR developments for healthcare have focused on directly supporting medical interventions, facilitating communication between doctors and patients, and providing novel training methods within SBME. AR has been utilized for surgical planning and image guided interventions using a variety of techniques [54], including image overlays [55-58], head-mounted displays (HMDs), and direct projection [59, 60]. An AR system was developed that projected anatomical images down onto a patient's body in order to facilitate communication between doctors and patients [61]. Systems have also been developed using AR for training how to perform clinical breast exams (**Figure 17**) [62-65], applications involving depth perception for minimally invasive procedures (**Figure 18**) [66], and using free form projection displays to perceive anatomy in 3D on a mannequin surface (**Figure 19**) [67, 68].

Figure 17. Clinical Breast Exam Simulator
(© 2009 IEEE. Reprinted, with permission, from [63])

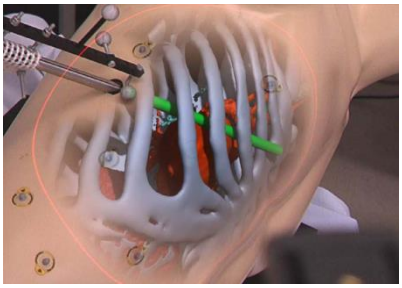


Figure 18. Contextual Anatomic Mimesis
(© 2007 IEEE. Reprinted, with permission, from [66])

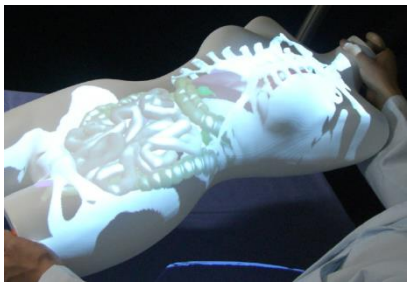


Figure 19. Free Form Projection Display
(© 2008 Association for Computing Machinery, Inc. Reprinted by permission. [68])

These previous developments provided a basis for AR use in healthcare, and among those specifically addressing education, each development pioneered new methods in AR for SBME. However, these can be improved for broader application and ease-of-use by the learner. The work by Kotranza, Lind, Deladisma, Pugh, and Lok [62-65] provided visual feedback to a learner while he or she practiced performing a clinical breast exam on a hybrid simulation mannequin and breast exam task trainer. This methodology merged FBMs, PTTs, and SBSs. In addition to the hybrid approach, the benefits of their methodology were:

1. real-time measurement of the user's applied force during the breast exam,
2. quantification of the total path and area of the breast examined, and
3. real-time visualizations provided to the user based upon his or her actions during the exam.

While the visualizations were tied directly to the user's actions, they were displayed to the user on an adjacent LCD monitor that merged the physical breast exam model and the user's hand with a virtual, computer-generated patient avatar and the force and path measurement data. This methodology has the advantage of being able to interchange virtual patient avatars rapidly, but has

the disadvantage of requiring the user to shift focus from the physical patient (mannequin) and task at hand to the adjacent monitor.

The work by Bichlmeier, Wimme, Heining, and Navab [66] provided visual feedback to the learner to improve depth perception for AR overlays using HMDs. The work merged AR overlays with simulation mannequins, cadavers, and humans (referred to hereafter as "subjects") to show the subject's underlying anatomy to a user wearing a HMD. Both the subject and the HMD were tracked using an optical tracking system in order to update the AR views shown by the HMD. This approach was favorable because the AR visuals would be registered to the subject regardless of whether the subject or HMD moved. Furthermore, the researchers engineered cues for improving depth perception including modifications to transparency and opacity of the AR visuals, shadows from external tools onto the AR anatomy visuals, and shadows from AR anatomy visuals onto internal AR visuals of the tools inside the subject. HMDs were advantageous because the user could keep focus on the subject rather than on auxiliary monitors. Yet, using AR overlays with HMDs has some disadvantages:

1) the AR visuals are limited to users wearing HMDs,
2) HMDs constrain the user's field of view to only that area shown through the HMD, and
3) a complex, high-accuracy optical tracking system is necessary for tracking subject, HMD-user, and medical tools in order to merge the AR visuals into the view that the user is provided through the HMD.

The work by Kondo, Shiwaku, and Kijima [67, 68] provided AR visuals of anatomy to a user through projection onto a physical object (a human torso phantom). Through the technique that the authors call "Free Form Projection Display" (FFPD), the user's view point and the torso

phantom are tracked with respect to each other. The FFPD then maps the anatomy onto the torso phantom and projects the anatomy onto the torso in physical space. Thus, the user does not need to wear a HMD. As the user moves the pose of the torso, or conversely his or her pose with respect to the torso, the FFPD updates the AR visuals to display the proper view of anatomy to the user. An advantage to FFPD is that the user does not have to wear a HMD. While the projected images are visible to anyone nearby, the projected view is only corrected for the pose-tracked user. Similar to methods utilizing HMDs, FFPD also requires high-accuracy tracking systems to track the pose of the user and the object that is projected on.

Building upon different aspects from the prior art, BodyExplorer combines virtual, computer-generated images of human anatomy projected with a standard digital light processing (DLP) projector onto the surface of a life-size, 3D plastic mannequin. The system is interactive in real time by allowing the user to modify the projected anatomy images using a custom mouse-pen pointed at the mannequin's surface (**Figure 20**). The user can also perform common medical procedures such as administering medications (**Figure 21**) or inserting a breathing tube (also known as endotracheal intubation) into the mannequin (**Figure 22**), and the user can see the responses of the procedure in real time displayed on the mannequin or auxiliary screens.

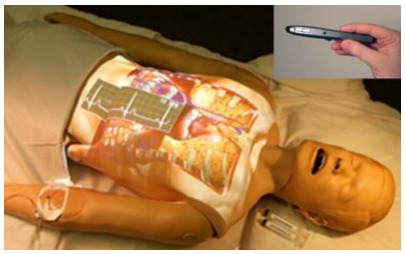


Figure 20. BodyExplorer System (© 2013, Joseph T. Samosky, PhD)

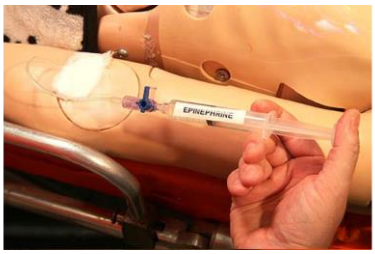


Figure 21. Automated Drug Recognition (© 2012, Joseph T. Samosky, PhD, found in [69])

Figure 22. Endotracheal Intubation (© 2011, Joseph T. Samosky, PhD, found in [70])

Over the course of BodyExplorer's development, individual AR prototypes have been engineered for the training of medical procedures including endotracheal intubation [70], medication administration [69], anatomy visualizations [71], and application of cricoid pressure during rapid sequence induction. BodyExplorer, like its predecessor AR technologies described above, contains elements for both measuring user interactions with the system and displaying information to the user based upon those measured interactions. Despite having common design elements, each of these systems were independently developed from the ground up. A common framework for AR developments within SBME could expedite innovations, removing the need to "reinvent the wheel" to research AR methodologies for other domains or procedures in healthcare education.

The work presented in this dissertation describes development of a modular framework to support continued innovations within SBME through the example of BodyExplorer. Building upon prior research for measuring and displaying user actions and feedback, the work presented in this dissertation also extends the measurements and displays needed to provide automated instruction using the modular framework.

2.5.2 Motivation for Specific Learning Module Development

This section defines the background and clinical significance related to two learning modules within BodyExplorer—one for applying cricoid pressure and the other for administering intravenous medications. Learning modules were designed for these tasks as examples showing how BodyExplorer can be used to develop training scenarios that address significant clinical needs. This section presents the background and clinical significance for these tasks.

2.5.2.1 Applying Cricoid Pressure

In the operating room, a rapid sequence induction of anesthesia (RSI) is performed when a patient is at risk for gastric aspiration. Gastric aspiration is feared when stomach contents travel up the esophagus from the stomach and down the trachea into the lungs after the patient has been rendered unconscious during induction of anesthesia. Patients at risk for gastric aspiration include trauma victims, the obese, pregnant women, patients with ascites, patients with gastric reflux, patients post gastric bypass, and patients who have eaten solid food within 6 hours of anesthesia. Variables include having solid food in the stomach, incompetent upper GI tract sphincters, pressure on the abdomen (e.g., obesity or pregnancy), and any factor which obtunds airway reflexes (such as general anesthesia). [72, 73]

A technique performed during RSI that is not performed during standard induction of anesthesia is the application of cricoid pressure, also known as Sellick's maneuver (after the originator Dr. Sellick) [74]. Application of cricoid pressure is performed in order to prevent gastric aspiration content from reaching the oropharynx and thus having access to the trachea. The procedure involves applying downward force to the cricoid cartilage (the only circumferential cartilage of the airway) which presses it against cervical vertebrae five [74]. The esophagus is thus compressed between the two structures which prevents cephalad movement of gastric content. When applying cricoid pressure, the amount of downward force needed to close the esophagus is debated, but the amount presented in the literature is 30-40 N [72, 74-76].

The benefits of applying cricoid pressure have been debated in the literature [73, 77]. Some debate that applying cricoid pressure shifts the esophagus off midline, and thus, does not fully occlude the esophagus between the cricoid cartilage and cervical vertebrae [78]. Others authors suggest that applying cricoid pressure makes endotracheal intubation more difficult [79]. The

mixed opinion on the benefit of applying cricoid pressure may also derive from untrained clinicians improperly applying cricoid pressure [75, 77]. Training methods for applying cricoid pressure using a plugged syringe [80, 81], a weight scale [82], an instrumented saline bag setup [83], and a PTT [84] have yielded positive results in support of more effective application of cricoid pressure. Despite the debate, applying cricoid pressure remains a standard procedure when performing RSI.

As part of the development for BodyExplorer, we created a quantitative method for training the technique of applying cricoid pressure. Unlike other training methods for applying cricoid pressure, our method is integrated within a FBM, allowing users to practice applying cricoid pressure in an integrated fashion with a team of clinicians. The user can practice applying cricoid pressure while another clinician intubates the mannequin, all while under the added time constraint of the mannequin being without oxygen. Furthermore, our method promotes self-learning through automated instruction.

2.5.2.2 Administering Intravenous Medications

Administration of medication happens every day in hospitals and other patient care settings, yet errors in administration are common. In a comprehensive analysis on medication errors performed by the Institute of Medicine, the frequency of error rates during medication administration in hospitals was reported to occur in 2.4–11.1 per 100 medication administration opportunities [85]. In outpatient nursing facilities, similar rates are reported at 6–20 per 100 opportunities [85]. Focusing specifically on intravenous medication errors, two studies in Europe found higher error rates of 34 per 100 opportunities [86], and 49 per 100 opportunities [87]. In a recent study on intravenous medication errors observed in the operating room in a large, tertiary care academic medical center, medication errors and adverse drug events were reported as 5 in 100 for

comparison with the other rates [88]. The types of errors observed by Nanji et al in order of highest to lowest frequency were:

1. labeling error,
2. wrong dose,
3. omitted medication or failure to act or administer a medication when needed,
4. documentation error,
5. monitoring error,
6. wrong medication,
7. and inadvertent bolus [88].

As the IOM study emphasizes, preventing medication errors is a complex problem to remedy. While automatic labeling and bar code systems for medications have potential to reduce error rates [89-92], they have not been demonstrated to be completely effective. In a study by Nanji et al, errors persisted in labeling, documentation, dosing, and wrong medications despite the presence of bar code systems [88]. Focusing on improved training for safe medication administration, a brief simulation intervention has been demonstrated to be effective at reducing the rate of observed medication errors by an order of magnitude (from 30% to 3%) [93]. Continuing to improve upon methods for safe medication administration through innovations at the point-of-care and through effective training methods could lead towards resolving the problem of preventable medication errors.

In order to build upon these findings and address reducing preventable medication errors through improved training methods, our research group developed a novel method for recognizing the administration of simulated intravenous medications within BodyExplorer [69]. Unlike current technologies utilized for detecting simulated intravenous medications in the training environment,

our method identifies the administered simulated intravenous medication based upon a property of the administered medication itself, similar to how the human body reacts based upon properties of the actual medications that are administered. Scanning bar codes or other medication identifiers like RFID tags attached to syringes—both methods currently utilized by other training technologies—doesn't ultimately measure what gets into the (simulated) patient. Our method directly measures what is injected. Using this new method, errors such as wrong dose, wrong medication, failure to act, or inadvertent bolus can be detected, and consequences can be presented to the clinician in a clear, illustrative manner to improve learning and training. We hypothesize that once the system is utilized within a training curriculum, it will lead to the translational effects of reducing preventable medication errors in the clinic—a hypothesis requiring further empirical testing outside the scope of this dissertation.

3.0 INTEGRATED SYSTEM DEVELOPMENT

This chapter presents the initial development of the prototypes that preceded BodyExplorer. It also describes the reengineering of the prototypes that was required to integrate them into a single architecture. The first two sections describe the involvement of end-users to define the overall themes and design goals for BodyExplorer development. In the next section, development of the Learning Modules is presented. The initial prototypes are summarized, design specifications for system integration are presented, design specifications for two specific Learning Modules are defined, and the modifications to the initial prototypes to fit within the design specifications are described. The final section of the chapter presents the engineering verification of the Learning Module Sensor subsystems.

3.1 USER-CENTERED DESIGN: END-USER INTERVIEWS

User-Centered Design (UCD) is a design strategy largely attributable to Don Norman from his work in the late 1980s [94, 95]. UCD has often been described in different forms by different people, but the overarching concept is consistent with Norman's original sentiment: products should be designed to meet the needs of their end-users. In order to accomplish this non-trivial task, end-users should participate in the design process.

Development of BodyExplorer utilized UCD principles by incorporating the end-users—students, instructors, clinicians, and other staff related to healthcare simulation—throughout the design process, during prototype development, integration, and validation of design goals.

Involving end-users will be reiterated throughout this dissertation when defining design goals and throughout the system validation experiments.

In order to inform development of the design themes and goals for BodyExplorer, end-users were interviewed (N=37) regarding their past and present experience within healthcare education and their experience with simulation. Those interviewed included students, professors, simulation specialists, didactic instructors, clinical instructors, program directors, fellows, department chairs, and school deans. Throughout these interviews, end-users shared some of their frustrations related to using current commercial simulators.

Common themes observed during the end-user interviews aligned with findings from Jansen et al [38]. End-users stated that simulation requires many resources. Simulation requires many instructors to deploy—someone to direct the scenario, someone rating student performance, someone planning debriefing—in addition to a simulation specialist, or technician, to run the simulation technology. Some noted that it is challenging for instructors to learn how to teach using simulation, "It takes too much time for instructors to learn how to incorporate simulation into their classrooms or learn how to use or program the simulator." Possible solutions to address these challenges would be for simulation technology to operate automatically, be pre-programmed with curriculum and evaluation metrics, and be easier to operate. Some commercial, computer-controlled FBMs do incorporate some of these features, but the implementation is not well-received by the end-users. The automated programs "do not behave exactly as [they] would like" and "programming the simulator takes too much time."

In addition to human resources and educating instructors, end-users stated that simulation requires large financial investment for space and technology. One simulation director interviewed at a community college stated that it took years to acquire the two rooms that they now use for

simulation, and that the program's yearly operating budget for new supplies and equipment is roughly $50,000. Another program director said that they required $200,000 in order to setup the physical space for the simulation lab and purchase one computer-controlled FBM. Often, due to the limited budgets, programs are not able to purchase many of the disposables for use on the FBM. Regardless of the size of program or school that the end-users were from, there was agreement that large grants are necessary in order for simulation programs to exist at institutions. Departmental budgets alone are not enough to begin simulation programs, and sometimes not enough to sustain current simulation efforts.

3.2 GENERAL THEMES FOR DEVELOPMENT

From the end-user interviews and correlation with findings from Jansen et al, primary overarching objectives for BodyExplorer were identified. First, BodyExplorer should reduce the amount of resources required to perform simulation. Reduced resources should include

1. less personnel to run the simulation,
2. less time required to learn how to use the simulator, and
3. less overall costs relating to time, money, and infrastructure.

Closely related to reducing the necessary resources, BodyExplorer should be easy and intuitive to use. Its use should not require specialized training or an advanced degree. Additionally, BodyExplorer should support continued innovations in simulation.

With the primary overarching objectives in mind, the following design goals for BodyExplorer were generated:

1. BodyExplorer should be able to operate on its own—not requiring someone specially trained, such as a simulation technician, to run the simulator.
2. BodyExplorer should support self-learning through automated instruction without requiring a physical instructor to always be present—not replacing the instructor, but acting as an "instructor multiplier."
3. BodyExplorer's automated instruction should provide quantitative, metric-driven feedback to the user based upon observed actions by the user—similar to how an instructor will provide corrective guidance and feedback to learners.
4. BodyExplorer should be easy to use—usable by all types of end-users.
5. BodyExplorer should support modular upgrades—supporting continued improvements and additions.

Design and assessment of each component of BodyExplorer defined in this dissertation were developed with these design goals in mind, and they will be acknowledged throughout.

3.3 DEVELOPMENT OF LEARNING MODULES

3.3.1 Initial Prototypes

Prior to development of BodyExplorer, a suite of prototypes for individual applications was developed by different student research design teams in the Simulation and Medical Technology Research and Development Center since 2008. Each prototype derived from an unmet educational need that was presented to the research team by clinical educators. BodyExplorer development has leveraged previous work from these prototypes into a larger, integrated system.

The first prototype was the Smart Syringe [96]. The system was designed to address medication administration safety and was utilized alongside the simulation curriculum in a research study showing the impact of simulation in reducing medication administration errors [93]. The system measured the rate of injection using a flow meter (Liquid Flow Sensor, Model 101, McMillan Company, Georgetown, TX, USA) when a user injected fluid from a syringe into IV tubing. The system provided the user with feedback on the injection rate in real-time using both an on-screen graph of flow and a colored light near the injection site (**Figure 23**). The color displayed near the injection site was related to the rate of injection—red was too fast, blue was too slow, and green was the correct rate.

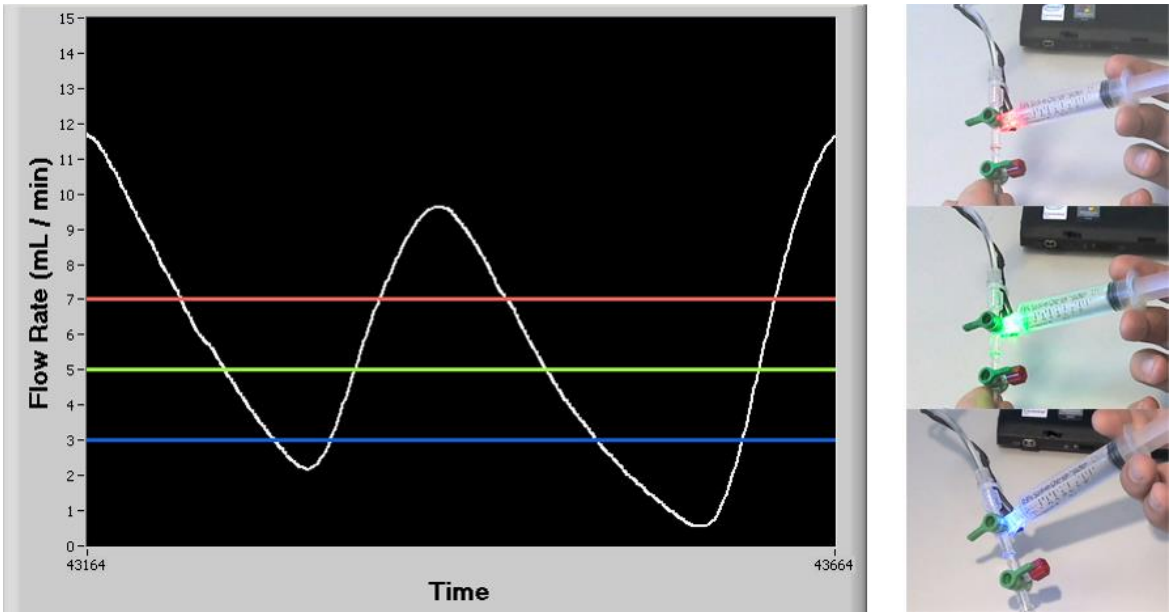


Figure 23. Smart Syringe on-screen graph of flow rate and on-syringe color feedback
(© 2009, Joseph T. Samosky, PhD)

The next prototype developed was the Interactive Projective Overlay system, with initial contributions from Baillargeon, Brown, Chaya, Enders, and Samosky [70]. The Interactive Projective Overlay system was engineered to address a need described by University of Pittsburgh

School of Medicine Anesthesiologist, Dr. William McIvor, to show students what happens with the intubation tube inside the body during intubation. The system utilized a linear array of Hall-effect sensors to measure the location and insertion depth of a modified endotracheal tube during practice laryngoscopy on a simulation mannequin. A small neodymium disk magnet was mounted in the tip of the endotracheal tube in order to measure the tube's location and depth with respect to the Hall-effect sensors mounted alongside the mannequin's trachea and esophagus. Feedback was provided to the user through AR by projecting a "see through" view onto the mannequin's chest displaying a representation of the mannequin's airway with real-time update of the endotracheal tube's position with respect to the airway as measured by the Hall-effect sensors. With the projection onto the mannequin's neck and chest, the user was provided the sense of x-ray vision of the endotracheal tube as they inserted the tube into the mannequin (**Figure 24**). Feedback was also provided through an audible alarm sounding if the user performed an incorrect action and mistakenly placed the endotracheal tube into the mannequin's esophagus.

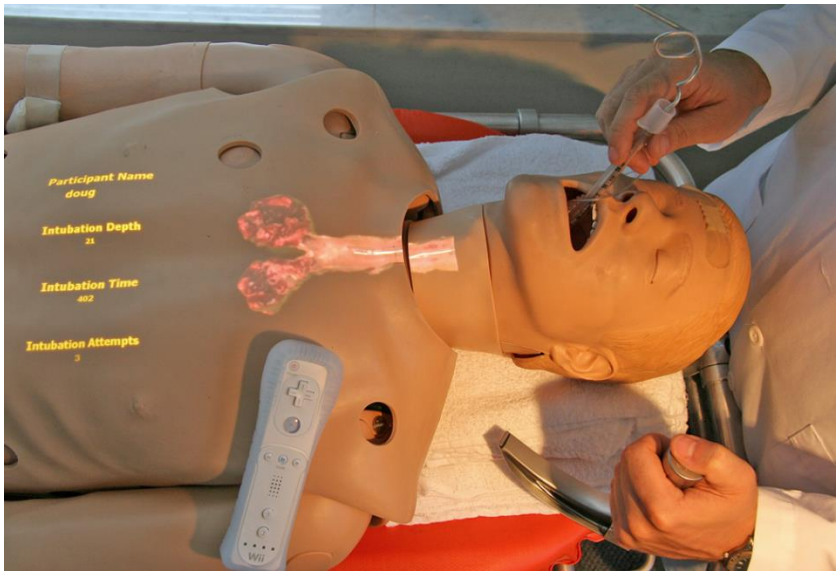


Figure 24. Interactive Projective Overlay system
(© 2011, Joseph T. Samosky, PhD, found in [70])

Building upon the projected anatomy during intubation, and to address challenges with using cadavers in anatomy and physiology labs, a prototype of BodyExplorer's anatomy windowing was developed, called BodyWindows, with initial contributions from Wang, Nelson, Bregman, Hosmer, Weaver, and Samosky [97]. Using a modified wireless mouse, the user could open, move, and resize viewport windows on the surface of the mannequin to reveal underlying anatomy. The visualization system was integrated with a system for measuring the injection of simulated medications into the mannequin [69] that leveraged previous developments on the Smart Syringe. Upon injection, the simulated medications are recognized and animations of physiology are updated accordingly.

Another prototype was developed for learning how to apply cricoid pressure based upon a need identified by Dr. David Metro at the University of Pittsburgh School of Medicine and Department of Anesthesiology [98]. Cricoid pressure was previously defined in **Section 2.5.2.1**, and to reiterate, it is applied to seal the esophagus from any gastric contents traveling cephalad, up the esophagus, and into the airway during a rapid sequence induction of anesthesia. The system, shown in **Figure 25**, measured the force applied to the cricoid of an airway demonstration model (Airway Demonstration Model, 252500, Laerdal Medical, Stavanger, Norway) using three load cells mounted between the model and the table below (25lbs Load Cell, Model FC2231-0000-0025-L, TE Connectivity Measurement Specialties, Hampton, VA, USA). While this initial prototype did not allow for differentiation on where the user was touching, it did provide the user with feedback on how hard they were pressing using on-screen visuals and visuals projected onto a mannequin's neck (**Figure 26**). Similar to the visualization methods used in Smart Syringe, graphs of real-time applied pressure and a continuous color mapping related to applied pressure (**Figure 27**) were utilized to guide the user on their performance. Additionally, an on-screen

animation showed a representation of the internal anatomy displacement caused by the application of cricoid pressure (**Figure 28**). However, the modified Airway Demonstration Model was still separate from the mannequin's neck.

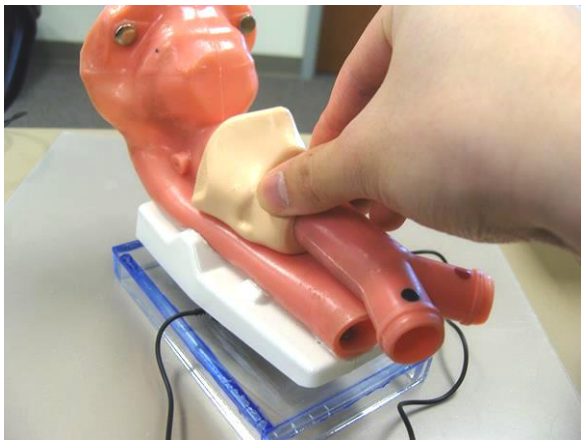


Figure 25. Prototype for measuring applied force to an airway demonstration model
(© 2014, Douglas A. Nelson Jr.)

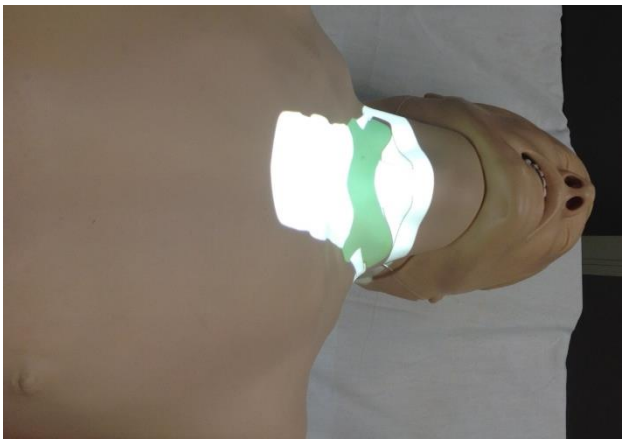


Figure 26. Cricoid pressure prototype projecting onto the mannequin's neck
(© 2014, Douglas A. Nelson Jr.)

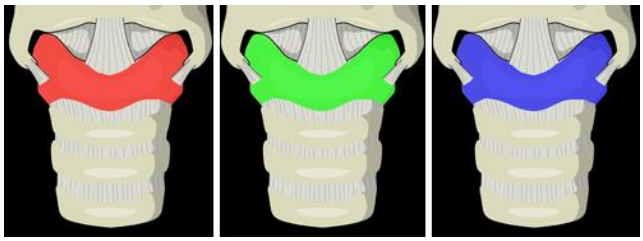


Figure 27. Continuous color mapping corresponding to the amount of force applied to the cricoid cartilage
(© 2014, Douglas A. Nelson Jr.)

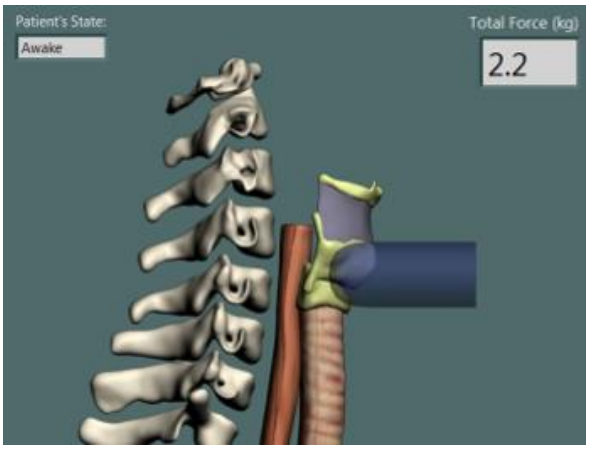


Figure 28. On-screen display showing image sequence matched to applied cricoid pressure
(© 2014, Douglas A. Nelson Jr.)

3.3.2 System Integration Design Specifications

Integration of the prototypes into BodyExplorer required specific design specifications to be established that also met the overarching themes and design goals that were defined in **Section 3.2**. The following section describes the design specifications for integration of the prototypes into BodyExplorer

3.3.2.1 Modular Design

The generalized function of each of the prototypes can be described by three subtasks:

1. measure one (or many) metric(s) related to the action of a user,
2. provide the user with an audible or visual representation related to the measured metric(s), and
3. use sounds or visuals to instruct the user on whether her/his actions are within the limits of correct or incorrect actions.

For example, in the BodyWindows prototype [97], the identity, rate, and volume of a simulated medication was measured. Anatomy of a beating heart was displayed to the user. When the user acted by injecting a simulated medication, the rate of the beating heart changed accordingly based upon what was injected, and the system alarmed if the injection was too fast.

Using the above generalization, each BodyExplorer Learning Module is comprised of the following 3 subsystems:

1. Sensors,
2. Displays of Anatomy and/or Physiology, and
3. Automated Instruction and/or Feedback.

Figure 29 shows a block diagram of the subsystems and interactions between subsystems for a BodyExplorer Learning Module. Each Sensors subsystem includes hardware and software components. The hardware includes sensors for measuring user actions and data acquisition hardware to relay data from the sensors back to the software for analysis. The Sensor subsystem includes specific software for analyzing the sensor data, making inferences on user action based upon the sensor data, and providing updates on those inferences of user action to the Automated Instruction subsystem. Each Display subsystem includes components for aural and visual displays of anatomy and physiology including computer generated images (CGI) and relevant body sounds, such as heartbeat or breathing. The Display subsystem is linked with software to animate the aural and visual displays based upon updates from the Automated Instruction subsystem. Each Automated Instruction subsystem includes software logic, and instructions and feedback components. The software logic analyzes input from the Sensor subsystem. The set of defined instructions and feedback can be triggered based upon the analysis of the software logic on the measured user actions. The instructions and feedback provide animation update instructions to the Display subsystem based upon the instructions and feedback associated with user actions.

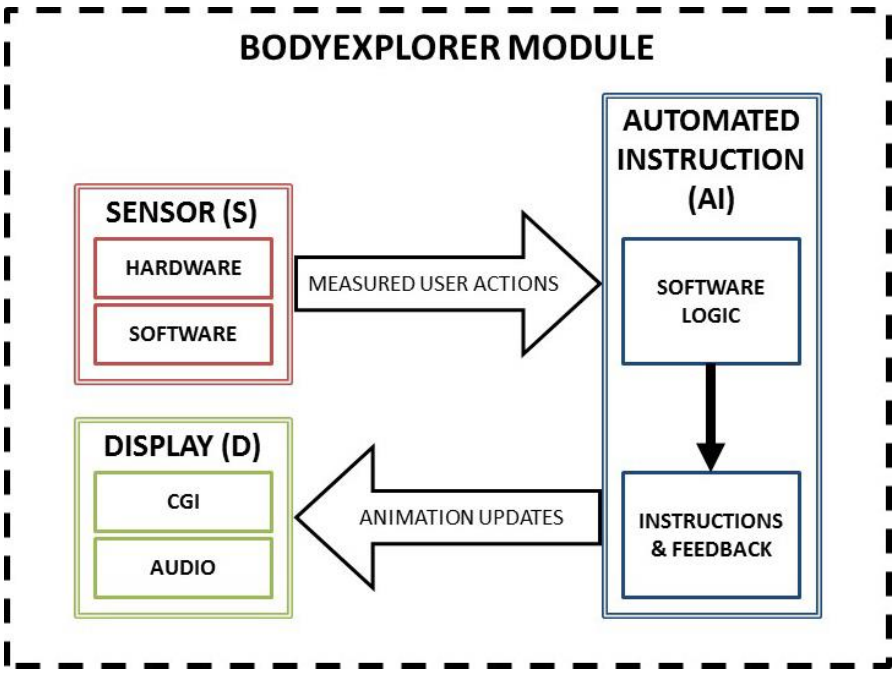


Figure 29. BodyExplorer Learning Module system diagram
(© 2016, Douglas A. Nelson Jr.)

Relating back to the themes and design goals outlined in **Section 3.2**, BodyExplorer should support modular upgrades. As such, the BodyExplorer Learning Modules support a common interface where Sensor, Display, and Automated Instruction subsystems can be interchanged for new, updated versions without breaking functionality of the Learning Module and overall system. Referencing **Figure 29**, any one of the boxes inside the dashed border should be able to be replaced with a new version of that subsystem or individual component without breaking functionality of the Learning Module—an Automated Instruction subsystem can be replaced with a whole new subsystem, or just the instructions and feedback component can be updated.

Furthermore, this modularity is supported across the entire BodyExplorer system architecture. The modular architecture is illustrated in **Figure 30**. Individual Learning Modules can be combined under the BodyExplorer system architecture. The architecture supports the addition or subtraction of Learning Modules, while still maintaining overall system functionality.

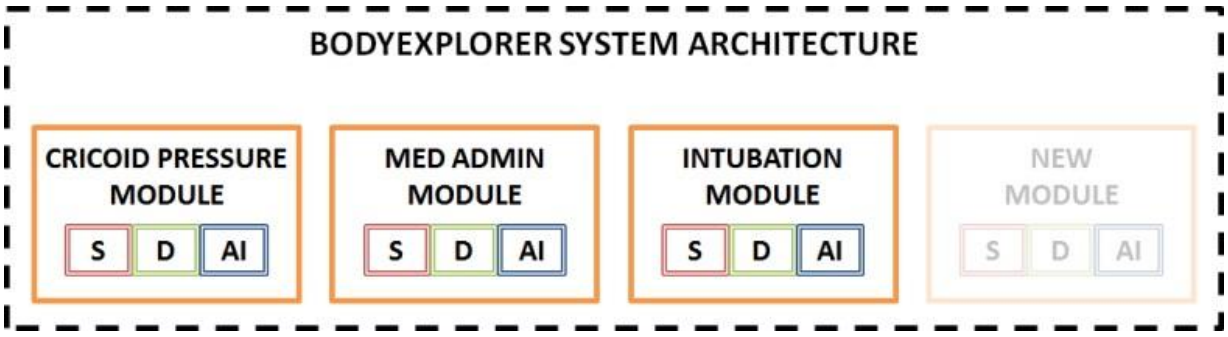


Figure 30. BodyExplorer system architecture diagram
(© 2016, Douglas A. Nelson Jr.)

Modifications of the initial prototypes to fit the BodyExplorer system architecture will be described in detail in **Section 3.3.4**.

3.3.2.2 System Components

Utilizing the system architecture described in the previous section, BodyExplorer builds upon the physical hardware demonstrated in the prototype defined in a conference paper, BodyExplorerAR [71]. **Figure 31** shows the components of BodyExplorer setup in a simulated clinical environment. A standard projector is mounted above a mannequin torso with head and neck. Images of anatomy and physiology are projected onto the surface of the mannequin. A user can interact with the projected images by using a modified wireless mouse pen (hereafter, the "input device") that is tracked by a camera mounted on the projector (Wii Remote with built-in IR camera). By incorporating BodyExplorer Learning Modules for Cricoid Pressure, Medication Administration, and Endotracheal Intubation, the user can also interact with the anatomy and physiology by performing clinical tasks on the mannequin just as they would on a human patient during the application of cricoid pressure, administration of medications, and endotracheal intubation, respectively. In addition to the projected images, a wireless auxiliary display is placed adjacent to

the mannequin, and can be moved around freely, to provide additional views of anatomy, physiology, or instruction to the user when appropriate. The auxiliary display also has functionality to display a menu to change modes and options.

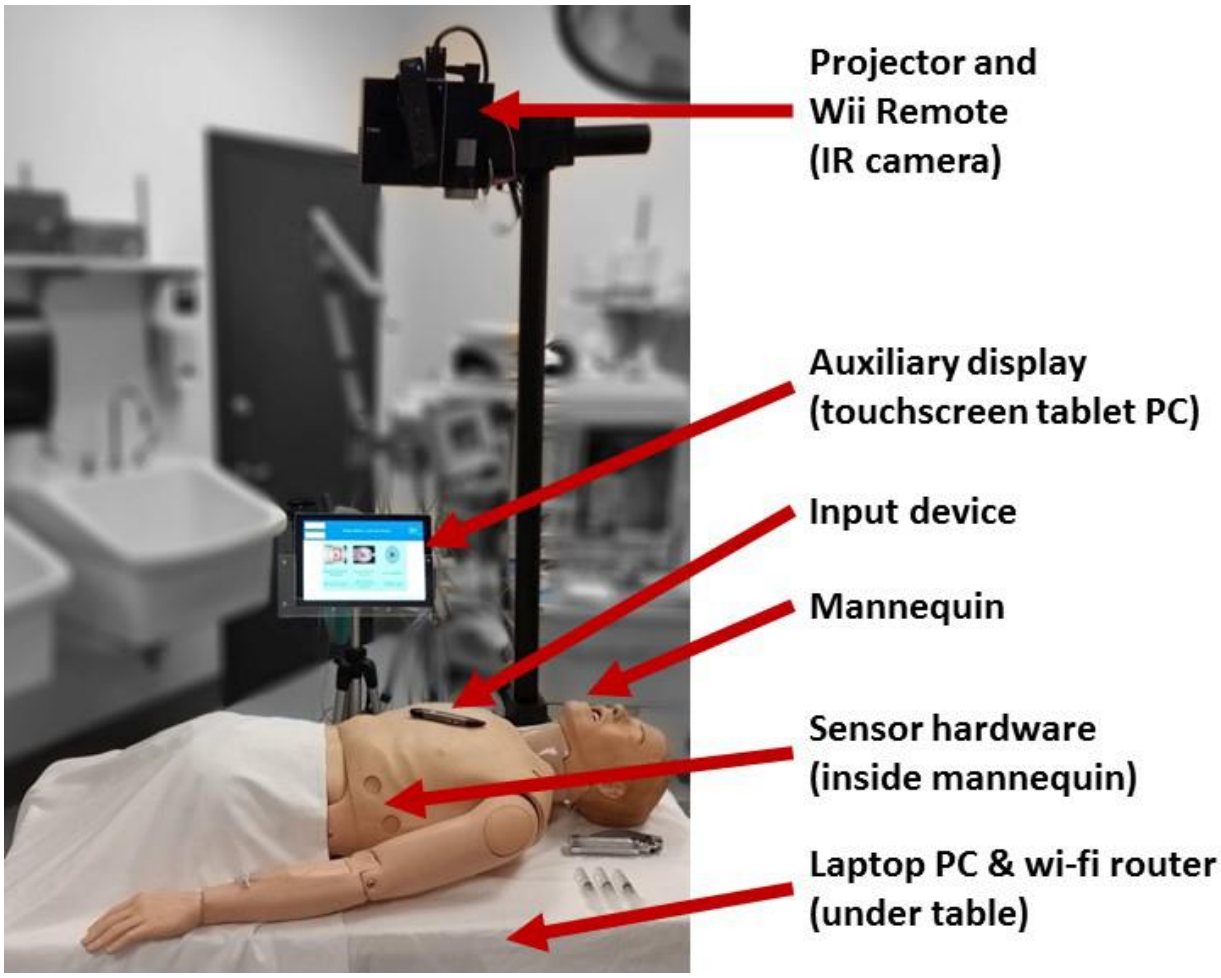


Figure 31. BodyExplorer system
(© 2016, Douglas A. Nelson Jr.)

A block diagram of the system architecture is shown in **Figure 32**. The laptop computer and auxiliary display communicate with each other over a local wireless network. Using the local wireless network, additional wireless devices can be connected to the system in the future, such as other computers or even personal mobile phones. The color scheme used in the system architecture

diagram in **Figure 29** for subsystems is preserved (red for Sensor, blue for Automated Instruction, and green for Display). The Sensor subsystem is split between the mannequin and the laptop. The mannequin contains the Sensor subsystem hardware. Supporting modularity, the hardware plugs into a USB hub inside the mannequin, and a single USB cable passes the sensor data from the mannequin to the laptop. Addition of new Sensor subsystem hardware is supported within the system architecture as shown by the faded dashed lines in the block diagram. Each of the data streams from the Sensor subsystem hardware is processed by the Sensor subsystem software before being passed to the Automated Instruction subsystem. The Automated Instruction subsystem processes the data based upon the internal logic associated with the Learning Module and drives the output to the:

1. speakers,
2. Display subsystem projecting anatomy and physiology onto the mannequin,
3. updates to the user interface, and
4. across the network to the auxiliary display.

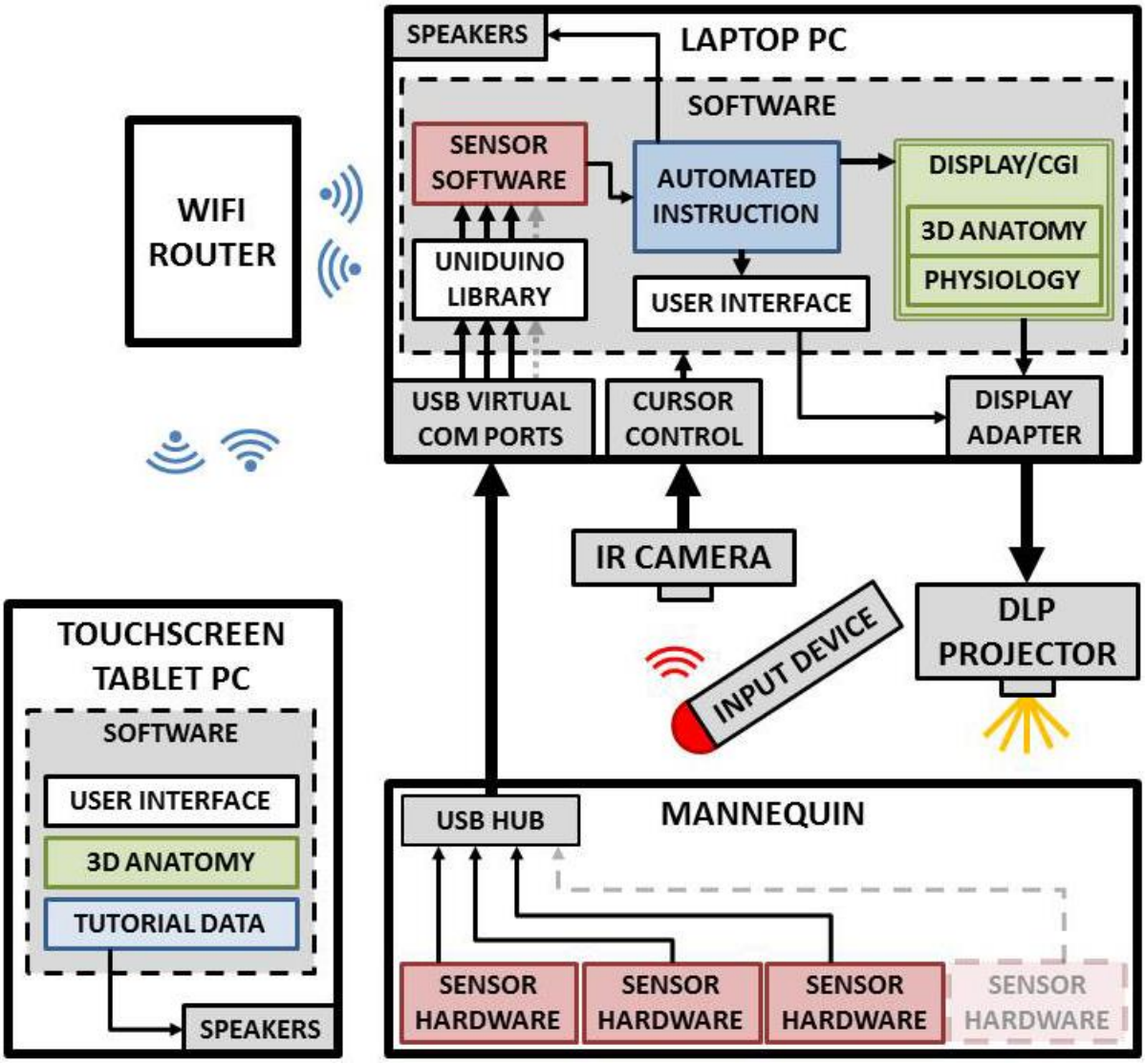


Figure 32. Block diagram of BodyExplorer system
(© 2016, Douglas A. Nelson Jr.)

3.3.3 Specific Learning Module Design Specifications

Two specific clinical tasks were selected for demonstrating automated instruction functionality within BodyExplorer. The motivation for selecting these clinical tasks was previously defined in

Section 2.5.2. The following sections describe the design specifications for creating the Cricoid Pressure Learning Module and the Medication Administration Learning Module.

3.3.3.1 Cricoid Pressure Learning Module Design Specifications

Development of the Learning Module was performed by consulting the literature on applying cricoid pressure and under clinical guidance from subject matter experts (SMEs). The SMEs consulted were Board Certified Anesthesiologists and Anesthetists at the University of Pittsburgh School of Medicine and School of Nursing. The primary SMEs were Dr. David Metro, Dr. Steven Orebaugh, and Dr. John O'Donnell. Under the guidance of the literature and SMEs, the following primary design specifications were determined:

1. The module should be able to distinguish when pressure is applied to the correct and incorrect areas of the neck.
2. Applied pressure (or force) should be held within the literature-defined pressure range.
3. Applied pressure (or force) should be held from the administration of the anesthetic medications until after successful endotracheal intubation has been verified.
4. Instruction should be provided to locate the cricoid cartilage and with respect to how much pressure (or force) to apply.

From these primary design specifications and further discussions with our clinical SMEs, the following testable engineering requirements were established:

1. The measurement system should detect and differentiate whether a user is pressing on the thyroid or cricoid cartilage when at least 10N of force is applied to each structure.
2. The measurement system should be able to measure applied force to the cricoid cartilage, continuously, over the range of 0-50N, with accuracy of ±2N.

3. Measurement of force applied to the cricoid cartilage should not drift more than ±2N when constant force is applied for 30 seconds over the range of 0-50N.

Engineering requirement (1) addresses the primary design specification (1) identified by the SMEs. The thyroid and cricoid cartilage are key cartilage elements of the upper airway and are palpable externally on the anterior neck. The entire structure is sometimes referred to as the Adam's apple in deference to a pronounced gender difference in the prominence of the structure with males more prominent than females. The v-shaped notch on the cephalad rim of the thyroid cartilage forms the thyroid notch. This thyroid notch is the first and most cephalad cartilage surface landmark identified when using a three-finger technique with the cricoid cartilage being most caudad. After locating the thyroid notch with the index finger, the body of the thyroid cartilage is palpated with the middle finger and thumb. Moving caudad, midline and lateral structures are palpated. Immediately inferior to the caudad rim of the thyroid, a depression or space is felt. This depression represents location of the membrane between the cricoid and thyroid cartilages and is described as the crico-thyroid membrane. Again moving caudad, the hard, cartilaginous structure immediately inferior to the cricothyroid membrane is the cricoid cartilage. The minimal number of cartilaginous structures to distinguish between is 2—the thyroid cartilage and the cricoid cartilage—when performing the progression to locate the cricoid cartilage.

Engineering requirements (2) and (3) address the primary design specifications (2) and (3), respectively, identified by the SMEs and in the literature. The amount of force to apply is about 10 N when the patient is awake and 30-44N for a patient under anesthesia. The range was expanded slightly in the engineering requirements to ensure measurement accuracy outside the prescribed application range. The accuracy of ±2N was chosen based upon previous variance found among users during their attempts to apply cricoid pressure [82, 83].

To meet the engineering requirements for the Cricoid Pressure Learning Module, modifications were made to the original, bench-top prototype. The primary modifications included:

1. incorporating the system for measuring force applied to the cricoid cartilage into a mannequin's neck,
2. adding functionality for differentiating touch applied to the thyroid and cricoid cartilage, and
3. adding automated instruction to teach a learner how to find the cricoid cartilage on the neck and apply the correct force to seal the esophagus to address primary design requirement (4).

Modifications (1) and (2) are described in **Section 0**, and modification (3) will be described in the chapter devoted to automated instruction development, **Chapter 6.0** . Verification of system performance against the engineering requirements will be presented in **Section 4.1**.

3.3.3.2 Medication Administration Learning Module Design Goals

Development of the Learning Module for safe medication administration resulted from discussions with stakeholders and subject matter experts (SMEs) involved in clinical education. Observations of simulation scenarios and interviews with instructors and simulation technicians were performed to understand how current medication administration training is performed, specifically focusing on how simulation technologies are being utilized to assist in the training. We observed that during instances when medications were administered to full-body mannequins during simulations, the RFID tag systems for automatic recognition of injected drug simulants was not used when it was available. Often, the method for recording what was administered was done by a hand gesture from an instructor to a simulation technician operating the simulator, or verbalization of the medication

and amount given by the user—the accurate content of the syringe was not identified and the amount administered was quantified by an instructor looking at the syringe pre- and post-injection, if at all. When presenting initial work towards development of our method for detecting administered drug simulants at the 12th Annual International Meeting on Simulation in Healthcare, discussions with other stakeholders in clinical education supported our findings.

One of the reasons identified for not using the automatic system was that it did not work reliably. The RFID system for medication identification is comprised of RFID readers located in the left antecubital region (elbow) and the mouth of the FBM. The RFID reader detects the presence of a RFID tag that can be attached with Velcro to any syringe. The size of the RFID tag aside, the larger issue is that when multiple RFID-tagged syringes are adjacent to either RFID reader, the system cannot identify what syringe is doing the injecting during administration. This issue is problematic because administration of IV medication should be followed with a saline flush to ensure that the dose of medication entered the patient and is not sitting in the IV tubing outside the patient. As such, injections would require a minimum of 2 syringes adjacent to the site of injection—a case where the RFID system would fail at identifying administration automatically.

To address the limitation of current methods for automatic detection of administered IV medications during training, and based upon discussions with SMEs, we developed the following primary design goals:

1. Administered IV medications should be identified only after they enter the mannequin.
2. Identification of administered IV medications should be accurate even when multiple syringes of medications are present near the site of injection.
3. The volume and rate of injection for administered medications should be quantified.

4. The volume and rate of injection of the administered medication should automatically update the mannequin's physiology without instructor or technician intervention.

Our previous research addressed these primary design goals [69], but did not quantify system performance. The following parameters will be evaluated to establish a baseline assessment of accuracy and to understand the limits of the Medication Administration Learning Module Sensor subsystem:

1. Accuracy of medication identification
2. Accuracy of measured volume

3.3.4 Prototype Modifications

The initial prototypes described in **Section 3.3.1** required modifications to fit the design specifications of the BodyExplorer system architecture defined in **Section 0**. As part of the integration, the combination of the prototypes required the hardware of the Sensor subsystems to fit inside the mannequin. Furthermore, a software development environment needed to be chosen to support the software of the Sensor subsystems under the BodyExplorer system architecture. This section describes the rationale and methods for the hardware modifications and selection of the software development environment.

3.3.4.1 Data Acquisition Hardware

The data acquisition hardware for each of the initial module prototypes was the USB-6009 (USB-6009, National Instruments Corporation, Austin, TX). In designing for future expansion and modularity, while also minimizing cost and size to fit inside the mannequin, a new data acquisition platform was required. The open-source Arduino platform, specifically the Arduino Nano

(Arduino Nano v3.0, Gravitech LLC, Minden, NV), was chosen to replace the USB-6009 in BodyExplorer. While the Arduino Nano has less bit resolution on its analog-to-digital converter (10-bit compared to 14-bit), lower full-scale range (0-5VDC compared to -10-10VDC), and lower sampling rate (9kS/s compared to 48kS/s), the Arduino is about an order of magnitude less expensive (retail prices of $35 compared to $300), has a smaller footprint by almost an order of magnitude (7.7 cm^2 compared to 70.0 cm^2), and a smaller overall volume (10 cm^3 including header pins compared to 160 cm^3 including terminal blocks). The Arduino platform also easily integrates with the new software development environment, defined in more detail later in **Section 0**. Furthermore, switching to the open-source Arduino platform enables future development to build upon the microcontroller architecture for more customized hardware without being tied to a specific product or manufacturer. This hardware modification affords the opportunity for future innovators to design new sensor modules that can be added to BodyExplorer and directly aligns with the overarching themes for development.

3.3.4.2 Endotracheal Intubation Learning Module Sensor

The Hall-effect and magnet measurement system in an early prototype [70] was modified to reduce cost and save space around the trachea. Additional space was required around the trachea to accommodate the addition of a cartilage model of the Cricoid Pressure Learning Module. Both reduction of cost and saving of space were accomplished by reducing the number of Hall-effect sensors from seven on the trachea to four. Reducing the number of Hall-effect sensors to four maintained linear accuracy of the algorithm for calculating depth of intubation from the response of the Hall-effect sensors during a complete insertion and removal of the tube, as shown in **Figure 33**. Ideally, the calculated depth should match the actual depth, as shown by the red line in the graph. The linear regression model (shown in black) of the raw data (shown in blue) is a close

approximation to the ideal response. The average absolute error between the calculated depth and ideal response (actual depth) is 0.42cm with standard deviation of 0.24cm and maximum absolute error of 1.02cm

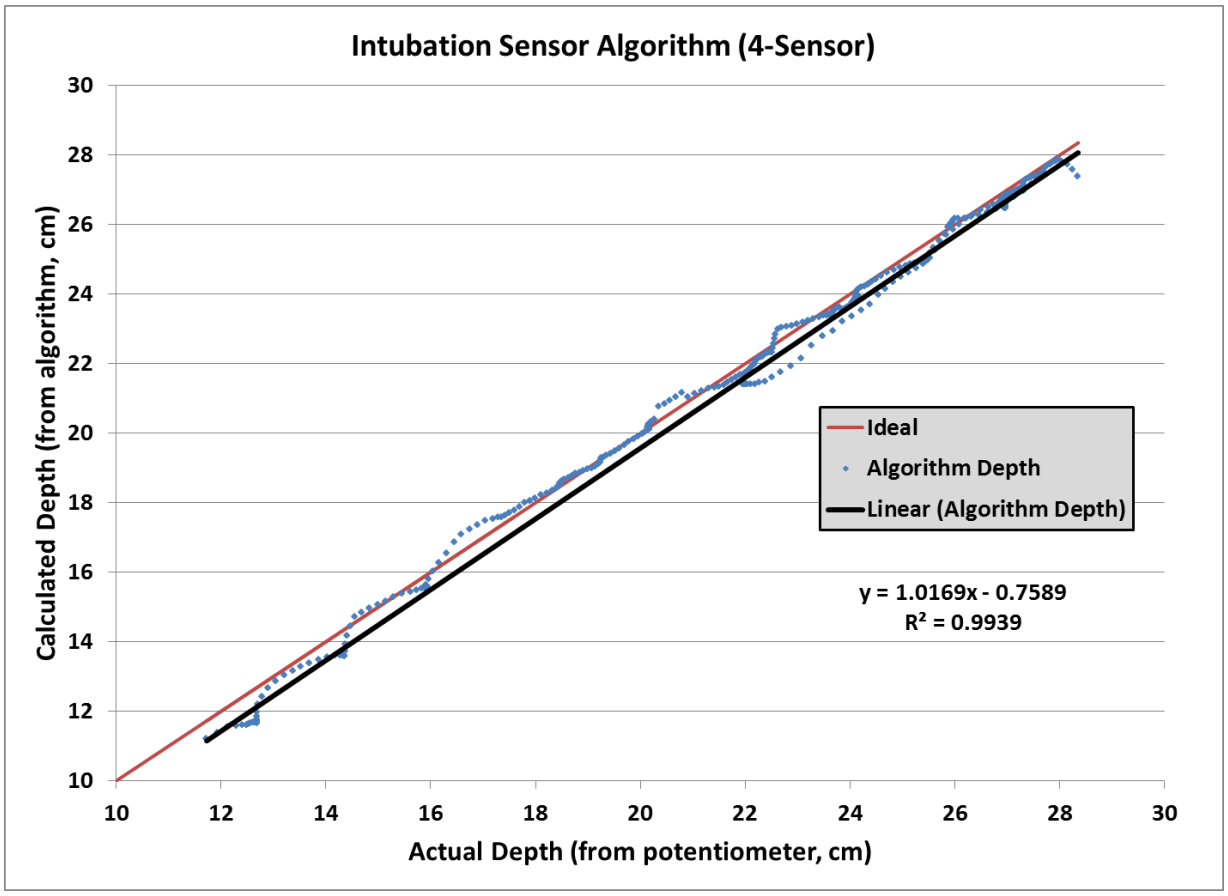


Figure 33. Results from the depth calculation algorithm using data from four sensors on the trachea.

Additionally, a mounting apparatus was designed and manufactured using laser cut polycarbonate (0.030" thick PETG, Total Plastics Inc, Pittsburgh, PA). The new mounting apparatus, shown in **Figure 34**, enabled consistent spacing between Hall-effect sensors as well as the ability to replace individual sensors when needed. The new mounting apparatus was affixed to the mannequin's trachea for measuring insertion depth of an endotracheal tube. Using a similar

method with laser cut polycarbonate, a fifth sensor was mounted to the esophagus to detect incorrect intubation into the esophagus.

Figure 34. New mounting apparatus for the Hall-effect sensors
(© 2015, Douglas A. Nelson Jr.)

Furthermore, a printed circuit board (PCB) was engineered and assembled (**Figure 35**) as a robust and scalable alternative to the initial prototype circuit on a solderless breadboard and solderable perfboard. The PCB includes headers for mounting the Arduino Nano microcontroller. The PCB also has an auxiliary power supply (+12VDC) that is scaled down by an onboard voltage regulator (LM317AHV, Fairchild Semiconductor) set to output +8VDC, and the scaled voltage provides regulated power to the Hall-effect sensors. The output from the sensors is filtered with a first-order low-pass RC filter ($R=1k\Omega$, $C=10\mu F$, $f_c \approx 16Hz$) and scaled to 0-5VDC using a voltage divider ($R_1=1k\Omega$, $R_2=1.5k\Omega$) before input into the microcontroller. The PCB also had a connector for attaching a linear potentiometer for calibrating the sensors. The circuit diagram for the endotracheal intubation electronics is shown in **Appendix A**.

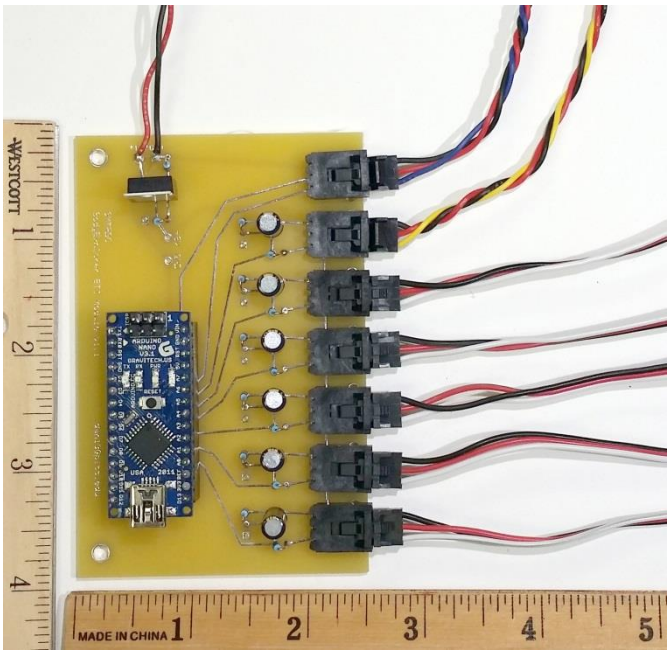


Figure 35. New PCB and wiring harness for measuring intubation depth
(© 2015, Douglas A. Nelson Jr.)

Finally, the PCB contained a connector for attaching a sixth Hall-effect sensor. This sixth Hall-effect sensor was added by Samosky as a means for inferring the amount of air that was injected into the intubation tube's cuff. To infer the amount of air, the sixth Hall-effect sensor was attached near the tip of a standard 10mL syringe. A neodymium magnet was inserted inside the syringe's plunger. The position of the plunger was then calculated based upon the magnetic field strength measured by the sensor affixed to the syringe (**Figure 36**). When the syringe was used to fill the intubation tube's cuff with air, the sensor measurement was communicated to the BodyExplorer software and the syringe's plunger position was calculated based upon calibration data. By measuring the syringe's plunger position, the system could infer how inflated/deflated the cuff was, and update the computer-generated visuals for the cuff projected on the body. This method for measuring cuff inflation was limited in that it acted as a surrogate to measuring the true volume of air entering the cuff. Additionally, the syringe was tethered to the mannequin.

Despite these limitations, the method provided a simple solution to add functionality for inferring inflation of the cuff during intubation in a way that was like how the cuff would be inflated in the clinic.

Figure 36. Modified syringe with Hall-effect sensor mounted near the tip for measuring plunger position, originally developed by Samosky
(© 2016, Douglas A. Nelson Jr.)

3.3.4.3 Medication Administration Learning Module Sensor

As defined in a previous publication [69], the algorithm for detecting drug simulant injections is based upon salinity of the solution injected being matched to a known solution calibration lookup table. After changing the data acquisition hardware to the Arduino Nano, modifications to the Medication Administration Learning Module Sensor were made to accommodate the lower bit resolution (10-bit) and dynamic range (0-5V) of the Arduino Nano.

First, design modifications were made to the conductivity sensor in order to minimize the volume of drug simulant between the two electrodes, which would theoretically minimize dilution

or mixing effects in the volume of fluid between the two electrodes. Two stainless steel electrodes (91793A057, McMaster Carr) were threaded into a tube fitting (5117K52, McMaster Carr). A twisted pair of wires was attached to the steel electrodes with wire terminals (A27224-ND, Digikey Electronics). The wires traveled to the electronic circuitry for measuring conductivity. The redesigned conductivity sensor is shown in **Figure 37**.

Figure 37. New conductivity sensor with wires connected to the stainless-steel electrodes (© 2016, Douglas A. Nelson Jr.)

Then, the relationship between salt concentration and the original circuit measurement was reviewed to guide circuit redesign (**Figure 38**). Analysis of the relationship revealed that the measured voltage at salt concentration of 40 g/L falls outside the dynamic range of the Arduino Nano. Thus, without any other modifications to the system, drug simulants mixed at 40 g/L will saturate measurements on the Arduino when injected.

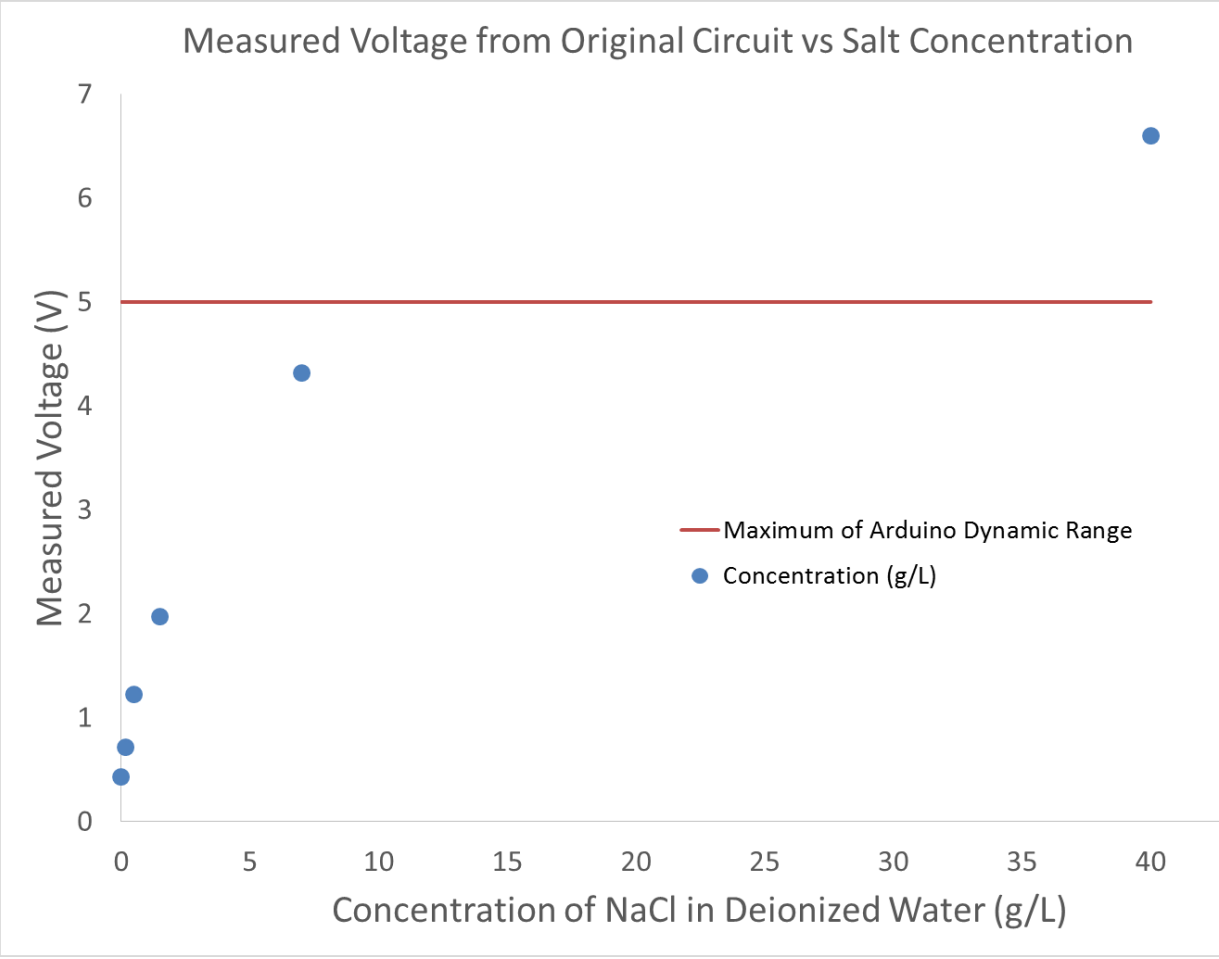


Figure 38. Original circuit measurements of injected drug simulants with concentrations of 0.0, 0.167, 0.5, 1.5, 7.0, and 40.0 g/L.

The nonlinear relationship observed between measured voltage and salt concentration over the graphed span is not surprising based upon conductivity versus concentration data reported for aqueous sodium chloride in the CRC Handbook on page 5-73 [99]. However, the relationship approaches linearity for low concentrations of salt. To utilize lower concentrations of salt as drug simulants, the measurable signal must be increased to utilize more of the Arduino's dynamic range and overcome the lower bit-resolution. These modifications are necessary to ensure adequate spacing between calibrated set-points in the lookup table for identifying injected drug simulants.

Modifications to increase the measurable signal were made from the original conductivity circuit presented by da Rocha in [100] and our research group's previous work in [69]. **Figure 39** shows the replacement of the process for half-wave rectification with gain adjustment and full-wave rectification to provide a method for tuning the output voltage and increasing measured voltage in the modified circuit. The modified circuit diagram is shown in **Appendix A**.

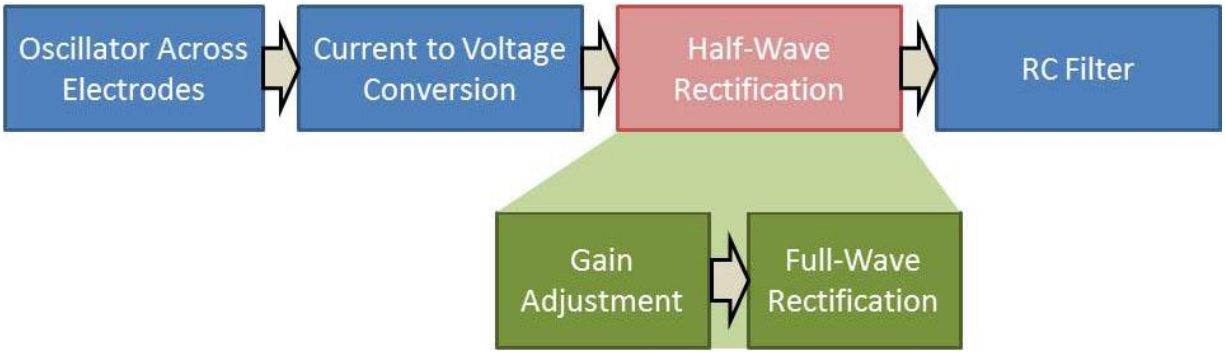


Figure 39. Block diagram of the modified electronic circuitry for the drug simulant recognition system (© 2016, Douglas A. Nelson Jr.)

The relationship between measured voltage and salt concentration using the modified circuit is shown in **Figure 40**, and the relationship approximates linearity with high correlation (R^2=0.984).

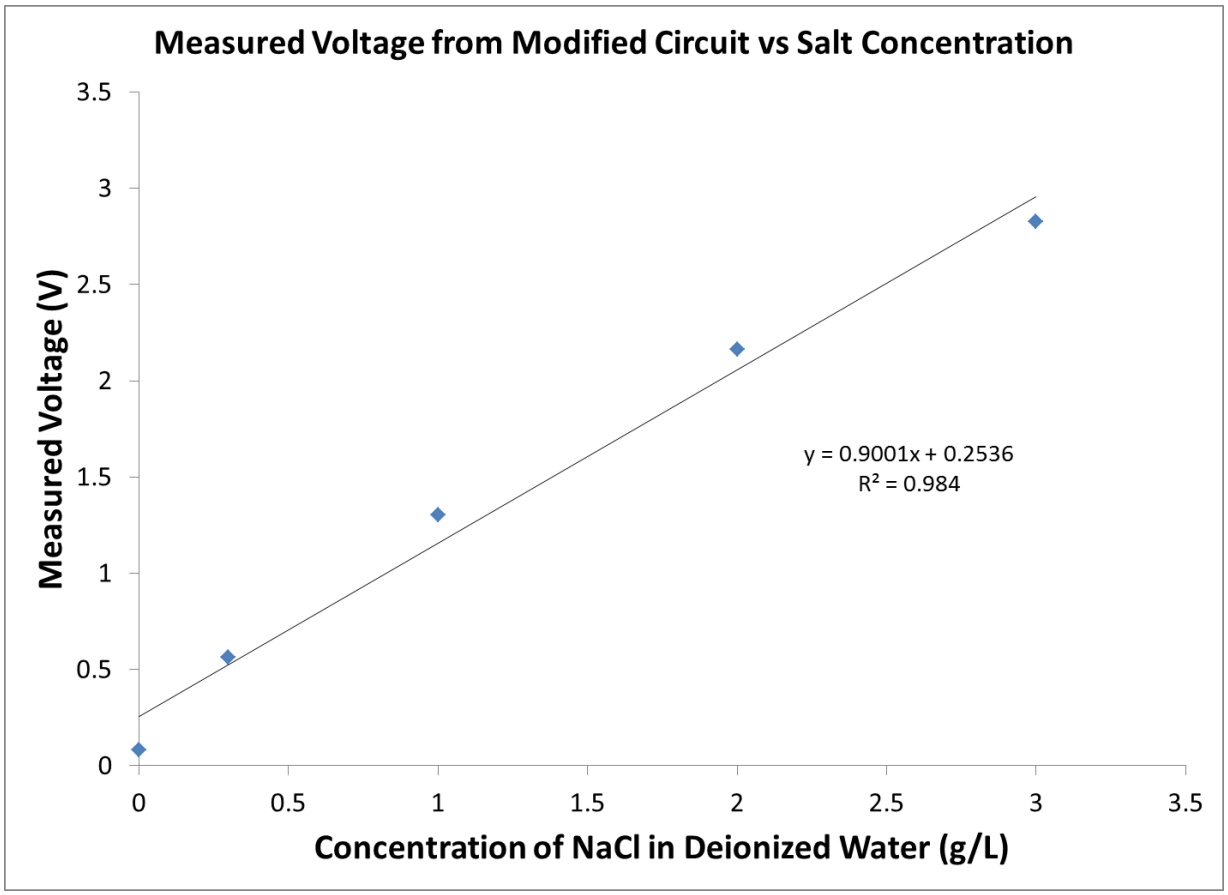


Figure 40. Modified circuit measurements of injected drug simulants with concentrations of 0.0, 0.3, 1.0, 2.0, and 3.0 g/L.

A PCB (**Figure 41**) was engineered and assembled to replace the solderless breadboard and solderable perfboard of the original system. Modifications to the electronic circuit defined in **Figure 39** were included in the PCB design, as well as headers for mounting the Arduino Nano microcontroller.

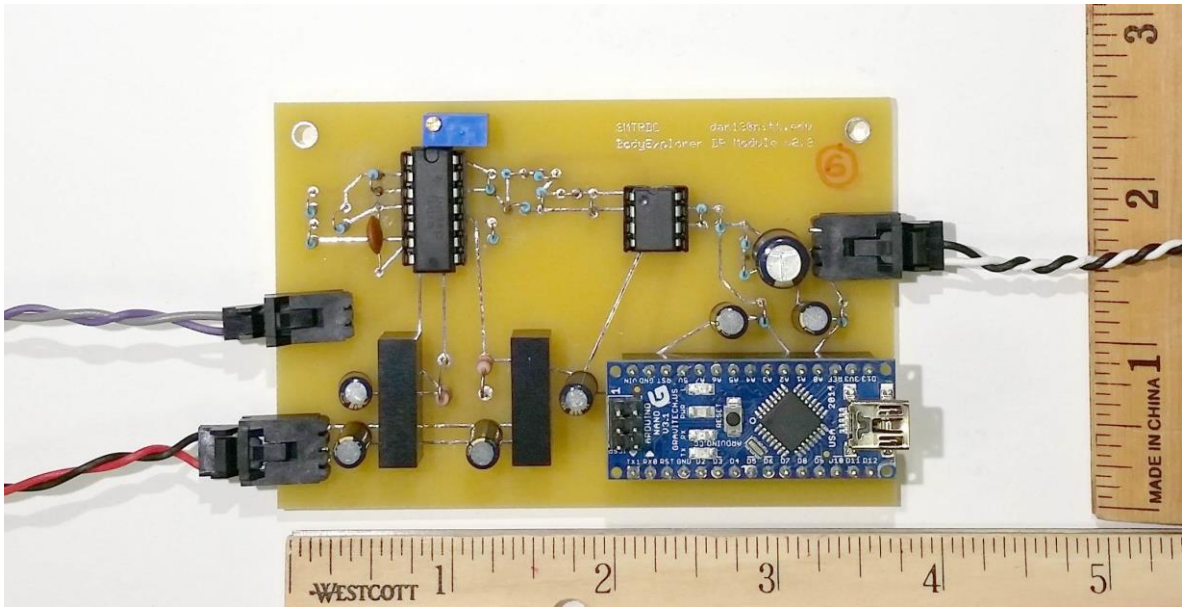


Figure 41. New PCB and wiring harness for drug recognition
(© 2016, Douglas A. Nelson Jr.)

3.3.4.4 Cricoid Pressure Learning Module Sensor

In order to meet the engineering requirements for the Cricoid Pressure Learning Module defined in **Section 3.3.3.1**, modifications were made to the original, bench-top prototype. The primary modifications included:

1) incorporating the system for measuring force applied to the cricoid cartilage into a mannequin's neck,
2) adding functionality for differentiating touch applied to the thyroid and cricoid cartilage, and
3) adding automated instruction to teach a learner how to find the cricoid cartilage on the neck and apply the correct force to seal the esophagus.

Modifications (1) and (2) will be described in this section, and modification (3) will be described in the chapter devoted to automated instruction development, **Chapter 6.0** .

Incorporating Cartilage Models into the Neck

Modifications to the neck were necessary to incorporate the system for measuring force applied to the cricoid cartilage into the mannequin's neck. First, the native cricoid and thyroid cartilage was removed. The native cartilage was not anatomically accurate and did not have the proper surface area for learning how to apply cricoid pressure. Then, after referencing cadavers, textbooks, and digital MRI datasets, we designed and 3D printed a model of the cricoid and thyroid cartilage.

There were slight alterations made during the design of the cartilage model differing from true anatomical accuracy (**Figure 42**). These modifications were made iteratively to adjust and improve the fit around the airway of the mannequin whilst maintaining the proper surface area and landmarks that the end-user would touch. First, the cricoid and thyroid cartilage were fused together along the lateral sides. The pieces of cartilage were fused along the sides in order to maintain a consistent spacing of the gap between the cricoid and thyroid cartilage—the crico-thyroid membrane area. Initial designs with the pieces separated did not maintain consistent spacing of the crico-thyroid membrane area, and the thyroid cartilage was free to slide around the mannequin's larynx, losing registration with respect to the cricoid cartilage. Additionally, the inner width of the cricoid cartilage ring was extended in the anterior/posterior direction to allow the mannequin's trachea to fit through the cricoid cartilage.

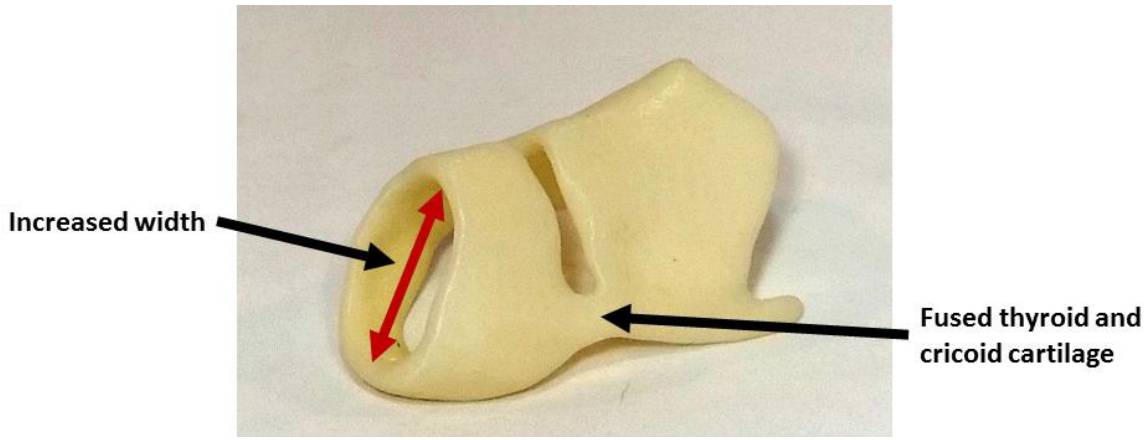


Figure 42. 3D printed cartilage model including the thyroid and cricoid cartilage
(© 2016, Douglas A. Nelson Jr.)

Adding Touch Detection and Force Measurement

After modifying the cartilage in the neck, sensors for detecting touch and applied force were integrated. Two different sets of sensors were utilized: 1) force sensing resistors (FSRs), and 2) a load cell. The FSRs (0.5" Round Force Sensing Resistor – FSR 402, Part Number 30-81794, Interlink Electronics) were installed on the anterior faces of the cricoid and thyroid cartilage, two on each piece of cartilage, with one on each side of midline. The primary function of the FSRs is to detect location of user's touch—is the user touching the cricoid or thyroid cartilage? Natively, the FSRs can only detect forces applied directly to the sensor's surface area. This was problematic because the system should detect touch to any area of the cartilage. To overcome this limitation, vacuum-formed plastic shells were manufactured in the shape of the anterior faces of the cricoid and thyroid cartilage. The completed sensor assembly on each piece of cartilage (**Figure 43**) is comprised of: 1) 3D printed cartilage base, 2) 2 FSRs adhered to the anterior face of the cartilage, one on each side of midline, 3) an adhesive "puck" covering the surface of each FSR, and 4) the vacuum-formed plastic shell, custom to that piece of cartilage.

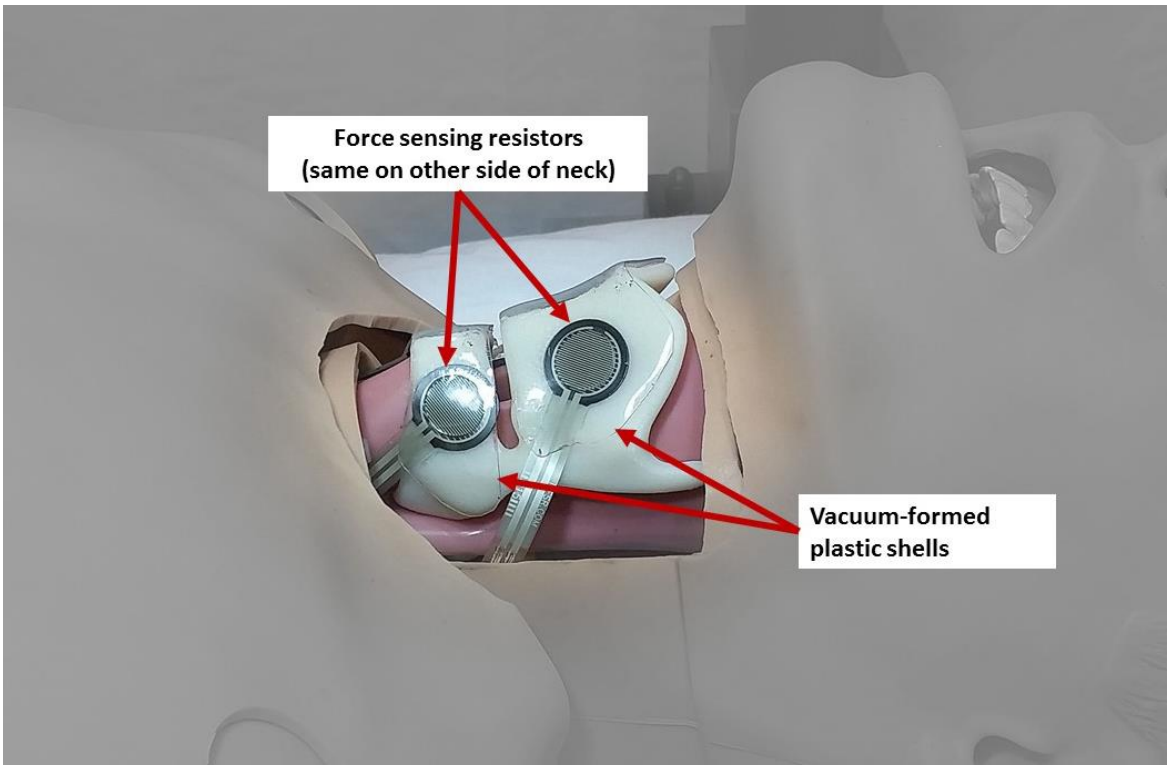


Figure 43. Sensor assembly on the anterior faces of the cartilage model
(© 2016, Douglas A. Nelson Jr.)

While the FSRs primarily measure location of touch, an amplified button load cell (25 lbf Load Cell Force Sensor, FC2231-0000-0025-L, TE Connectivity Measurement Specialists) was incorporated to provide accurate measurement of applied force during application of cricoid pressure. The load cell linearly measures applied forces in the range of 0-25 lb (0-111.1 N). A load cell housing assembly was created to position and mount the load cell in the mannequin's neck, posterior to the cricoid cartilage (**Figure 44**). When cricoid pressure is applied, force is transmitted through the cricoid cartilage, pinching the esophagus and, in this case, the load cell. The load cell quantifies the amount of force applied to the cricoid cartilage.

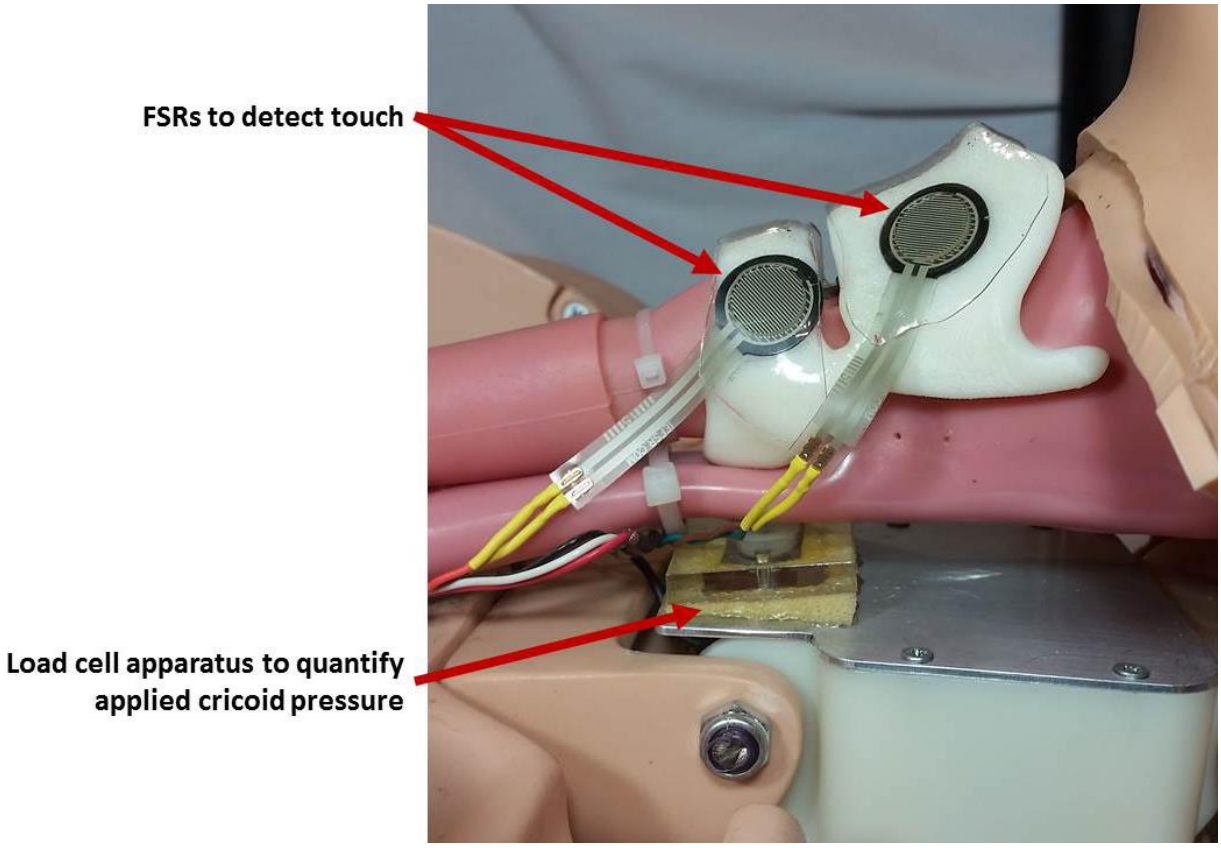


Figure 44. Load cell housing assembly mounted in the mannequin's neck
(© 2016, Douglas A. Nelson Jr.)

The load cell housing assembly (**Figure 45**) includes an aluminum backer plate for mounting the load cell to a larger aluminum mounting plate. The large aluminum mounting plate was manufactured to replace the preexisting steel plate in the mannequin's neck, and the large aluminum mounting plate allows the load cell to be mounted posterior and centered behind the cricoid cartilage in the mannequin's neck. The larger aluminum mounting plate has a hole centered above the load cell. The hole allows the button of the load cell to pass through the large aluminum mounting plate, and sit proud above the plate. A polycarbonate plate with centered "puck" was mounted onto the top of the large aluminum mounting plate, with the "puck" centered over the protruding button of the load cell. A piece of foam borders the edge of the polycarbonate plate and

is adhered between polycarbonate plate and the large aluminum mounting plate to ensure that the "puck" on the polycarbonate plate does not natively touch the load cell's button. The foam spacer ensures that the "puck" and button of the load cell only come into contact—and thus the load cell measures force—when force is applied to the top of the polycarbonate plate as opposed to the polycarbonate plate bottoming out on the large aluminum plate. The load cell housing assembly mounted in the mannequin's neck enables the load cell to measure force applied to the anterior side of the cricoid cartilage.

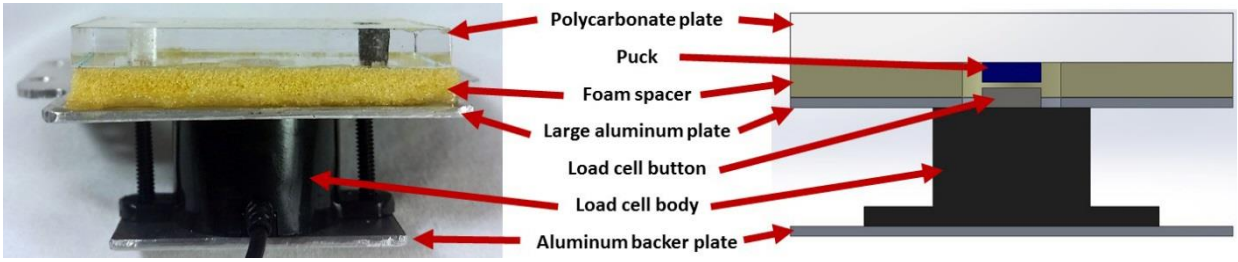


Figure 45. Load cell housing assembly in profile to show the individual components. The left image shows the actual assembly with puck and load cell button hidden. The right images shows a computer-aided drafting model of the assembly with cross-sectional cut to show the puck and load cell button.

Incorporating New Neck Skin

After instrumenting the mannequin's neck with sensors for detecting touch and measuring applied force, the exterior surface of the neck was modified to allow the cartilage structures to be palpated. The native neck skin was approximately 3/8 inch thick, single-ply vinyl and did not permit the user to palpate for the cartilage structures beneath the skin—a necessary task to identify the cricoid cartilage before applying pressure. The native neck skin was removed from the neck on the anterior side above the cricoid and thyroid cartilage, permitting new neck skin to cover the opening. A new neck skin was manufactured, with original recipe and process developed by Andy

Hosmer, using a custom 3-ply silicone configuration. **Figure 46** shows the layering for the new neck skin. Skin-tinted platinum-cure silicone (Dragon Skin 10, tinted with Silpig coloring, Smooth-On), was brushed into an acrylic mold (Layer 1). Then, a thin layer of softer platinum-cured silicone (Dragon Skin 10 with added Slacker, Smooth-On) was poured over the cured skin-tinted silicone already in the mold (Layer 2). Finally, another layer of the initial skin-tinted silicone was brushed onto the cured silicone (Layer 3). Before the final layer of silicone cured, 2 sheets of fabric (Organza Fabric, Jo-Ann Fabrics, item #4491676) were flattened and pressed into the silicone to reinforce the silicone (Layer 4).

Figure 46. Layering of silicone and fabric, forming a new neck skin that allows the cricoid and thyroid cartilage to be palpated on the mannequin's neck
(© 2015, Andrew L. Hosmer)

After final curing of the silicone, the new, 3-ply neck skin was de-molded and cut to cover the opening on the neck above the cricoid and thyroid cartilage, wrap around the neck, and securely fasten. Additional tabs were shaped out of the cephalad side of the silicone neck skin for attaching to the native neck skin through slots just below the mannequin's jaw line (**Figure 47**). These tabs ensure that the silicone neck skin does not slide around the neck.

Figure 47. New neck skin cut to shape and ready to install on the mannequin
(© 2016, Douglas A. Nelson Jr.)

3.3.4.5 Wireless Input Device

Early versions of BodyExplorer utilized a modified wireless computer mouse as an input device that allowed users to interact with projected images of anatomy on the surface of the mannequin [97]. **Figure 48** shows an example of the first input device used during the first round of end-user testing. Since the incorporation of the first wireless input device, multiple design iterations were performed on the device, and improvements have been made based upon user-feedback received during structured usability testing sessions (See **Section 5.0** for details related to the user feedback during usability testing).

Figure 48. Input device for interacting with BodyExplorer's projected images
(© 2016, Douglas A. Nelson Jr.)

3.3.5 Software Modifications

3.3.5.1 LabVIEW

The initial prototypes' software was developed using LabVIEW (National Instruments, Austin, TX). As a result, the initial integration of the prototypes' software was performed using LabVIEW. However, various limitations were realized with continued development in LabVIEW. Due to only a subset of the OpenGL library being implemented and accessible from within LabVIEW's software, inefficiencies in the display of graphical anatomy objects limited real-time system performance. Thus, complete system integration was not possible using LabVIEW and resulted in separate applications for:

1. anatomy windowing and IV medication administration, and

2. rapid sequence induction of anesthesia, including anatomy visualizations, IV medication administration, cricoid pressure application, and “x-ray vision” during intubation.

One of the main reasons separate applications were required is because the displayed anatomy and physiology animations that were derived from 2D image sequences utilized a large amount of system resources to render in the application. From a development standpoint, the animations composed of 2D image sequences were not scalable to support creation and integration of future learning modules into the system. Additionally, the combination of heart and lung animations, with separate control of heart rate and respiration rate, was not feasible using this method.

3.3.5.2 Unity

After identifying these limitations for continued software development using LabVIEW, the Unity game engine was chosen as the new software development environment. While LabVIEW is designed for rapid creation of tools for data acquisition, data visualization, and test automation, Unity is designed for the rapid creation of games with different levels, 3D graphics and animations, as well as scaffolding for user interactions in the real world to control actions in the game. Learning Modules within BodyExplorer can be developed as different levels in the Unity game. The 3D anatomy and animations can be displayed in levels, when necessary, and have associated animations that are controlled programmatically, either caused by user actions or by preprogrammed automation in the Learning Module. Different views of the anatomy can rapidly be created in Unity by simply changing the position and location of the in-scene camera or by changing visibility of different anatomy objects in the scene. Furthermore, new developments in the Unity platform support cross-platform compatibility between PC, Mac, Linux, Android, and

iOS operating systems—a feature that supports the overarching themes and design goals for modularity and reuse.

Unity uses an integrated development environment that allows creating software through both graphical representation and procedural lines of code in C#. The graphical representation allowed rapid creation of user interface elements such as menus, buttons, and 3D anatomy models. Within Unity, the graphical representations could easily be connected to custom procedural lines of code. Furthermore, "prefabs" could be created within Unity for common interface elements. For example, a prefab for the 3D anatomy with animations for respiration and heartbeat could be created and connected to the procedural code for modifying the speed of the respective animations. This prefab could be used in development of additional Learning Modules simply by dragging and dropping the prefab into the Learning Module. Any modifications to the prefab, including updates to the underlying procedural code, would be instantly propagated to all instances of the prefab across all Learning Modules.

Making the switch to Unity also required modifications to the data acquisition methods and technologies employed to read measurements from BodyExplorer Learning Module Sensor subsystems. The rationale for switching to the Arduino microcontroller platform was mentioned previously in **Section 3.3.4.1**. In order to interface between the Learning Module Sensor subsystem software and hardware, the Uniduino library was utilized (EDWON, LLC, [101]). The library helped to streamline the software and hardware transition, allowing sensor data to be passed directly into Unity from the Arduino Nano devices.

Moving to Unity allowed the 2D images of anatomy in previous prototypes to be represented in the software as 3D anatomy objects composed of polygons mapped with images to provide the visual color and texturing. **Figure 49** shows the 3D anatomy integrated into the Unity

development environment with the polygons of the lung objects highlighted in green. The polygons are mapped with the respective organ coloring and texture.

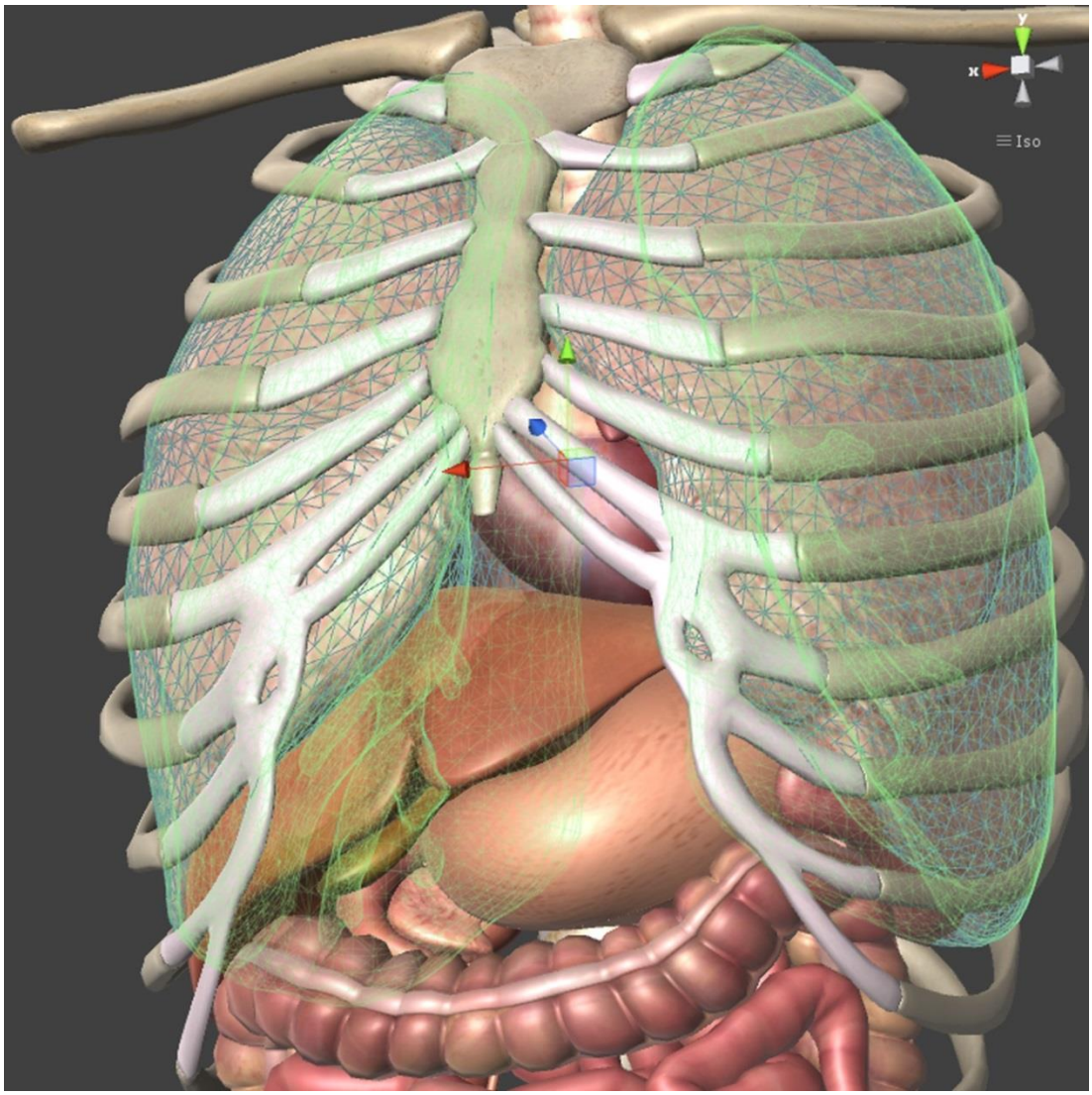


Figure 49. 3D anatomy integrated into the Unity development environment
(© 2016, Douglas A. Nelson Jr.)

The male 3D torso anatomy model was purchased from Zygote (Zygote Media Group, Inc, [102]). The model was modified and animated by a contracted artist (Jeremy Tabor, [103]) and optimized for programmatic control within Unity. Additional 3D models and animations were created by the

contracted artist for the esophagus, trachea, animated breathing lungs, as well as an animated endotracheal tube with inflatable cuff on the end. A separate 3D model of the heart with heartbeat animation was purchased from TurboSquid (Human Heart Animated, Product ID 441382, TurboSquid, Inc., New Orleans, LA, USA; originally downloaded 22-March-2011) and was integrated with the Zygote anatomy by the contracted artist, before both models were imported into Unity. The resulting 3D anatomy in Unity allowed programmatic control of respiration and heartbeat animations, an animation for insertion of an endotracheal tube into the trachea or esophagus, and an animation for inflation and deflation of the endotracheal tube's cuff.

To meet the design requirement of modularity and expandability, the StrangeIoC software architecture was chosen to support development. StrangeIoC integrates with the Unity platform, is written in C#, and supports rapid development [104]. The architecture utilizes an inversion-of-control framework, specifically dependency injection, to decouple the software modules from strict dependence on one another. Dependency injection allows for modular development, and updates within the software, while preserving overall functionality [105]. Using StrangeIoC and its binding framework, data and instructions can be passed seamlessly between BodyExplorer modules, and even provides functionality for cross-network messaging. Cross-network messaging is important for synchronization between the main BodyExplorer laptop, the auxiliary display, and any other future device that is incorporated into the system architecture. Furthermore, the cross-network messaging could permit the individual Sensor subsystems to be wirelessly connected within the system architecture.

To illustrate the importance of the functionality provided through integrating StrangeIoC into the BodyExplorer system architecture, an example of menus will be described. The BodyExplorer integrates an auxiliary display on a wirelessly connected touchscreen tablet PC with

the primary laptop that reads sensor data and controls the projected anatomy. Functionality exists for switching between showing or hiding anatomy labels within menus both on the auxiliary display (**Figure 50, left**) and on the projected anatomy (**Figure 50, right**).

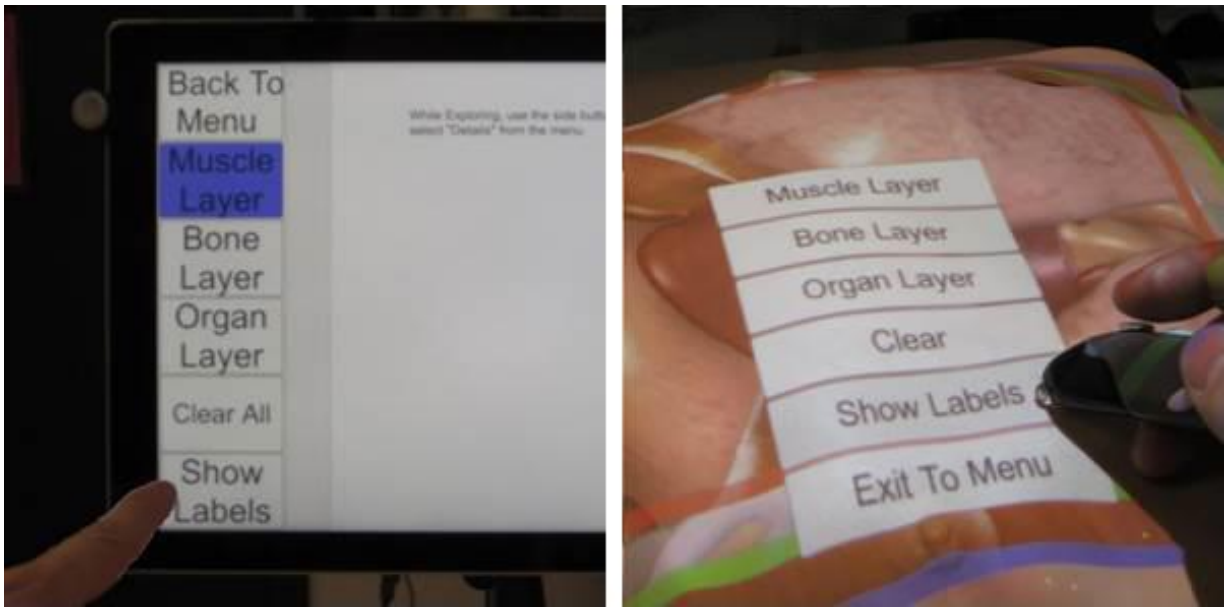


Figure 50. Show Labels menu selection on wireless auxiliary display (left) vs on projected anatomy (right) (© 2016, Douglas A. Nelson Jr.)

When the user selects from one menu, the other menu should know about the change and switch the label on the menu to reflect the change. If the user selects “Show Labels” on the projected anatomy menu, then the “Show Labels” button on the auxiliary display should know of the selection, both menus should change the menu label to be “Hide Labels”, and the functionality within the projected anatomy should update to highlight and show the labels to organs on the body when they are pointed to by the user (**Figure 51**).

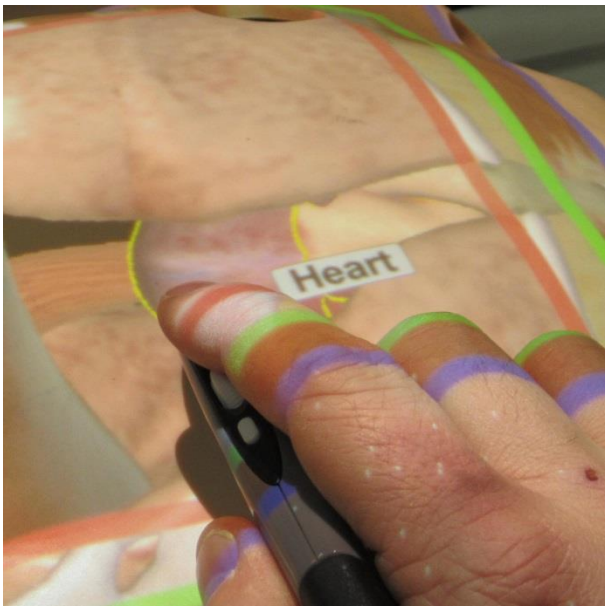


Figure 51. Show Labels functionality on the body
(© 2016, Douglas A. Nelson Jr.)

In this example, we have three things that need to update when a menu item is selected:

1. the selected menu item needs to update its label,
2. the menu item corresponding to the selected menu item on the other networked device (auxiliary or laptop) needs to update its label, and
3. the underlying program functionality needs to change to provide labeling of organs when pointed to by the user.

Using StrangeIoC, a global signal is created for "Toggle Labels". Whenever either menu selection occurs (auxiliary or laptop), the "Toggle Labels" signal is dispatched by the architecture across the BodyExplorer network. The auxiliary display menu, projected laptop menu, and the code for showing/hiding labels all have functionality that executes when the "Toggle Labels" signal is received.

While the previous example for synchronization between and across menus may seem trivial, the benefits to modularity are immense. By using this architecture, any other future

additions to the system that need to update when a “Show Labels” button is pressed can listen for the “Toggle Labels” signal to be dispatched, without requiring modifications to preexisting code. Furthermore, if in the future, a “Show Labels” button is selected on other networked devices or anywhere else within the BodyExplorer system architecture, the selection only has to provide functionality for dispatching the “Toggle Labels” signal and all other functionality for toggling the labels will be preserved.

The synchronization is not limited to menu and user interface elements. Cross-network signaling and synchronization is also possible for passing data from the Sensor subsystems, driving the Automated Instruction subsystem, and updating the Display subsystem. Furthermore, incorporating the StrangeIoC architecture into the BodyExplorer system architecture permits future modifications to the software to occur rapidly, leveraging previous development efforts, and without breaking preexisting, working functionality in the system.

4.0 INTEGRATED SYSTEM VERIFICATION

The following sections describe the verification of system performance for the Learning Module Sensor subsystems against the engineering requirements defined in **Section 0**. The verification testing for the Cricoid Pressure Learning Module Sensor subsystem and the Medication Administration Learning Module Sensor subsystem are included in this section.

4.1 CRICOID PRESSURE LEARNING MODULE

As part of the development for BodyExplorer, we created the Cricoid Pressure Learning Module. The Learning Module is a quantitative method for training the technique of applying cricoid pressure. Unlike other training methods for applying cricoid pressure, our method is integrated within a FBM, allowing users to practice applying cricoid pressure in an integrated fashion with a team of clinicians. The user can practice applying cricoid pressure while another clinician intubates the mannequin, all while under the added time constraint of the mannequin being without oxygen.

4.1.1 Load Testing Apparatus

After the modifications were made to integrate the Cricoid Pressure Learning Module into the mannequin's neck, the system performance with respect to the engineering requirements was verified. A load testing apparatus was constructed to apply loads to the neck area of the mannequin in a repeatable fashion. The apparatus is shown in **Figure 52**. Four adjustable legs on the platform

allow the apparatus to be positioned above the mannequin's head, neck, and shoulders while the mannequin is supine on a flat surface. A center slot on the surface of the apparatus is aligned to the midline of the mannequin's neck, and the slot allows a positioning stage to be moved superiorly and inferiorly along the midline. A linear bearing is mounted in the middle of the positioning stage. The linear bearing ensures vertical loading of test weights onto the neck area while minimizing friction. The loading platform assembly consists of:

1. a loading platform where test weights can be added,
2. a centering rod to keep the test weights centered on the loading platform,
3. a stainless steel vertical shaft to translate the force from the loading platform, through the linear bearing, and to the neck area, and
4. an aluminum angle attached to the mannequin-side of the vertical shaft to simulate the angled fingers that apply force to the neck during cricoid pressure.

The end of the aluminum angle that contacts the mannequin's neck is coated in two layers of foam to soften the contact area like fingertips and to prevent possible tears in the neck skin. The positioning stage can be positioned over the cricoid or thyroid cartilage areas for repeatable load testing trials.

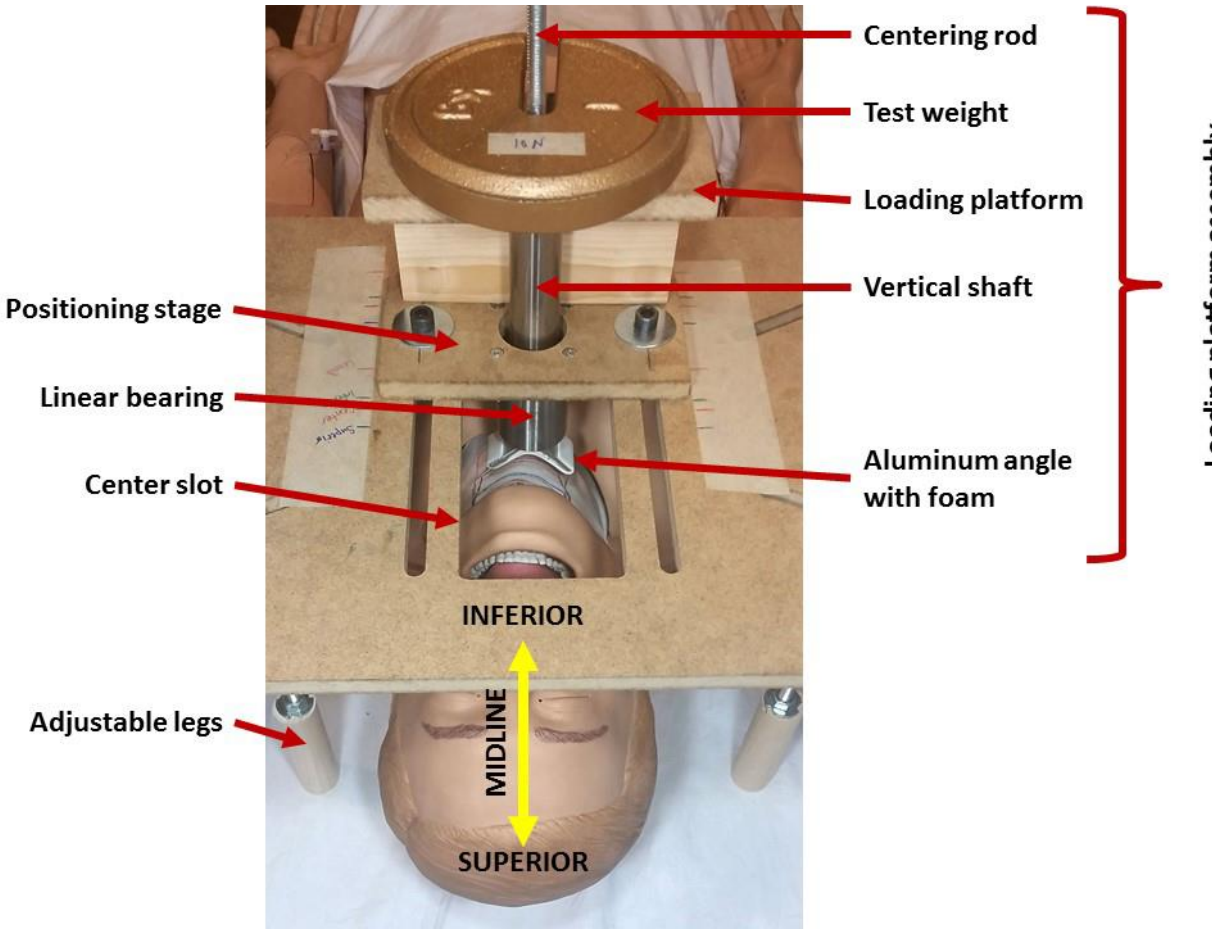


Figure 52. Load testing apparatus for verifying cricoid pressure sensor module performance. (© 2016, Douglas A. Nelson Jr.)

Load testing trials using the apparatus were performed to assess the accuracy and reliability of the Cricoid Pressure Learning Module Sensor subsystem. The accuracy and reliability of the subsystem refer to the subsystem's ability to:

1. properly recognize when a user is touching the thyroid or cricoid cartilage, and
2. properly measure the applied force when a user is pressing the cricoid cartilage in the anterior/posterior direction.

A series of experiments were performed to verify subsystem performance. The following sections describe the experimental methods, results, and discussion.

4.1.2 Experiment 1: Touch Location Accuracy

4.1.2.1 Methods

Per engineering requirement (1), "the system should detect and differentiate whether a user is pressing on the thyroid or cricoid cartilage when at least 10N of force is applied to each structure." The load testing apparatus was used to position the stage and loading platform above 4 regions on the neck's midline, as shown in **Figure 53**:

1. cricoid,
2. inferior thyroid,
3. center thyroid, and
4. superior thyroid.

Multiple regions of the thyroid cartilage were included in the testing because the thyroid cartilage has larger surface area than the cricoid cartilage, and the method for measuring touch relies upon touching force to be translated through the skin vacuum-cast plate to the FSRs positioned between the plate and 3D printed cartilage. Due to this measurement method, touch detection accuracy at the extremities of the thyroid is of interest.

In order to ensure repeatable positioning of the stage and loading platform above the 4 neck regions throughout testing, the neck and surface of the load testing apparatus was colorfully outlined demarking each region's boundaries (**Figure 53)**. With the stage and loading platform centered over a neck region, 1.0 kg was applied to the loading platform—an applied force to simulate a meaningful touch. The loading platform was lowered onto the neck region, and the cricoid and thyroid sensor data was recorded over 5 seconds while the loading platform was in contact with the neck region. For each of the 4 neck regions, 50 repetitions of loading were performed. The order of loading on each neck area was randomized at runtime by the data

acquisition software to prevent ordering bias. The expected result was that the cricoid and thyroid sensor data would accurately detect meaningful touch for all neck regions.

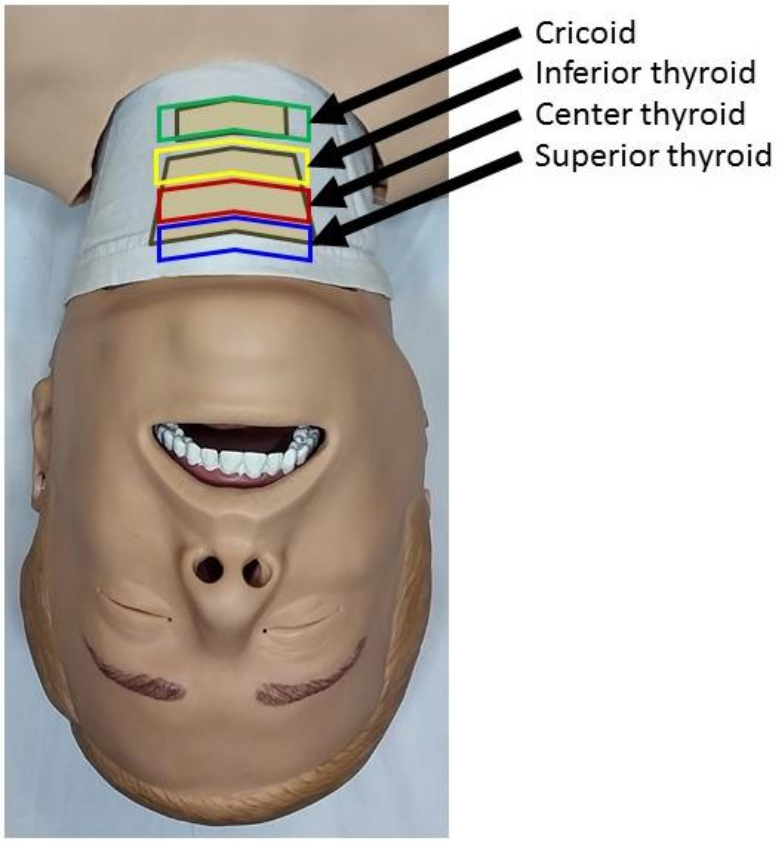


Figure 53. Neck regions where "touching" loads were applied using the load testing apparatus to assess system accuracy at distinguishing between touching cricoid or thyroid cartilage
(© 2016, Douglas A. Nelson Jr.)

4.1.2.2 Results

The data from the 200 trials are shown in **Table 1.** The cricoid and thyroid sensor data measured touch in all trials when load was applied to the cricoid, center thyroid, and superior thyroid regions. The cricoid and thyroid sensor data measured touch on ***both*** the cricoid and thyroid in all trials when load was applied to the inferior thyroid region.

Table 1. Frequency of measured touches at each neck region

Neck Region Touched	Measured Cricoid Touched	Measured Thyroid Touched
Cricoid	50	0
Inferior thyroid	50	50
Center thyroid	0	50
Superior thyroid	0	50

4.1.2.3 Discussion

The results of the cricoid and thyroid sensor data from the cricoid, center thyroid, and superior thyroid regions met expectations and accurately distinguish touch in all trials. However, the results from the inferior thyroid region did not fully meet expectations. When the load was applied over the inferior thyroid region, sensor data conditions for both cricoid touched and thyroid touched were met, and the system detected that both pieces of cartilage were being touched. While it is possible that the end of the loading platform in contact with the neck was indeed in contact with both the cricoid and thyroid cartilage when positioned over the inferior thyroid region, it is also possible that the applied force is being distributed through the skin to the closely-adjacent cricoid cartilage. Regardless of the cause, this observation could be useful in providing additional location detection resolution. For example, the system does not currently implement a method for detecting pressure to the cricothyroid membrane located between the thyroid and cricoid cartilages., It may be possible to detect when a user is touching the cricothyroid membrane using magnitudes of the responses of the FSRs on the cricoid and thyroid cartilage that measure touch. Conceivably, if magnitude of thyroid FSRs is high, then the user is touching thyroid; if magnitude of cricoid FSRs is high, then the user is touching cricoid; if magnitude of cricoid and thyroid FSRs are higher than baseline, but lower than the magnitude set for detecting touch, then the system could infer that the

user is touching the cricothyroid membrane without requiring the addition of another sensor to directly measure the cricothyroid membrane.

Consequently, the location detection resolution can provide instructions to a user for applying cricoid pressure. The software logic could be updated to default to identifying thyroid touched in all cases where the system detects touching on both the thyroid and cricoid cartilage. This functionality is valid because the instructions are to "move down on the neck" if thyroid touching is detected when cricoid touching is desired.

In relationship to broader methods for measuring a user's touch on different areas of the body during training, the same concept for utilizing fusion of sensor data to interpret actions while minimizing the required number of physical sensors could have rich implications, both from a practical sense in reducing cost of the system and in the level of inference that an automated instruction system could make on the actions of the user.

4.1.3 Experiment 2: Force Measurement Calibration

4.1.3.1 Methods

To assess engineering requirements (2) and (3), the load cell for quantifying applied force to the cricoid cartilage needed to be calibrated. To perform calibration, the load testing apparatus was used to position the stage and loading platform above the cricoid region, centered midline on the mannequin's neck (**Figure 52**). Test weights were applied to the loading platform ranging from 0-5.0 kg (49N) in 1.0 kg (9.8N) increments, for a total of 6 different test weight conditions. Each test weight condition was repeated 3 times for a total of 18 trials. Trials were randomized at runtime by the data acquisition software to prevent ordering bias.

During each trial, the loading platform was loaded with the appropriate test weight(s). Then, the loading platform was lowered onto the cricoid region, and data was recorded for 60 seconds (trial, time, applied weight, repetition, load cell measurement). After completion, the loading platform was raised and held in the raised position while test weights for the next trial were applied to the loading platform. The data were plotted and analyzed using Microsoft Excel and IBM SPSS 24, respectively. The load cell measurements are reported in Arduino Units (units) because there is not a need to convert them to voltage values for analysis in the software. (For conversion, 1 Arduino Unit is equal to 5 volts / 1024 units—due to the 10-bit analog to digital converter—or roughly 4.9 mV / unit.) All further results reporting load cell measurements will be reported in "units." The expected result was a linear relationship between applied weight and load cell measurement. Assuming a linear relationship, the measured weight of the loading platform was added to the applied test weight to perform regression resulting in an equation to convert load cell measurement to total applied weight. As such, instead of traditionally performing the analysis with load cell measurement as the outcome variable and total applied weight as the predictor, the roles were reversed during analysis to result in a prediction of total applied weight for given load cell measurements.

4.1.3.2 Results

Descriptive statistics were calculated on the data from the 18 trials in SPSS. The results are summarized in **Table 2.** The assumptions for linear regression were verified in SPSS and a linear model was determined. The model significantly predicts Total Applied Weight values ($p<0.001$) with a correlation coefficient, R^2=99.2%, by the equation:

$$(Total\ applied\ weight, N) = 0.177 * (Load\ cell\ measurment, Arduino\ units) - 15.930$$

A graph of the Load Cell Measurements and linear regression model was plotted in Excel and is shown in **Figure 54**.

Table 2. Descriptive statistics for force measurement calibration experiments

			Load Cell Measurements	
Applied Test Weight (kg)	**Total Applied Weight (kg)[1]**	**Total Applied Weight (N)[2]**	**Mean ± SD**	**[Minimum, Maximum]**
0.0	0	0	105.1 ± 0.39	[104, 107]
1.0	1.855	18.179	182.7 ± 1.28	[180, 185]
2.0	2.855	27.979	236.8 ± 1.52	[234, 239]
3.0	3.855	37.779	299.8 ± 2.20	[296, 303]
4.0	4.855	47.579	361.7 ± 1.47	[358, 364]
5.0	5.855	57.379	421.2 ± 0.71	[419, 423]

[1] Total Applied Weight includes the weight of the loading platform (0.855 kg)
[2] Conversion of 1 kg = 9.8 N used
[3] Load Cell Measurements in "Arduino Units," or "units"

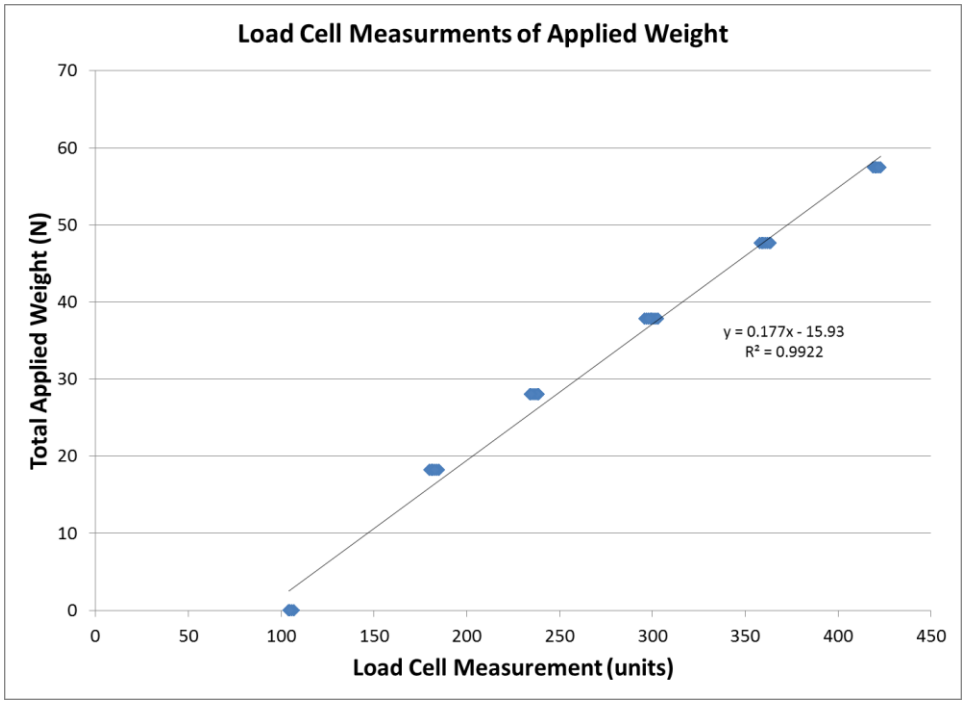


Figure 54. Calibration curve for relating load cell measurement to applied weight on the cricoid cartilage.

4.1.3.3 Discussion

These results support that mounting the load cell in the neck as presently mounted retains the linear relationship between load cell measurement and total applied weight on the cricoid region. The standard deviation of the mean load cell measurements, while small, are most likely derived from both the published accuracy of the load cell (±1% of 25lb, or approximately ±1.1N) and the bit quantization of the analog to digital converter of the Arduino. The maximum range of load cell measurements during this experiment was 7 Arduino units (applied test weight = 3.0 kg), which also had the maximum standard deviation from the mean of 2.2 Arduino units. Conservatively, 3 standard deviations out from the mean would be ±7 Arduino units. Using the model's slope, a change of ±7 Arduino units would result in ±1.2 N—which falls within the engineering design goal of ±2 N, but still needs to be verified after incorporating the calibration into the software. Furthermore, since the data from each trial were recorded over 60 seconds, these results support the engineering design goal that the load cell measurements do not drift more than ±2 N when constant force is applied over that period.

4.1.4 Experiment 3: Force Measurement Accuracy

4.1.4.1 Methods

To verify measurement accuracy and repeatability after calibration, the force loading procedure from Experiment 2 was repeated across the same range of test weights, but in 0.5 kg increments and for 30 second trials instead of 60. The load testing apparatus (**Figure 52**) was used to position the stage and loading platform above the cricoid region, centered midline on the mannequin's neck. Test weights were applied to the loading platform ranging from 0-5.0 kg (49N) in 0.5 kg (4.9N) increments, for a total of 11 different test weight conditions. Each test weight condition was

repeated 10 times for a total of 110 trials. Trials were randomized at runtime by the data acquisition software to prevent ordering bias.

During each trial, the loading platform was loaded with the appropriate test weight(s). Then, the loading platform was lowered onto the cricoid region of the mannequin's neck and data was recorded for 30 seconds (trial, time, applied weight, repetition, load cell measurement). After completion, the loading platform was raised and held in the raised position while test weights for the next trial were applied to the loading platform. The data were plotted and analyzed using Microsoft Excel and IBM SPSS 24, respectively.

4.1.4.2 Results

Descriptive statistics were calculated on the data from the 110 trials in SPSS. The results are summarized in **Table 3**. The mean absolute error across all trials was 1.30 ±0.77 N, with a maximum absolute error of 3.01 N corresponding to when no weight was applied. A graph of the calculated versus applied weight was plotted in Excel and is shown in **Figure 55** plotted against the ideal results—a line with slope of 1 and intercept of 0. A graph of the calculation error for each trial was plotted in Excel and is shown in **Figure 56**. The largest error is found when no weight is applied to the cricoid.

Table 3. Descriptive statistics for loads applied to the cricoid cartilage

			Calculated Weight from Model (N)	Absolute Error |Calculated – Applied| (N)		
Applied Test Weight (kg)	**Total Applied Weight (kg)[1]**	**Total Applied Weight (N)[2]**	**Mean ± SD**	**Mean**	**Minimum**	**Maximum**
0.0	0	0	2.88 ± 0.09	2.88	2.66	3.01
0.5	1.355	13.279	12.05 ± 0.22	1.23	0.89	1.77
1.0	1.855	18.179	16.40 ± 0.31	1.78	1.19	2.25
1.5	2.355	23.079	21.06 ± 0.15	2.01	1.66	2.37
2.0	2.855	27.979	26.35 ± 0.31	1.63	1.08	2.31
2.5	3.355	32.879	31.98 ± 0.26	0.90	0.49	1.55
3.0	3.855	37.779	37.33 ± 0.44	0.45	0.08	1.49
3.5	4.355	42.679	42.70 ± 0.33	0.28	0.02	0.69
4.0	4.855	47.579	47.98 ± 0.47	0.51	0.03	1.45
4.5	5.355	52.479	53.57 ± 0.31	1.10	0.44	1.51
5.0	5.855	57.379	58.93 ± 0.31	1.56	0.15	2.09
Mean Absolute Error ± Std. Dev.: 1.30 ± 0.77 N						

1 Total applied weight includes the weight of the loading platform (0.855 kg)
2 Conversion of 1 kg = 9.8 N used

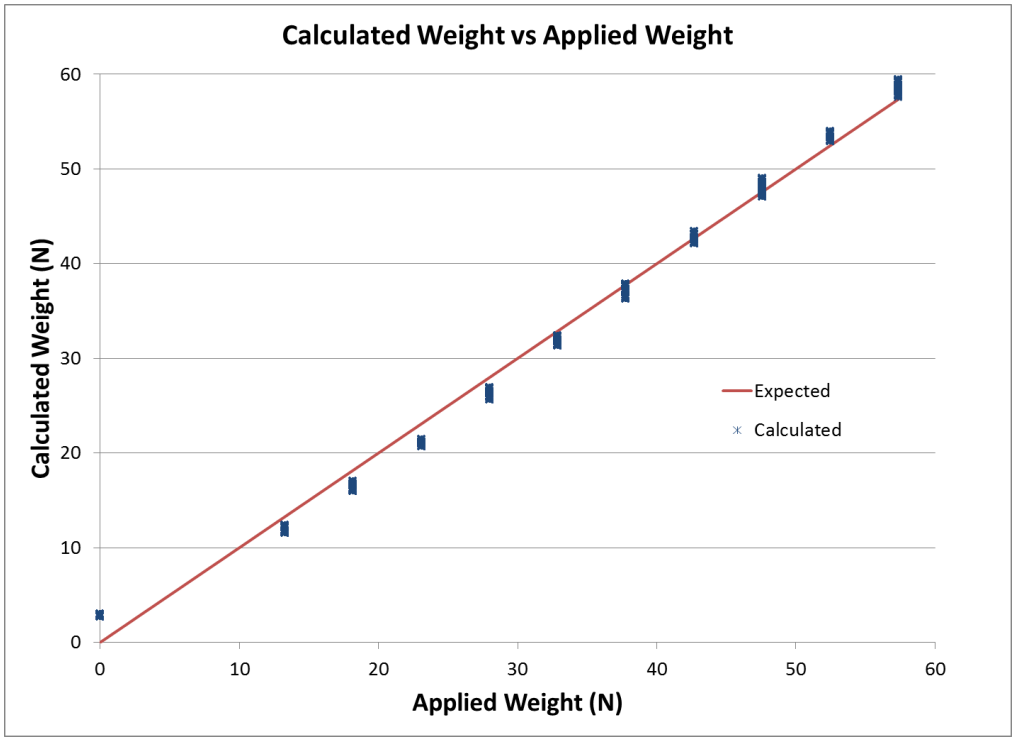


Figure 55. Curve showing linear relationship between the calculated and applied weight on the cricoid cartilage

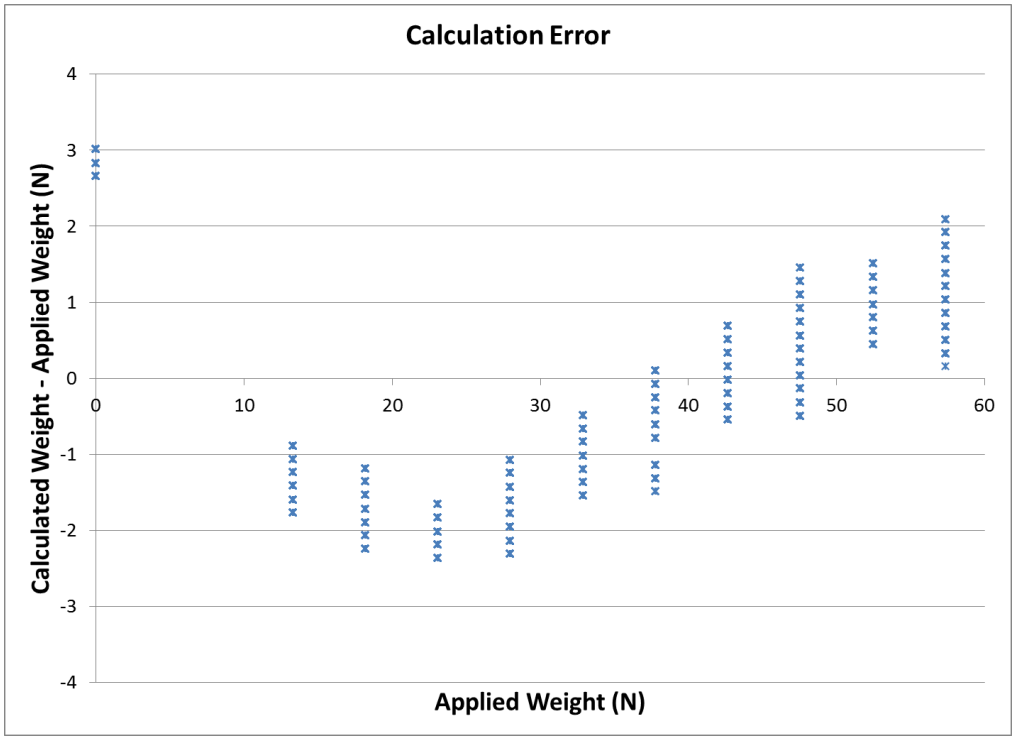


Figure 56. Calculation error versus applied weight on the cricoid cartilage

4.1.4.3 Discussion

The mean absolute error for the calculation of applied force during application of cricoid pressure is 1.3±0.77 N over the range of 0 to 60 N. The mean absolute error for the calculation improves to 1.1±0.60 N when the model offset values are removed at calculations when no force is applied. This updated calculation for mean absolute error is more appropriate to report because the software logic in the Cricoid Pressure Learning Module only reports force measurements to the user when touch is detected on the cricoid cartilage, i.e. when force is applied to the cricoid.

From the graph of calculation error in **Figure 56**, the model for calculating applied force tends to overestimate the applied force when the applied force is less than 10 N or over 40 N, and tends to underestimate the applied force when the applied force is between 10 N and 40 N. The instructions in the Cricoid Pressure Learning Module are to apply force between 30-40 N. The worst-case scenarios based upon the calculated mean absolute errors would mean that a user could be instructed to apply more pressure when they are applying between 30-32 N. While this situation is not ideal, the quantitative feedback guides the user towards the middle of the recommended application range and provides feedback that is not possible using current training methods.

There are limitations to this experimental design. This experiment was performed in a controlled environment, applying test weights in a repeatable fashion. Future studies could assess measurement accuracy with the head and neck in different positions. Other studies could also be performed with participants applying cricoid pressure while using a secondary force-measurement device such as a hydraulic systems with tubing connected to a pressure transducer [76, 83] or weighing scales [82].

4.2 MEDICATION ADMINISTRATION LEARNING MODULE

As part of development for BodyExplorer, and to address reducing preventable medication errors through improved training methods, we developed a novel method for recognizing the administration of simulated intravenous medications. Unlike current technologies utilized for detecting simulated intravenous medications in the training environment, our method identifies the administered simulated intravenous medication based upon a property of the administered medication itself. The following sections describe baseline assessment of accuracy and understanding of the limits for the Medication Administration Learning Module Sensor subsystem according to the design specifications. The parameters assessed were:

1. accuracy of medication identification, and
2. accuracy of measured volume.

4.2.1 Factorial Experiment

4.2.1.1 Methods

Factorial experiments were performed in order to establish a baseline measure of accuracy and to understand the limits of the system. The two factors analyzed in this experiment were salt concentration (the "medication ID") and volume to administer. The experiment aimed to understand the effect that varying each factor had on the outcome of properly identifying an administered medication and assess the accuracy of properly calculating the injected volume. Because salt concentration and administered volume are both continuous variables, the experimental design was limited to 5 levels of salt concentration (or 5 medication IDs) and 4 levels

of administered volumes. By limiting the continuous factors to ordinal, a 5x4 full factorial experiment was performed. The factors are listed in **Table 4**.

Table 4. Factors and levels for factorial experiments

Salt Concentration (g/L)	Medication ID
0.0	Saline
0.3	Propofol
1.0	Epinephrine
2.0	Esmolol
3.0	Succinylcholine

Volume to Administer (mL)
1.0 mL
2.0 mL
5.0 mL
8.0 mL

Additional constraints around ordering of injections were included in the experimental design. In the clinic, administration of an intravenous medication is followed by a saline flush in order to clear the medication from the IV line and fully administer the dose to the patient. During testing, the 0.0 g/L solution was used for the saline flush (hereafter, "saline" refers to the 0.0 g/L salt solution). For consistency, the volume of the saline flush was always the same volume as the administered medications (hereafter, "non-saline medications", or just "medications," refers to the solutions with 0.3, 1.0, 2.0, and 3.0 g/L salt concentration). This method also allowed observation of the effect that volume of saline flush had on the outcome of properly identifying the flush after administration of each medication.

As a measure to minimize procedural errors during factorial experiments, the factors were blocked on the volume to administer. For example, all the trials for 2.0 mL were performed in sequence during a block. Then, all the trials for 5.0 mL were performed in sequence for another block, and so on and so forth until all 4 blocks were completed. Each medication was injected and followed by a saline flush. This paired process was repeated 10 times during a block for each

medication. Randomization of trial order during each block was performed to account for potential ordering bias.

For all trials, injections were administered into the IV port of BodyExplorer by a trained research assistant, and the setup is shown in **Figure 57**. The BodyExplorer software ran as normal, with the auxiliary display managing the randomization and presentation of trial sequencing. Baseline detection windows for identifying each medication were included in the software calibration file and remained constant throughout all trials. The detection window thresholds are shown in **Table 5**. These thresholds were determined by finding the resting conductivity values after injecting each simulated medication, then expanding the detection window symmetrically above and below the resting value, ensuring that adjacent detection windows did not overlap while leaving a buffer between detection windows. The fluid outflow collection container was placed on a scale (±0.1g, Scout Pro SP4001, Ohaus Corporation, Pine Brook, NJ, USA) to measure the mass of the collected outflow after each injection. The difference between the pre-injection and post-injection mass recordings was used to determine the volume of the injection by multiplying by the density of water as 1.0 g/mL (approach rationale in **Appendix B**). The collection container was emptied after each block of injections.

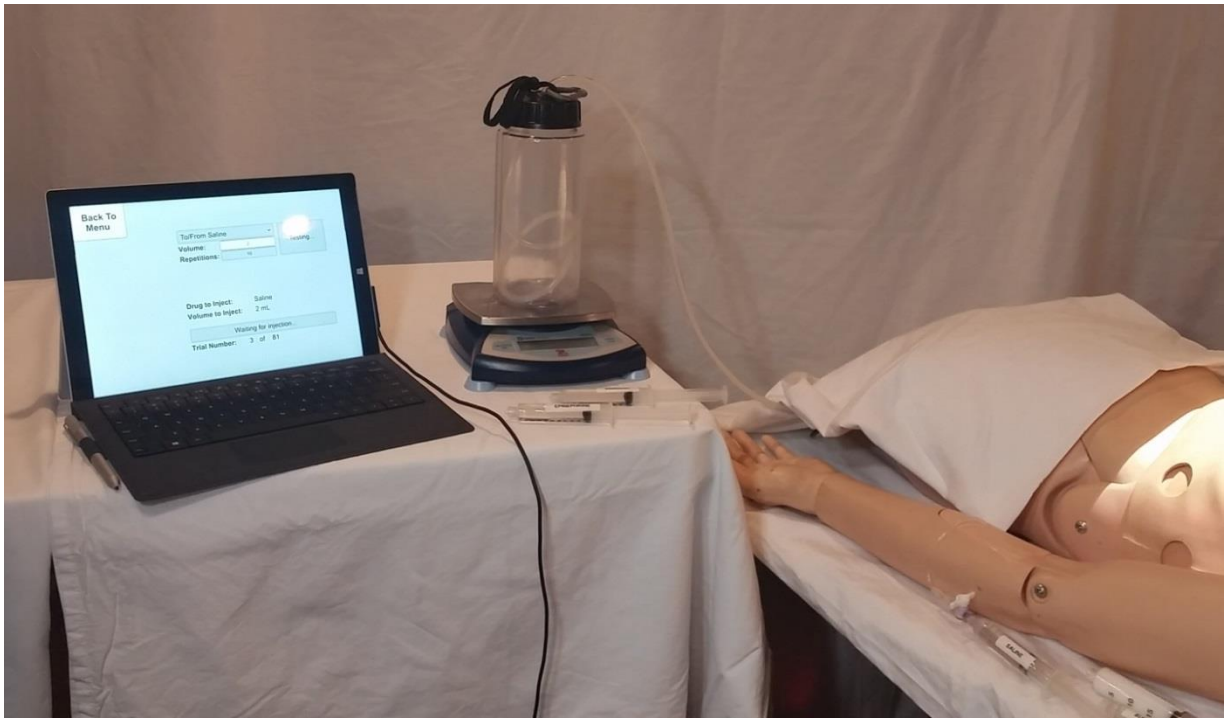


Figure 57. Experimental setup for factorial testing
(© 2016, Douglas A. Nelson Jr.)

Table 5. Detection Windows during factorial testing

Medication ID	Resting Threshold (Arduino units)	Offset (Arduino units)	Detection Window [Min, Max] (Arduino units)
Saline	20	0	[0, 40]
Propofol	112	35	[77, 147]
Epinephrine	250	50	[200, 300]
Esmolol	450	50	[400, 500]
Succinylcholine	600	85	[515, 685]

The data files for each trial were recorded on both the auxiliary display and the primary BodyExplorer computer. The primary computer recorded the raw sensor data throughout the entire experiment. The auxiliary display recorded a summary of the individual trial's data that included trial classification data and calculated data. The classification data included:

1. the trial number,
2. the time that the trial's injection ended (which was matched with the timestamp in the raw sensor data recorded on the primary computer in order for the injection data summary to be matched with the raw sensor data),
3. the expected medication ID to be identified,
4. the expected volume to be calculated,
5. the repetition number, and
6. the identification of the previously administered medication.

The calculated data included:

1. the identified medication ID,
2. the calculated volume, and
3. the conductivity sensor value at the completion of injection.

The actual volumes measured by the scale for each trial were added to the data file with the summary data from each trial at the end of the experiment for comparison with the calculated volumes from the system. The data were analyzed using SPSS (v24.0, IBM).

Ideally throughout these experiments, the system should properly identify each injected medication and the subsequent saline flush. Since the calculated volume from the software is determined by multiplying time increments by measured flow rate, the accuracy of calculated volume is expected to match the actual volume injected (as measured by the scale) within the limits of the flow meter (±1% full-scale; Liquid Flow Sensor, Model 101, McMillan Company, Georgetown, TX, USA).

The results and discussion are split into 3 parts: 1) proper identification of non-saline medication injections, 2) proper identification of saline flush injections, and 3) accuracy of calculated volumes.

4.2.1.2 Proper Identification of Non-Saline Medication Injections

There were a total of 160 trials of non-saline medication injections performed. For each of the 4 medications, 10 repetitions of injections were performed at each of the 4 levels of volume, as described in **Section 4.2.1.1**.

Results

All 4 of the non-saline medications were properly identified across all trials, regardless of volume. Descriptive statistics for all injection trials of the non-saline medications are shown in **Table 6**. The minimum and maximum conductivity values were all within the detection window for the respective medication (group minima and maxima shown in **Table 7**).

Table 6. Descriptive statistics for conductivity measured during injections of non-saline medications at each level of volume

		Salt Concentration (g/L) Medication ID			
Volume to Administer (mL)	**Statistic**	**0.3 Propofol (Arduino units)**	**1.0 Epinephrine (Arduino units)**	**2.0 Esmolol (Arduino units)**	**3.0 Succinylcholine (Arduino units)**
1.0	Mean ± SD	111.1 ± 2.7	255.1 ± 6.1	419.1 ± 6.4	546.0 ± 10.8
	[Min, Max]	[106, 115]	[242, 261]	[413, 431]	[532, 560]
2.0	Mean ± SD	114.6 ± 0.9	263.2 ± 1.3	435.3 ± 2.2	568.6 ± 3.2
	[Min, Max]	[113, 116]	[261, 265]	[433, 440]	[563, 574]
5.0	Mean ± SD	113.5 ± 0.8	262.3 ± 1.0	435.9 ± 1.6	568.7 ± 1.2
	[Min, Max]	[112, 114]	[260, 263]	[433, 438]	[567, 571]
8.0	Mean ± SD	114.6 ± 0.8	265.5 ± 1.4	441.8 ± 1.3	557.5 ± 1.9
	[Min, Max]	[113, 116]	[263, 267]	[440, 444]	[574, 580]

Table 7. Comparison of measured ranges and detection windows for all volumes

Medication ID	**Salt Conc. (g/L)**	**Measured Conductivity Range [Min, Max] (Arduino units)**	**Detection Window [Min, Max] (Arduino units)**
Propofol	0.3	[106, 116]	[77, 147]
Epinephrine	1.0	[242, 267]	[200, 300]
Esmolol	2.0	[413, 444]	[400, 500]
Succinylcholine	3.0	[532, 580]	[515, 685]

A graph of the detection windows with respect to the trial data for all volumes are shown in **Figure 58**. The whiskers on the plots show the extrema while the black horizontal lines show the mean values. At larger salt concentrations, the means of the trial data drop below the means of the

detection windows, and the trial data are skewed towards the lower extrema of the detection windows.

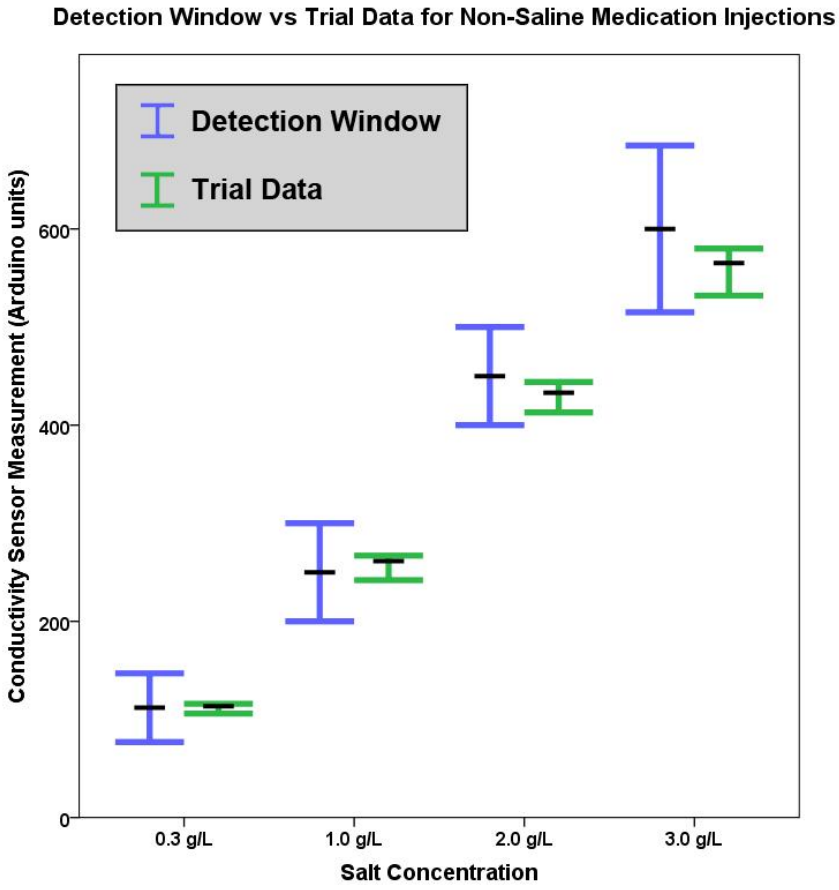


Figure 58. Detection windows vs trial data for non-saline medication injections.

Discussion

The primary goal of this experiment was to assess the accuracy of properly identifying injections of simulated medications. Four medications were identified in all trials when injected after a saline flush. When comparing the measured conductivity ranges across all injection volumes to the detection windows for the respective medications, the minimal differences from the detection

window boundaries were 9 Arduino units from the lower boundary and 31 Arduino units from the upper boundary. This suggests that the detection windows can be further optimized for injections of these medications when following a saline flush. The sequencing for the injections of the medications guaranteed that the medications would always follow a saline flush. This sequencing meant that the conductivity measurements at the start of injection would always be increasing. Thus, the detection algorithm within the software can be modified to shift the detection windows lower when an increasing change in conductivity is measured at the start of injection. This modification to shift the detection windows based upon initial change in conductivity can also allow the width (or range) of the detection windows to be decreased (or "tighter" detection windows). This would potentially allow more simulated medications to be created and be differentially detected within the 0.0 to 3.0 g/L range of salt concentrations.

In addition to optimizing the detection windows, flow rate should be analyzed to assess its contribution to accurate detection. Incorporating flow rate as a third variable into the detection analysis, a model could be developed that instantaneously predicts what simulated medication is being injected based upon the initial injection rate, rate of change in conductivity, and knowing what was previously injected.

This experiment was limited by the injection sequencing—medication injections always followed a saline flush. While this sequencing was designed to follow safe medication administration protocol, it does not fully define all possible cases for detecting injections of simulated medications. As part of further experiments to optimize the detection windows, the other permutations of injection sequencing should be assessed—medication_A followed by medication_A, medication_A followed by medication_B, and medication_B followed by medication_A. These experiments were later performed and are described in **Section 4.2.2**.

4.2.1.3 Proper Identification of Saline Flush Injections

There were a total of 160 trials of saline flush injections performed. A saline flush was performed after each medication injection, as described in **Section 4.2.1.1**.

Results

The results from the saline flush injection trials are summarized in **Table 8**. The saline flush was properly identified in all trials except for trials where the volume to administer was 1.0 mL—highlighted in **Table 8**. The detection algorithm did not properly identify the 1.0 mL saline flush once after following an injection of epinephrine. The algorithm only properly identified the 1.0 mL saline flush once after following an injection of esmolol. The algorithm failed to identify all 10, 1.0 mL saline flushes following injections of succinylcholine.

Table 8. Frequency of properly detecting the saline flush out of 40 flushes for each non-saline medication across all volumes

Medication Being Flushed		**Volume of Saline Flush to Administer (mL)**			
Salt Concentration (g/L)	**Medication ID**	**1.0**	**2.0**	**5.0**	**8.0**
0.3	Propofol	10	10	10	10
1.0	Epinephrine	9	10	10	10
2.0	Esmolol	1	10	10	10
3.0	Succinylcholine	0	10	10	10

Descriptive statistics for the saline flushes are shown in **Table 9**, highlighting the data corresponding to the incorrectly identified saline flushes. The mean conductivity values for saline flushes coming after injections of esmolol and succinylcholine are outside the detection window for saline (0 to 40), while the mean for flushes coming after epinephrine was within the detection window. The conductivity maximum for epinephrine and minimum for esmolol are outside and within the detection window, respectively. Those extremes account for the one failed and one

successful injection identifications for epinephrine and esmolol, respectively, highlighted in **Table 9**.

Table 9. Descriptive statistics of conductivity during saline flushes

		Salt Concentration (g/L) Medication Being Flushed by Saline			
Volume to Administer (mL)	**Statistic**	**0.3 Propofol (Arduino units)**	**1.0 Epinephrine (Arduino units)**	**2.0 Esmolol (Arduino units)**	**3.0 Succinylcholine (Arduino units)**
1.0	Mean	21.4 ± 1.1	30.4 ± 6.5	49.3 ± 10.3	73.0 ± 14.2
	[Min, Max]	[20, 24]	[23, 45]	[24, 64]	[53, 92]
2.0	Mean	17.2 ± 0.3	18.4 ± 0.4	21.2 ± 1.4	24.3 ± 3.3
	[Min, Max]	[17, 18]	[18, 19]	[18, 24]	[21, 32]
5.0	Mean	16.7 ± 0.2	17.0 ± 0.3	17.4 ± 0.3	17.9 ± 0.5
	[Min, Max]	[16, 17]	[16, 18]	[17, 18]	[17, 19]
8.0	Mean	16.9 ± 0.1	16.9 ± 0.1	17.0 ± 0.2	17.2 ± 0.2
	[Min, Max]	[16, 17]	[16, 17]	[16, 18]	[17, 18]

A 3D plot showing the relationship between mean conductivity at completion of saline flush with respect to the medication being flushed and the volume of the flush is shown in **Figure 59**.

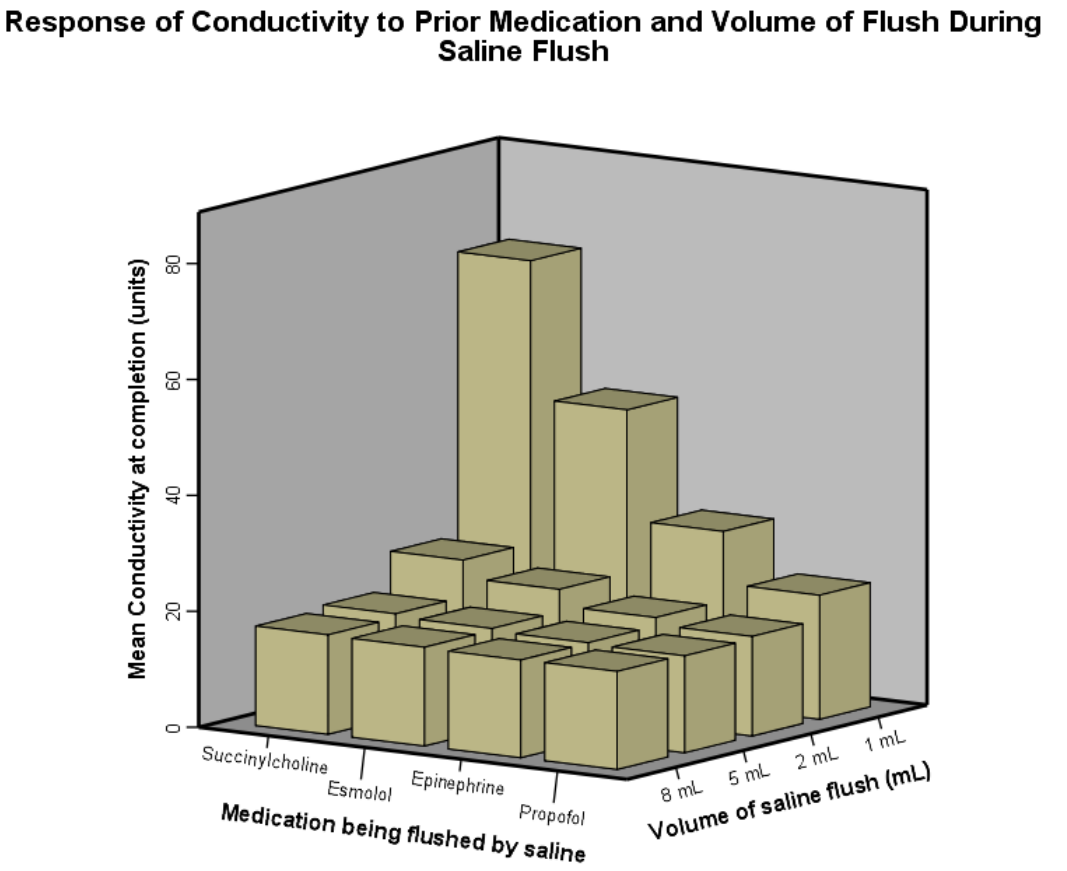


Figure 59. Response of conductivity to prior medication and volume during saline flush

To further analyze the effects of the injection parameters on proper identification of the flush, a one-way Welch ANOVA was conducted to determine if the conductivity of the injection was different for groups of volume, splitting the data over groups of the medication being flushed. The difference in conductivity for levels of volume for each medication flushed was statistically significant. Results from the one-way Welch ANOVA are shown in **Table 10**.

Table 10. Results of the one-way Welch ANOVA on conductivity differences

Medication Being Flushed by Saline		Welch's F-Statistic
Salt Concentration (g/L)	**Medication ID**	
0.3	Propofol	$F(3, 18.342) = 56.658$, $p \leq 0.0005$*
1.0	Epinephrine	$F(3, 15.896) = 51.517$, $p \leq 0.0005$*
2.0	Esmolol	$F(3, 17.008) = 60.089$, $p \leq 0.0005$*
3.0	Succinylcholine	$F(3, 15.870) = 66.977$, $p \leq 0.0005$*
*Statistically significant at the $p < 0.05$ level		

Group means were assessed for statistically significant differences using the Games-Howell post hoc test. There were statistically significant differences between mean conductivities for 1.0 mL saline flushes when compared with saline flushes of all other volumes for flushes after each medication. The results of these comparisons are shown in **Table 11**.

Table 11. Results of the Games-Howell post hoc tests on differences in mean conductivity between saline flushes of 1.0mL and all other flush volumes for flushes after each medication

Salt Concentration Medication ID	Compared Volume	Mean ± SD	Mean ± SD for 1mL	Difference in Means ± Std. Error	p-value
0.3 g/L Propofol	2 mL	17.1 ± 0.3	21.4 ± 1.1	4.2 ± 0.4	$p \leq 0.0005$*
	5 mL	16.7 ± 0.2		4.7 ± 0.4	$p \leq 0.0005$*
	8 mL	16.9 ± 0.1		4.5 ± 0.4	$p \leq 0.0005$*
1.0 g/L Epinephrine	2 mL	18.4 ± 0.4	30.4 ± 6.5	12.0 ± 2.0	$p = 0.001$*
	5 mL	17.0 ± 0.3		13.4 ± 2.0	$p \leq 0.0005$*
	8 mL	16.9 ± 0.1		13.5 ± 2.0	$p \leq 0.0005$*
2.0 g/L Esmolol	2 mL	21.2 ± 1.3	49.3 ± 10.3	28.1 ± 3.3	$p \leq 0.0005$*
	5 mL	17.4 ± 0.3		31.9 ± 3.3	$p \leq 0.0005$*
	8 mL	17.0 ± 0.2		32.3 ± 3.3	$p \leq 0.0005$*
3.0 g/L Succinylcholine	2 mL	24.3 ± 3.3	73.0 ± 14.2	48.8 ± 4.6	$p \leq 0.0005$*
	5 mL	17.9 ± 0.5		55.1 ± 4.5	$p \leq 0.0005$*
	8 mL	17.2 ± 0.2		55.9 ± 4.5	$p \leq 0.0005$*
*Statistically significant at the $p < 0.05$ level					

The difference in means increase as volume decreases and as salt concentration of the medication being flushed increases. These trends are shown in **Figure 60**; the means of the saline flush at 1.0 mL were subtracted by the means at other volumes to normalize the mean conductivities for comparison on one graph. The comparisons depicted in the graph and the data from the table show that the mean conductivities varied by no more than 10 units for flushes of 2.0 mL and greater; whereas, flushes of 1.0 mL resulted in mean conductivities varying up to 50 units

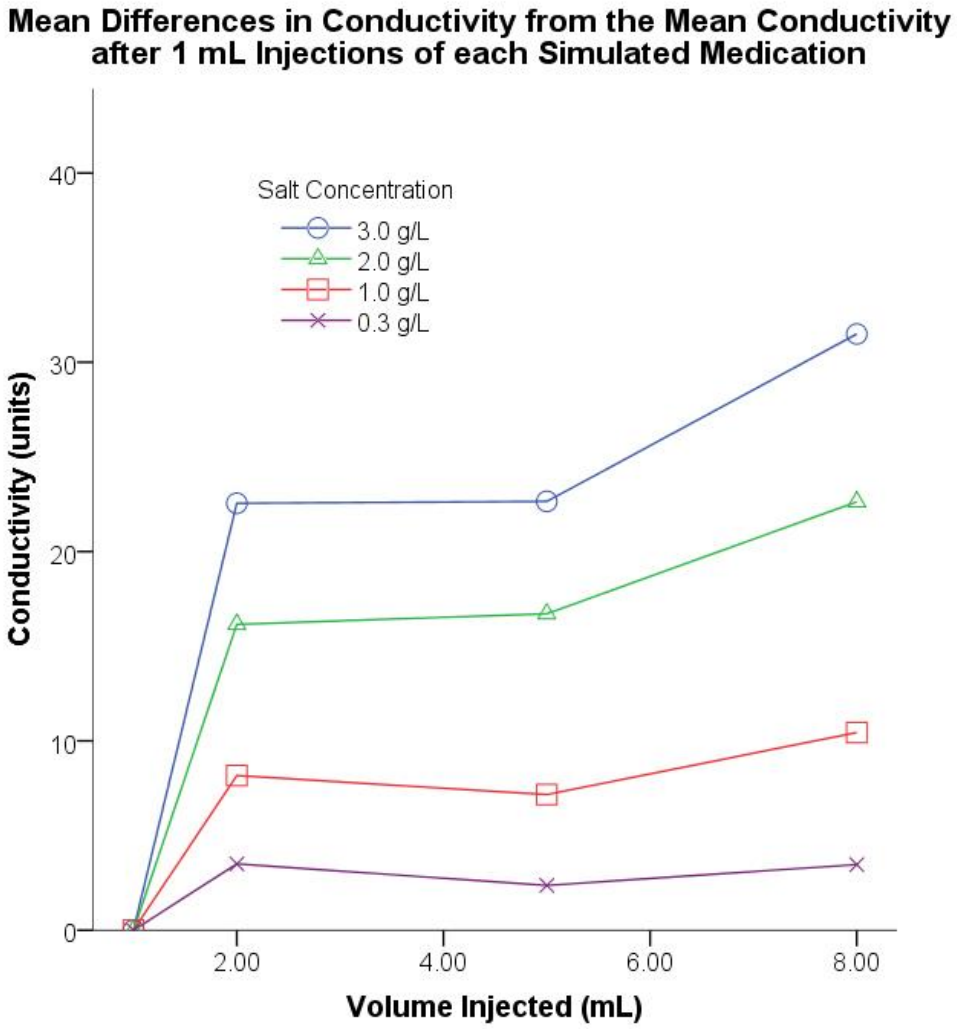


Figure 60. Relationship between mean differences in conductivity after saline flushes

Discussion

The results from the saline flush experiments suggest that as the difference in salt concentration between the solution being flushed and the saline flush increases, more volume of saline flush is necessary for proper identification of the flush. While all injections were properly identified in the previous experiment when medications were injected after a baseline of saline, the same was not true for all injections of the saline flush after medication injections. These findings suggest that there is a limit to the minimal volume necessary during saline flushes to correctly identify the saline flush with the current detection algorithm. All saline flushes of 2.0 mL and greater were properly identified by the current detection algorithm.

In all instances of saline flushes, the salt concentration of the flush was less than the salt concentration of the previously administered medication. Thus, the measured conductivity in these instances would always be decreasing. Similar to methods presented in the previous section's discussion about optimizing the detection windows for increasing conductivities, the detection windows can be optimized for decreasing conductivities, too. The detection algorithm can be modified to shift the detection windows upwards from their baseline values when the detection algorithm identifies that the measured conductivity is decreasing during injection. This modification to the detection algorithm may correct the incorrect identification of 1.0 mL saline flushes observed in this experiment.

Additional experimentation is necessary to further assess the electrical properties of the conductivity sensor design. The sensor design may be retaining salt residue in the sensor after high salt concentration medications are administered. Thus, larger volumes of Saline flush are necessary to remove the salt residues before proper identification can occur. It is also possible that there is electrolyte bias occurring in the solution, causing a charging capacitor effect on the sensor's

electrodes when higher concentrations of salt solution are injected. Even though the sensor's circuit applies an alternating current across the sensor's electrodes, it is possible that the signal is not precisely centered about zero. Such bias would cause the positive sodium ions to collect near the negatively biased electrode and the negative chloride ions to collect near the positively biased electrode. If the capacitor effect is found to be supported, modeling the conductivity sensor response as a charging or discharging capacitor would potentially result in another method for instantaneously identifying administered medications.

4.2.1.4 Accuracy of Measured Volumes

There were a total of 320 trials of injections performed. A total of 80 trials were performed at each of the 4 levels of volume (1.0, 2.0, 5.0, and 8.0mL), as described in **Section 4.2.1.1**.

Results

Descriptive statistics for measured volume of all injection trials are shown in **Table 12**. The mean absolute difference provides a measure of error of the software calculation volume. The overall mean absolute difference across all trials was 0.29 ± 0.33mL.

Table 12. Descriptive statistics for difference between calculated and actual volume injected

Goal Volume (mL)	Mean Absolute Difference ± SD (|Calculated – Actual|) (mL)	[Min, Max] (mL)
1.0	0.07 ± 0.05	[-0.21, 0.10]
2.0	0.08 ± 0.06	[-0.25, 0.29]
5.0	0.21 ± 0.13	[-0.33, 0.53]
8.0	0.82 ± 0.18	[0.26, 1.64]
OVERALL	0.29 ± 0.33	[-0.33, 1.64]

The assumptions for linear regression were verified in SPSS and a linear model was determined between calculated and actual volume. The model significantly predicts calculated volume ($p<0.001$) with a large coefficient of determination, R^2=99.6%. The equation of the model is:

$$(Calculated\ Volume, mL) = 1.13 * (Actual\ Volume, mL) - 0.26$$

A graph of the calculated versus actual volume and linear regression model was plotted in SPSS and is shown in **Figure 61**.

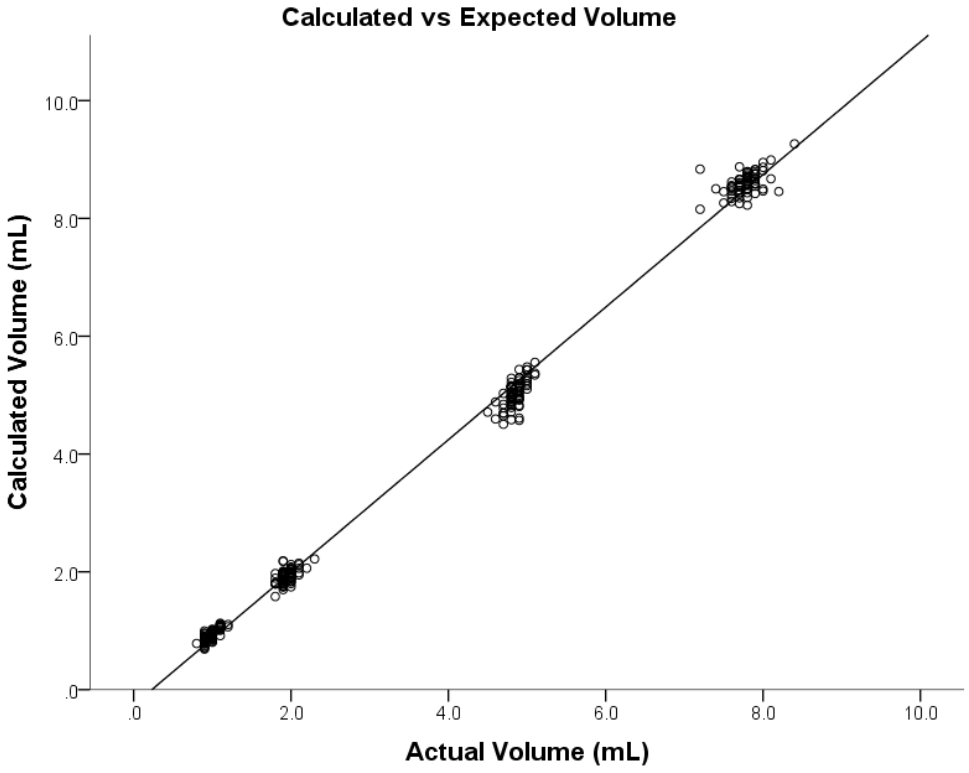


Figure 61. Linear relationship between the calculated and actual volume injected

Discussion

The results from the Accuracy of Measured Volumes experiment show that the current method for calculating volume by multiplying delta time by delta flow rate results in roughly a 10% error from the actual injected volume. The data from this experiment yielded a linear regression model with slope of 1.13 and intercept of -0.26, but with an R^2=99.6% which indicate a strong fit of the linear model to the data. Due to this large coefficient of determination, this linear model can be applied as a correction factor to improve the calculated volumes within the software. Further exploration should be performed to understand if the observed 10% error is linear with larger injection volumes, specifically during infusions of 0.5 to 1.0 L. These larger volumes would be like a situation where a clinician is administering fluids over time via an IV bag or using an infusion pump.

4.2.2 Additional Medication Injection Experiments

As identified earlier in **Section 4.2.1.2**, there were additional permutations of medication injections to fully assess system accuracy at detecting administered medications. In addition to the previous testing where injections of medications were always followed by saline flushes, the system should accurately detect:

1. repeated injections of the same medication, including repeated saline flushes, and
2. injections of different medications without a saline flush between injections.

This section describes the methods and reports the results of the additional permutations of medication injections.

4.2.2.1 Methods

Injections were performed similar to the methods described in **Section 4.2.1.1**. The main difference during these experiments was that injection ordering was not constrained to always have a saline flush performed after injection of every medication. In each of the following sections, the sequencing for the injections will be defined for that section.

For these experiments, larger quantities of medications needed to be created to replenish stock. After replenishing the supply of medications, detection windows were calculated as done previously. Thresholds were determined by finding the resting conductivity values after injecting each simulated medication, then expanding the detection window symmetrically above and below the resting value, ensuring that adjacent detection windows did not overlap while leaving a buffer between detection windows. The resulting detection windows are shown in **Table 13**. Notable changes were to the detection windows for esmolol and succinylcholine; their resting thresholds were lower than previously determined. This suggests that previous testing had slightly higher salt concentrations for esmolol and succinylcholine.

Table 13. Detection Windows for additional medication injection experiments

Medication ID	Resting Threshold (Arduino units)	Offset (Arduino units)	Detection Window [Min, Max] (Arduino units)
Saline	20	0	[0, 40]
Propofol	112	35	[77, 147]
Epinephrine	250	50	[200, 300]
Esmolol	400	50	[350, 450]
Succinylcholine	530	50	[480, 580]

4.2.2.2 Proper Identification of Repeated Injections

In this section of experiments, the sequencing was chosen to assess the accuracy of the system at properly identifying repeated injections of the same medication, including repeated saline flushes, over 10 repetitions of each medication (saline, propofol, epinephrine, esmolol, and succinylcholine) and over the same injection volumes as before (1.0, 2.0, 5.0, 8.0mL). At the start of each medication's set of repetitions, that medication was injected to ensure that a baseline was set for recording injection repetitions (effectively, there were 11 repetitions of the same medication and data was only analyzed for the last 10 injections—the injections that directly followed the same medication). For each of the 5 medications (including saline), 10 repetitions of injections were performed at each of the 4 levels of volume, resulting in a total of 40 injections for each of the 5 medications. Thus, there were a total of 200 trials of injections performed.

Results

The results from the repeated injection trials are summarized in **Table 14**. All repeated injections were properly identified by the system for all medications across all volumes.

Table 14. Frequency of properly detecting the repeated injection out of 40 injections for each medication across all volumes

Medication Being Repeated		**Volume of Medication to Administer (mL)**			
Salt Concentration (g/L)	**Medication ID**	**1.0**	**2.0**	**5.0**	**8.0**
0.0	Saline	10	10	10	10
0.3	Propofol	10	10	10	10
1.0	Epinephrine	10	10	10	10
2.0	Esmolol	10	10	10	10
3.0	Succinylcholine	10	10	10	10

A graph of the detection windows with respect to the trial data for all volumes are shown in **Figure 62**. The whiskers on the plots show the extrema while the black horizontal lines show the mean values.

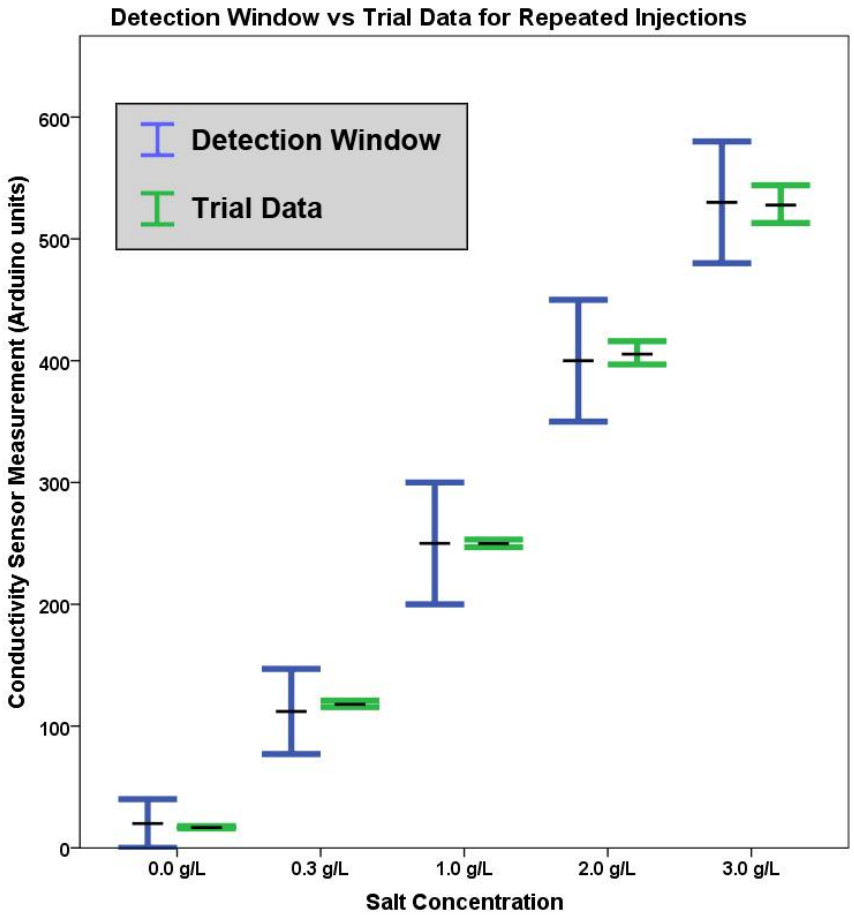


Figure 62. Detection windows vs trial data for repeated medication injections.

Discussion

The results from the repeated injection experiments suggest that the system can properly detect repeated injections of the same medication for at least 10 repetitions of that medication. Due to the data being tightly spread around the means and not skewed with respect to the detection windows,

the data suggests that repeated injections would always be properly detected by the system for these medications. The results also support that for medications with increasing concentrations of salt, there is more variability in the measured conductivity of the medication during injection.

4.2.2.3 Proper Identification of Paired Injections Without Saline Flush Between Injections

In this section of experiments, the sequencing was chosen to assess the accuracy of the system at properly identifying injections following a different medication, not including saline flushes (as this was assessed previously in **Section 4.2.1.2**), over 10 repetitions of each medication combination of the 4 medications (propofol, epinephrine, esmolol, and succinylcholine) and over the same injection volumes as before (1.0, 2.0, 5.0, 8.0mL). For example, 10 repetitions were performed for 2.0mL propofol injection following 2.0mL esmolol injection, and visa-versa, for all combinations of medications at all injection volumes. Similar to previous experiments, injection volumes were not permuted during this experiment; i.e. injection volumes were blocked so that all 2.0mL volumes were performed together, and 2.0mL injection volumes were not performed with 5.0mL injection volumes. To make this point clearer, 2.0mL propofol was never injected after 5.0mL esmolol—2.0mL injections of medication_A were always paired with 2.0mL injections of medication_B. The full set of all 12 injection pairings is shown in **Table 15**.

Table 15. Injection pairings for different medication injections without saline flush between injections

ID	Lower Salt Concentration followed by Higher Salt Concentration	ID	Higher Salt Concentration followed by Lower Salt Concentration
A	Propofol followed by Epinephrine	G	Epinephrine followed by Propofol
B	Propofol followed by Esmolol	H	Esmolol followed by Propofol
C	Propofol followed by Succinylcholine	I	Succinylcholine followed by Propofol
D	Epinephrine followed by Esmolol	J	Esmolol followed by Epinephrine
E	Epinephrine followed by Succinylcholine	K	Succinylcholine followed by Epinephrine
F	Esmolol followed by Succinylcholine	L	Succinylcholine followed by Esmolol

*IDs are given to each injection pairing to condense future tables and for future in-text referencing.

For each of the 12 injection pairings, 10 repetitions of injections were performed at each of the 4 levels of volume, resulting in a total of 240 injections for each of the 4 individual medications. Thus, there were a total of 960 trials of injections performed.

Results

The results from the paired injection trials are summarized in **Table 16**. All trials successfully identified the paired injections properly except for paired injections of 1.0mL for injection IDs (I) and (K), which corresponded to propofol (0.3 g/L) and epinephrine (1.0 g/L) being injected after succinylcholine (3.0 g/L), respectively. Out of 960 injections, only 11 were improperly identified by the system—the system was over 98% accurate over all injections during this experiment.

Table 16. Frequency of properly detecting the injection of paired injections for each of the 12 medication pairings across all volumes

Paired Injection			Volume of Medication to Administer (mL)			
ID	**Previous Salt Concentration (g/L)**	**Salt Concentration to Detect (g/L)**	**1.0**	**2.0**	**5.0**	**8.0**
A	0.3	1.0	10	10	10	10
B	0.3	2.0	10	10	10	10
C	0.3	3.0	10	10	10	10
D	1.0	2.0	10	10	10	10
E	1.0	3.0	10	10	10	10
F	2.0	3.0	10	10	10	10
G	1.0	0.3	10	10	10	10
H	2.0	0.3	10	10	10	10
I	3.0	0.3	2	10	10	10
J	2.0	1.0	10	10	10	10
K	3.0	1.0	7	10	10	10
L	3.0	2.0	10	10	10	10

Discussion

The results from the paired injection experiments are not surprising. In the previous experiments in **Section 4.2.1.2** where medications were always injected after a saline flush, the medications were properly identified in 160 out of 160 injections. In those cases, a medication with a higher concentration of salt solution was injected after a medication with a lower concentration of salt solution. Similarly, in this experiment for cases (A) through (F), a medication with a higher concentration of salt solution was injected after a medication with a lower concentration of salt solution. In these cases, the medications were properly identified in 240 out of 240 injections.

In the previous experiments in **Section 4.2.1.3** where saline flushes were performed after each medication injection, saline flushes (0.0g/L) of 1.0mL were not properly identified by the system in 10 out of 10 injections when following injections of succinylcholine (3.0g/L). Additionally, from previous experiments, saline flushes of 1.0mL were not properly identified by

the system in 9 out of 10 injections when following injections of esmolol (2.0g/L). In this experiment, it was observed that propofol (0.3 g/L) injections of 1.0mL were not properly identified by the system in 8 out of 10 injections when following injections of succinylcholine, and epinephrine (1.0 g/L) injections were also not properly identified by the system in 3 out of 10 injections when following injections of succinylcholine. In all of these cases, the system did not properly identify the injected medication when the injected medication had a lower salt concentration by at least 2.0g/L than the previously injected medication. These findings suggest that the system's current detection method is flawed for low-volume injections when a medication with -2.0g/L difference or more is injected after a medication with a higher salt concentration (i.e. a 0.5g/L medication injected after a 3.0g/L medication).

From the results of all injection experiments performed in this dissertation, when a medication with a higher concentration of salt solution is injected after a medication with a lower concentration of salt solution, there is high probability that it will be properly identified, especially with larger injection volumes. Conversely, when a medication with a lower concentration of salt solution is injected after a medication with a higher concentration of salt solution, there is a low probability that it will be properly identified with smaller injection volumes, especially when the difference in salt concentrations between the medications is high. This behavior is not amenable for detecting user errors during training scenarios, as the user may inject any volume of the medications in any order.

One possible solution is to limit the medications to be mapped across a tighter range of salt concentrations, but that would require low variability in conductivity measurements across all combinations of medication injections in order to achieve high accuracy when identifying

medication injections. Future testing could explore the limits to the range of salt concentrations while maximizing the amount of medications that are possible to properly identify by the system.

Another possible solution, as mentioned during previous discussion, is to develop a mathematical model that instantaneously predicts what simulated medication is being injected based upon the initial injection rate, rate of change in conductivity, and knowing what was previously injected. Future work should explore development of such a model and perform similar experiments on accuracy to compare with the current detection methods.

5.0 USABILITY TESTING

In addition to benchtop testing that assesses the accuracy and reliability of the measurement systems within BodyExplorer, testing BodyExplorer with respective end users provides support that the design meets the needs of the users or provides constructive feedback to improve the design to meet those needs. This chapter presents the methods and results for multiple testing sessions with respective end users throughout BodyExplorer's design process.

5.1 INTRODUCTION AND RATIONALE

Usability testing is performed in order to verify that an engineered system is actually usable by the end-users that the system was designed for [106, 107]. Iterative testing identifies functions or features of the system that require improvements to be made and verifies that new improvements rectify previously identified problems while also not introducing new problems. Efficiency analyses in human factors literature have identified that frequent testing with small numbers (N=5) of participants is often sufficient for detecting 80% of usability problems [108, 109].

Usability testing sessions were performed at three different stages of development. A combination of formative and summative testing designs, as defined in [107], permitted early designs of BodyExplorer to be evaluated by representative end users. Formative testing is a way to receive early feedback on a design, or multiple designs, before engineering all the functionality of the design into the system. Formative testing allows multiple design ideas to be tested with minimal development costs while guiding future design based upon feedback from end users.

Summative testing is typically performed after formative testing and is performed after most, if not all, of the design functionality has been incorporated into the system. The structure of BodyExplorer's usability testing sessions was developed using best practices such as: piloting sessions before holding testing, controlling for moderator and observer effects, using "think aloud" to understand user actions, counterbalancing to assess different designs, and probing during debriefing [106, 107].

Addressing overall design goal (4) presented in **Section 3.2** that the system should be easy to use, the primary objective of the usability testing sessions was to evaluate BodyExplorer's ease-of-use, specifically focused on user interactions with the input device—the device that end users use to interact with the images that are projected onto the mannequin. The secondary objective of the sessions was to gather constructive feedback from the end users on their opinions of the positive features, areas for improvement, and applicability to healthcare education based upon interacting with BodyExplorer during the sessions.

The following sections describe the three usability testing sessions. In each section, BodyExplorer's state of development is initially described to orient the reader to the functions and featured being used by study participants. Then, the methods and qualitative results based upon the secondary objective are presented. A brief discussion is presented based upon the results and to provide rationale for the subsequent usability testing session. After all of the individual sections for usability testing sessions, the final section provides quantitative analysis of the primary objective evaluating the usability of BodyExplorer through user interactions with the input device across all usability testing sessions. Finally, the qualitative and quantitative results from all usability testing sessions are discussed.

5.2 USABILITY TESTING SESSION 1: WINDOWING AND IV DRUG ADMINISTRATION, RAPID SEQUENCE INDUCTION

Prior to this round of usability testing, the prototypes defined in **Section 3.3.1** were at an initial stage of system integration. The data acquisition hardware and sensor systems for measuring administered medications, application of cricoid pressure, and depth of endotracheal intubation were installed inside the mannequin. The data acquisition hardware utilized the USB-6009, and the software was compiled using LabVIEW. The software was compiled into two applications; one for windowing and IV medication administration, and the other for rapid sequence induction (RSI), including medication administration, application of cricoid pressure, and endotracheal intubation (as defined in **Section 0**). An initial auxiliary display was incorporated using an iPad to provide feedback about cricoid pressure application. **Figure 63** shows the iPad auxiliary display in response to application of cricoid pressure. A colored indicator illuminates when the user touches the cricoid or thyroid cartilage, and a gauge indicator shows measured force on the cricoid.

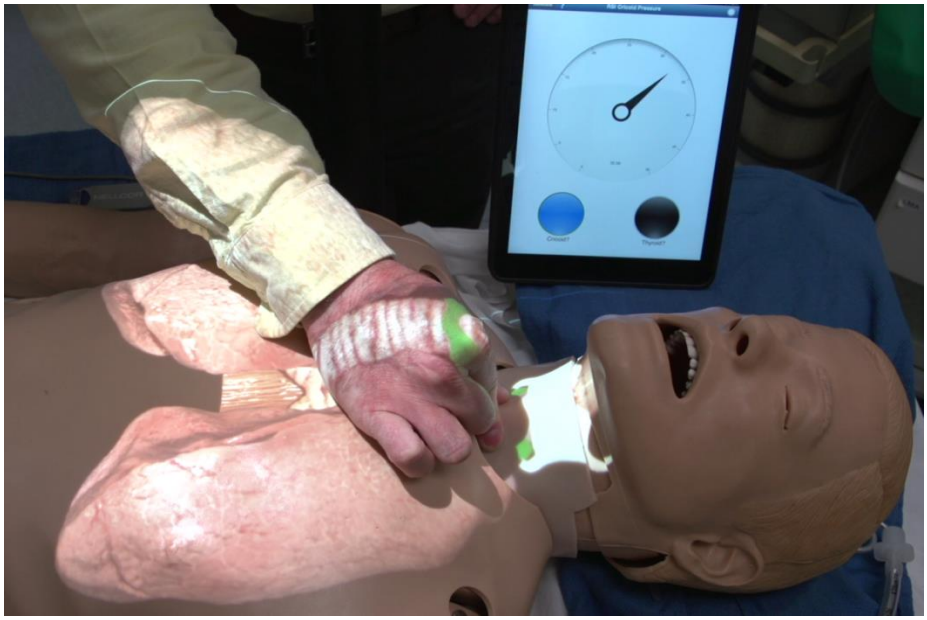


Figure 63. Cricoid pressure feedback on iPad
(© 2015, Douglas A. Nelson Jr.)

The anatomy being displayed by the software was comprised of 2D images, with animations for beating heart and breathing lungs comprised of cycling 2D images (this was prior to the integration of Zygote 3D anatomy models).

Users could manipulate the images of anatomy projected onto the mannequin's surface by using Input Device 1 shown in **Section 3.3.4.5**. However, Input Device 1 did not have functionality on the device for allowing the users to select an active anatomy layer for manipulating. Keyboard shortcuts needed to be pressed on the adjacent computer in order to toggle between active anatomy layers ("1" for manipulating the muscle layer, "2" for manipulating the bone layer, and "3" for manipulating the organ layer). Furthermore, switching between software applications for windowing and IV medication administration and the application for RSI required a technician to perform the switch. Keyboard shortcuts for selecting anatomy layers and requiring a technician to switch applications did not support the system design goals established for BodyExplorer

regarding self-usability. Thus, end user feedback was elicited during usability testing session 1 (hereafter, "UT1") to guide improvements to this initial version of the integrated system.

UT1 was aimed at exploring the baseline level of usability by end users. The purpose of this formative usability test was to identify difficulties that end users experienced while interacting with BodyExplorer, while also gathering feedback from the end users on additional functions and features important in future design iterations.

5.2.1 Methods

5.2.1.1 Piloting the Protocol

Prior to study enrollment, the protocol for UT1 was piloted internally by the usability testing team members, as recommended from best practices [106, 107]. The usability testing team was comprised of the PI (the author, hereafter referred to as "session moderator") and research assistants (session observers), including graduate and undergraduate students from the University of Pittsburgh School of Nursing and School of Arts & Sciences. The primary purpose of the pilot was twofold:

1. assess the protocol efficiency and identify additional resources that might be necessary to perform the sessions, and
2. train the usability testing team on procedures for data collection during usability testing sessions.

Additionally, by having team members follow the protocol as an end user, team members could develop a level of empathy for what the end users would be doing during the sessions.

The research assistants participated in pairs during the pilot; one assistant acted as the session participant, while the other assistant acted as an observer and collected data throughout the

session. The moderator guided the participants through the pilot sessions. Once the protocol was completed, the two assistants switched roles to gain the other perspective.

After running through the pilot with all usability testing team members, the protocol itself was not modified. However, key findings improved efficiencies for data collection and overall flow of the session. Positioning of the cameras for the video recordings needed to be checked before testing sessions to ensure they were properly pointed at the user, and stage instructions were added to the moderator's script to "Start Video Recording" at the appropriate place in the protocol. Spacing for notes by observers on the data collection tool was increased by including only one end user subtask per page. The data collection tool was also modified from two-sided to one-sided to reduce awkward page flipping by the observers during data collection. Additionally, the pilot showed that having a prop available was necessary for the think aloud demonstration of tying one's shoe; the assumption that participants would be wearing laced shoes was not guaranteed to be true.

5.2.1.2 Recruitment and Enrollment

After IRB approval, and completing the internal pilot, UT1 commenced. Representative end-users were recruited using a convenience sample from the University of Pittsburgh School of Nursing. Instructors, graduate students, and undergraduate students were recruited through mass email on School of Nursing distribution lists and by word of mouth from Usability Testing Team members. Study participants enrolled using the online scheduling software, SignUpGenius[110].

5.2.1.3 Data Collection

Each hour-long testing session was run by a session moderator and included only one study participant. At least two trained session observers from the Usability Testing Team were present to take observational notes using a data collection tool (**Appendix C**). The notes were based upon

observing the participant's actions and comments while using BodyExplorer. The testing sessions were recorded using four different video feeds (3 VGA at 704x480, 1 AVCHD at 1920x1080; all at 29.97 fps) from various angles to allow for post-session review and quantified analysis of usability, as will be detailed in **Section 5.5**. **Figure 64** shows the locations of the cameras with respect to BodyExplorer in the testing environment. A demographic tool (**Appendix C**) was used to collect information about the participant's background in addition to the participant's previous educational and simulation experiences.

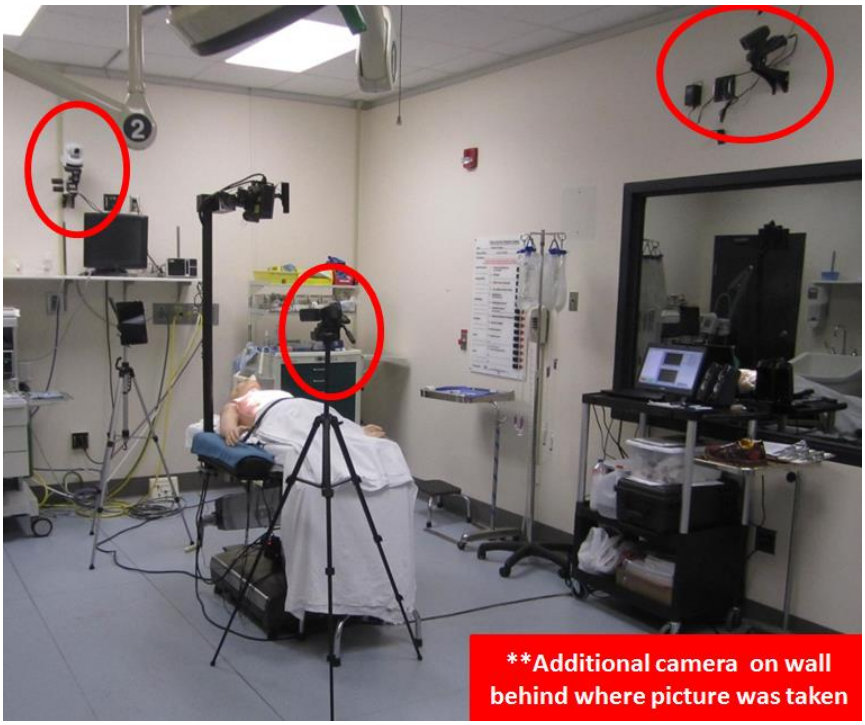


Figure 64. Setup of testing session at the University of Pittsburgh School of Nursing
(© 2015, Douglas A. Nelson Jr.)

5.2.1.4 Session Protocol

The overall structure for each hour-long testing session is outlined in **Table 17**.

Table 17. Testing session outline

Time Sequence (minutes)	Duration (minutes)	Description
00:00 – 00:03	3	Introductions, Informed Consent, Demographic Tool
00:03 – 00:06	3	Practice Think-Aloud, Introduction to BodyExplorer
00:06 – 00:35	29	System Testing – Task List
00:35 – 00:45	10	Debrief Questions, GAS Survey
00:45 – 00:60	15	Participant Compensation, Buffer Time, Reset System

Each testing session began with introducing the participant to the session observers and moderator, administering the informed consent script (**Appendix C**), and completion of the demographic tool. Participants were introduced to the think aloud method [107] and asked to demonstrate the method using a simple task of tying one's shoe. The participant was then given a task list for completing two interactive modules: Task 1: Windowing and IV Drug Administration, and Task 2: Rapid Sequence Induction (RSI). A summary of the tasks is provided in **Table 18**. The full task list with instructions for the participant can be found in **Appendix C**.

Table 18. Summary of tasks and subtasks performed during usability testing sessions

TASK	SUBTASK
Task 1: Windowing and IV Drug Administration	A. Open, move, and resize anatomy windows
	B. Play heart sounds and show ECG
	C. Administer simulated medications
Task 2: Rapid Sequence Induction (RSI)	A. Administer simulated medications
	B. Apply cricoid pressure
	C. Perform endotracheal intubation

The task list for each module instructed the participant to interact with the projected anatomy and perform medical procedures using BodyExplorer while thinking-aloud to describe their thoughts, actions, and impressions. When participants verbalized the subtask for opening,

moving, or resizing anatomy windows, the moderator pressed the appropriate keyboard shortcut to switch between active layers, simulating automatic window activation behavior for the user to open, move or resize that layer. After completing each task, the participant was asked structured debriefing questions from the moderator according to best practices [107]. The debriefing questions were also modeled according to an established simulation debriefing protocol for gathering, analyzing, and summarizing feedback (the GAS tool of the Structured and Supported Debriefing Model, [111]). The debriefing questions can be found in **Appendix C**.

5.2.2 Results

5.2.2.1 Demographics

There were a total of 15 participants, and the demographics are shown in **Table 19**. Participants had prior experience—either in simulation or in human patients—administering medications (N=11), performing endotracheal intubation (N=9), and applying cricoid pressure (N=10). There were 3 participants who had previously used initial prototypes of BodyExplorer outside of UT1.

Table 19. Participant demographics for UT1

User Group	N
Instructor	5
Graduate Student	7
Undergrad Student	3
TOTAL:	15

Prior Experience…	N
Administering Medication	11
Endotracheal Intubation	9
Applying Cricoid Pressure	10
Using BodyExplorer	3

Figure 65 shows a participant during usability testing with the moderator and three observers in the background. Observers recorded the comments and feedback from the participants.

Figure 65. Usability testing session during UT1 (© 2015, Douglas A. Nelson Jr.)

5.2.2.2 Feedback from Participants

Participants gave positive feedback about their ability to interact with the anatomy and physiology on the mannequin, specifically related to peeling back layers of the anatomy to see underlying structures—*"Wow, no way! That's wild. That's surprising, and almost shocking, to just have that happen the first time that opens up and watching it drag around and reveal."* Participants were excited about BodyExplorer's automatic reactions to their interactions with the mannequin, especially when the physiology automatically updated in response to administration of simulated medications—*"Wow, just double-checking, is there nobody behind the controls, this is all automated? That's pretty amazing!"* In comparison to other software applications for anatomy and physiology, participants were positive about BodyExplorer's integration of anatomy and physiology into one application—*"A lot of programs that are out there now, you can solely explore*

the anatomy, or you can go and do classic physiology experiments online, but to be able to see it right here as you are watching the heart beat is pretty cool."

Participants exhibited some difficulties and frustrations with the interface. The main difficulty with the interface was using the input device to manipulate anatomy windows. All participants commented on difficulty with the input device during testing sessions. Primary reasons for the difficulties derived from poor tracking of the input device (corresponding to poor control of the cursor on the mannequin) and difficulty using the button on the input device (either pressing the incorrect button or inadequate feedback that the button was pressed).. Participants commented that the input device was straightforward to use, it just didn't appear to behave properly all of the time—*"It seems pretty straightforward, but it takes a little bit of getting used to…The window resizing. It is intuitive, but it is cumbersome in the way that it is implemented…I just had to get time to get used to it."* Quantitative analysis on input device usability will be presented in **Section 5.5**.

Participants also suggested adding more content to the scenarios. While participants were positive about having the ability to peel back layers of anatomy, they wanted to be able to see more organs and internal structures, specifically cutaway views inside organs such as the heart, lungs, and gastrointestinal (GI) system—*"It would be really easy to visualize how the blood is flowing in the heart, and it's usually counterintuitive to how you learn it from textbooks because of the way that the anatomy is."* Multiple participants wanted the ability to see other patient vital signs, in addition to the ECG, such as oxygen saturation, blood pressures, and lung gas exchange—*"Being a clinician, it would be good to have the vital signs."* In addition to the normal anatomy and physiology displayed in the present version of the BodyExplorer, participants wanted to be able to

explore pathological conditions by visualizing anatomical differences, seeing variations of patient vital signs, and hearing the associated differences in auscultation during a patient assessment.

With respect to RSI in Task 2, participants appreciated being shown on the auxiliary display how hard they were pressing during application of cricoid pressure. A graduate student commented on not knowing how hard to press—*"We are told to apply 30 to 40 Newtons, but I never really knew what [applying] 30 to 40 Newtons felt like."* Participants also suggested adding the recommended pressure range to the display of applied pressure so that they could be reminded to hold pressure in the correct range.

5.2.3 Discussion

The observed difficulties and frustrations while participants used the input device were not unsurmountable problems to address during the next design iteration. The two primary reasons postulated for difficulty while using the input device derived from poor tracking of the device, corresponding to cursor positioning on the mannequin's surface, and missed button presses on the device—either the user pressed the wrong button or the system did not properly receive the "button press" signal from the device. When the method for tracking the input device cannot locate the device, the cursor position defaults to a screen position of (0,0), or the mannequin's right lower quadrant when projected onto the mannequin. Modifying the software to default to the last known cursor position could fix some of the glitches observed with the interface. However, this modification does not ultimately fix the root cause of the issue—the camera loses sight of the input device's IR-LED. Modifications to the input device design needed to be performed during the next iteration of testing to reduce the possibility of the camera losing sight of the input device's IR-LED.

The observation that users had difficulty with the button in the input device suggests that the button should be larger and more distinct from other buttons on the device. The active button was adjacent to an inactive button on the input device, and thus, the user would sometimes confuse the two. Furthermore, the tactile response of the button was subtle and did not provide enough tactile feedback to the user when the button was pressed. Future designs of the input device need to ensure that the button is distinct from others and provides enough tactile feedback when pressed.

Future content development should support additional visualizations of anatomy and physiology. Both normal and pathological conditions should be able to be visualized and compared. In addition to heart rate and ECG, other patient vitals should be incorporated and synchronized with the visuals of anatomy and physiology.

5.3 USABILITY TESTING SESSION 2: INPUT DEVICE MODIFICATIONS AND MULTIPLE USERS

The purpose of this second usability test (hereafter, "UT2") was to verify that the improvements made to the input device in response to the first usability test fixed the observed issues with the input device. UT2 also incorporated group sessions where multiple end users interacted with BodyExplorer at the same time. During individual and group sessions, user difficulties while using the system were documented in addition to the functions and features that the end users would like to see in future design improvements. Throughout all usability testing sessions, usability of the input device was quantified in order to understand if improvements to the design improved usability between sessions. The quantitative data will be presented in the final section of this chapter.

5.3.1 System Modifications for Usability Testing Session 2

The primary modification to the interface was a new design for the input device. During UT1, participants had difficulty making the cursor follow the device's tip (an issue with device tracking), and they also had difficulty with the input device's button (button presses were missed). The new input device (Device 2) had the IR-LED mounted at an angle offset from the body axis of Device 2; whereas the previous design (Device 1) had its IR-LED mounted directly in-line with its body axis. Device 2 also had a larger tactile button compared to Device 1's button. **Figure 66** shows these modifications to the Input Device. Device 2 also had a scroll-wheel that was added to allow participants to control the active anatomy layer during window manipulations. Users only used Device 2 throughout UT2.

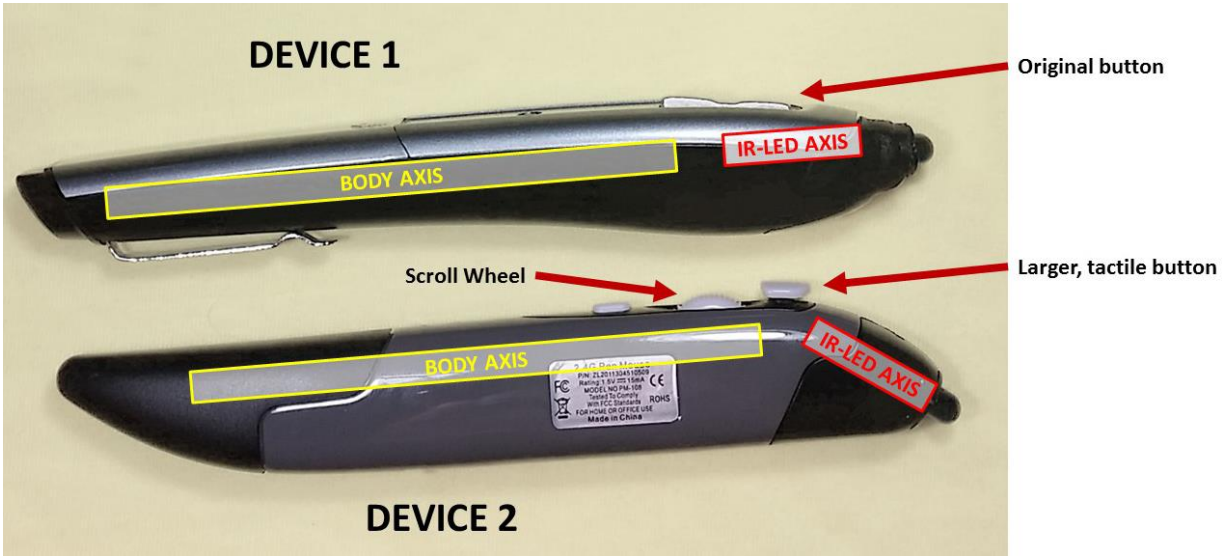


Figure 66. Design differences between input device 1 and 2
(© 2016, Douglas A. Nelson Jr.)

5.3.2 Methods

5.3.2.1 Recruitment and Enrollment

After IRB approval from both Robert Morris University and the University of Pittsburgh to add a testing site, UT2 commenced. Representative end-users were recruited from the Robert Morris University, School of Nursing and Health Sciences. Instructors, graduate students, and undergraduate students were recruited to participate in the study through mass email, by word of mouth, and by posted flyers on the Robert Morris University campus. As before, study participants enrolled using the online scheduling software, SignUpGenius [110]. Testing sessions were either individual or group sessions, with up to four study participants per group. Testing sessions were held at the Robert Morris University Regional Research and Innovations in Simulation Education (RISE) Center.

5.3.2.2 Data Collection

Similar to UT1, each hour-long testing session was run by a session moderator, and at least two session observers were present to take observational notes. The observers and moderator were the same as UT1, and sessions were recorded for post-session review and analysis. Quantitative analysis of the recordings was performed to assess usability with the input device and is presented in **Section 5.5**. A demographic tool (**Appendix C**) was used to collect information about the participant's background as well as the participant's previous educational and simulation experiences.

5.3.2.3 Session Protocol

The overall structure for each hour-long testing session was outlined previously in **Table 17**. The structure was the same for sessions with individual participants and groups. The tasks list was identical to those administered during UT1 (see **Table 18**).

During the group sessions, participants worked through the tasks list together. Each participant was given the opportunity to perform the tasks. The participants were permitted to talk with each other and learn from watching the other(s) interact with BodyExplorer.

5.3.3 Results and Discussion

5.3.3.1 Demographics

The participant demographics are shown in **Table 20**. Of the 12 total sessions 3 were group sessions. Within the 3 group sessions, one group had 2 undergraduate students, one group had 2 instructors, and one group had 1 instructor with 2 students. None of the participants previously used BodyExplorer.

Table 20. Participant demographics for UT2

User Group	Individual	Group	TOTAL
Instructor	3	3	6
Graduate Student	0	0	0
Undergrad Student	6	4	10
TOTAL	9	7	16

Prior Experience…	N
Administering Medication	15
Endotracheal Intubation	5
Applying Cricoid Pressure	4
Using BodyExplorer	0

Figure 67 shows two students and an instructor during in a group session performing the RSI task with an observer and moderator in the background.

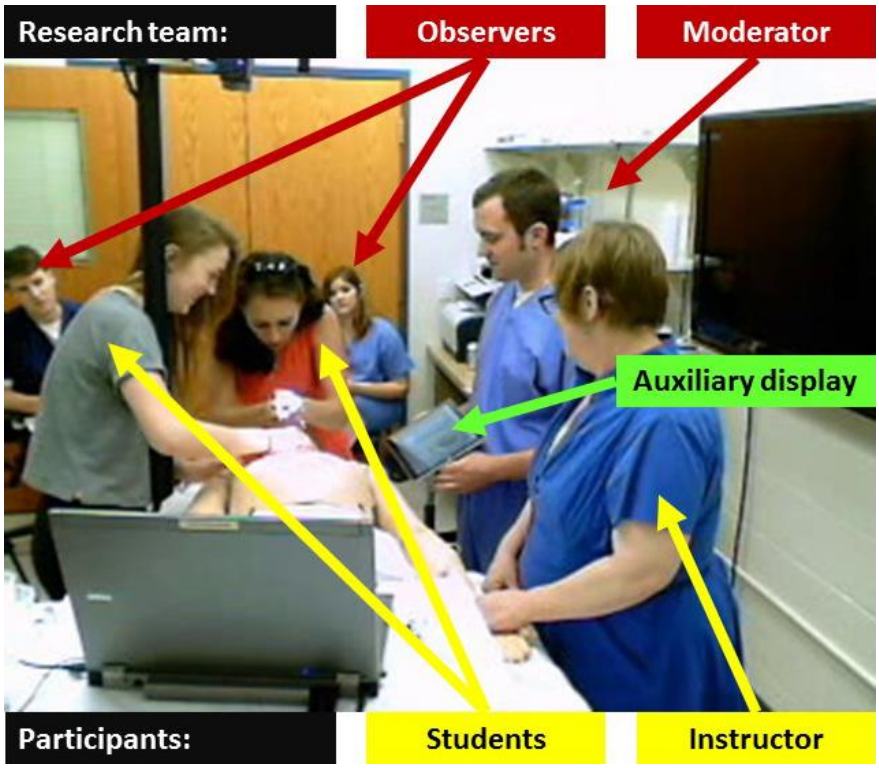


Figure 67. Group usability testing session at Robert Morris University
(© 2016, Douglas A. Nelson Jr.)

5.3.3.2 Feedback from Participants

Many of the comments from participants in UT2 were similar to comments from participants in UT1. Participants gave positive feedback about being able to interact with and see the internal anatomy on the mannequin. In addition to the normal anatomy and physiology, they desired the ability to see and hear different pathologies. When visualizing the organs, participants wanted the option to open the organs and see inside.

It was observed that participants had difficulties interacting with the anatomy windows using the input device. Quantification of the difficulties will be presented in **Section 5.5**. Participants had multiple suggestions for improvements related to interacting with the anatomy windows including:

1. intuitive window selection based upon what window the input device (or cursor) was hovering over on the body,
2. window labeling or highlighting based upon what window is currently active, and
3. changing the shape of the cursor based upon the active window or based upon what actions can be done with the cursor in that position (such as selecting, moving, or resizing).

In addition to intuitive window selection and labeling, participants suggested using a button, or other methods, to select the layers; scroll wheel selection was not easy to use. Participants also suggested enlarging the region in the center of the window that is used to move the window.

There were additional comments from participants in UT2 that differed from feedback received during UT1. After completing the RSI task, participants suggested:

1. showing the stomach rising and falling if esophageal intubation is mistakenly performed,
2. showing the anatomy from different views, such as rotating the mannequin onto its side or front in order to see a side or back view of the anatomy, and
3. using the medication administration task in an application to teach different titrations of medications and to show the expected physiological responses.

The UT2 sessions also incorporated group sessions where multiple participants interacted with BodyExplorer at the same time. During these group sessions, it was observed that participants would help one another when interacting with BodyExplorer. This observation supports the design decision to choose a projected AR system to foster interactions between users. This finding suggests that future analysis should look at comparisons between projected AR and HMD AR to see if interpersonal interactions differ between implementation methods. While the previous

Figure 67 shows one user reaching around the projector stand during application of cricoid pressure, participants did not comment on the stand being obtrusive while interacting with BodyExplorer in a group.

Additional development should ensure that the auxiliary display remains mobile. As shown in **Figure 67**, the auxiliary display was repositioned by the moderator from the table in the foreground to a position near the neck that allowed the user to see the cricoid pressure feedback adjacent to the neck.

5.4 USABILITY TESTING SESSION 3: NEW SOFTWARE AND PILOT TEST OF AUTOMATED INSTRUCTION

The purpose of this round of usability testing (hereafter, UT3) was to have users interact with the Unity version of BodyExplorer and evaluate new features afforded through Unity. The new features were guided by feedback received from previous user testing and aimed to guide developments towards self-use of BodyExplorer—addressing design goal (1) from **Section 3.2**. The primary features under evaluation were:

1. user login experience,
2. testing of two designs styles for the input device,
3. comparison of on-body versus on-screen menu types, and
4. an automated instruction tutorial method for using the input device.

Usability of the input device was quantified in order to understand if one design style was easier to use than the other, and to compare with data from previous usability testing sessions. The quantitative data will be presented in the final section of this chapter.

5.4.1 System Modifications for Usability Testing Session 3

5.4.1.1 New Software in Unity

The entire BodyExplorer software was re-architected using the Unity development environment based upon the design decisions outlined in **Section 0**. All the functions and features of the software from previous usability testing sessions were preserved.

Additional modifications were made to the software in response to the user feedback received from the previous usability testing sessions. In response to user feedback from UT2, affordances were built into the interface for manipulating anatomy windows on the mannequin with the input device. Colored borders were added to each anatomy window to show the user the boundaries of each anatomy window (**Figure 68**, **a**). Functional affordances found in standard Windows-Icons-Menus-Pointers were also added to the interface. Users could bring up menus on the mannequin for selecting different options or modes, such as selecting layers or showing labels (**Figure 68**, **b**). The cursor was modified to change its image based upon the type of hotspot that it was hovering over. Clicking and dragging the borders allowed the user to resize an anatomy window (Resize Hotspots, **Figure 68**, **c**). Furthermore, the entire area within the borders of each anatomy window was permitted to be clicked and dragged to move that anatomy window (Move Hotspot, **Figure 68**, **d**).

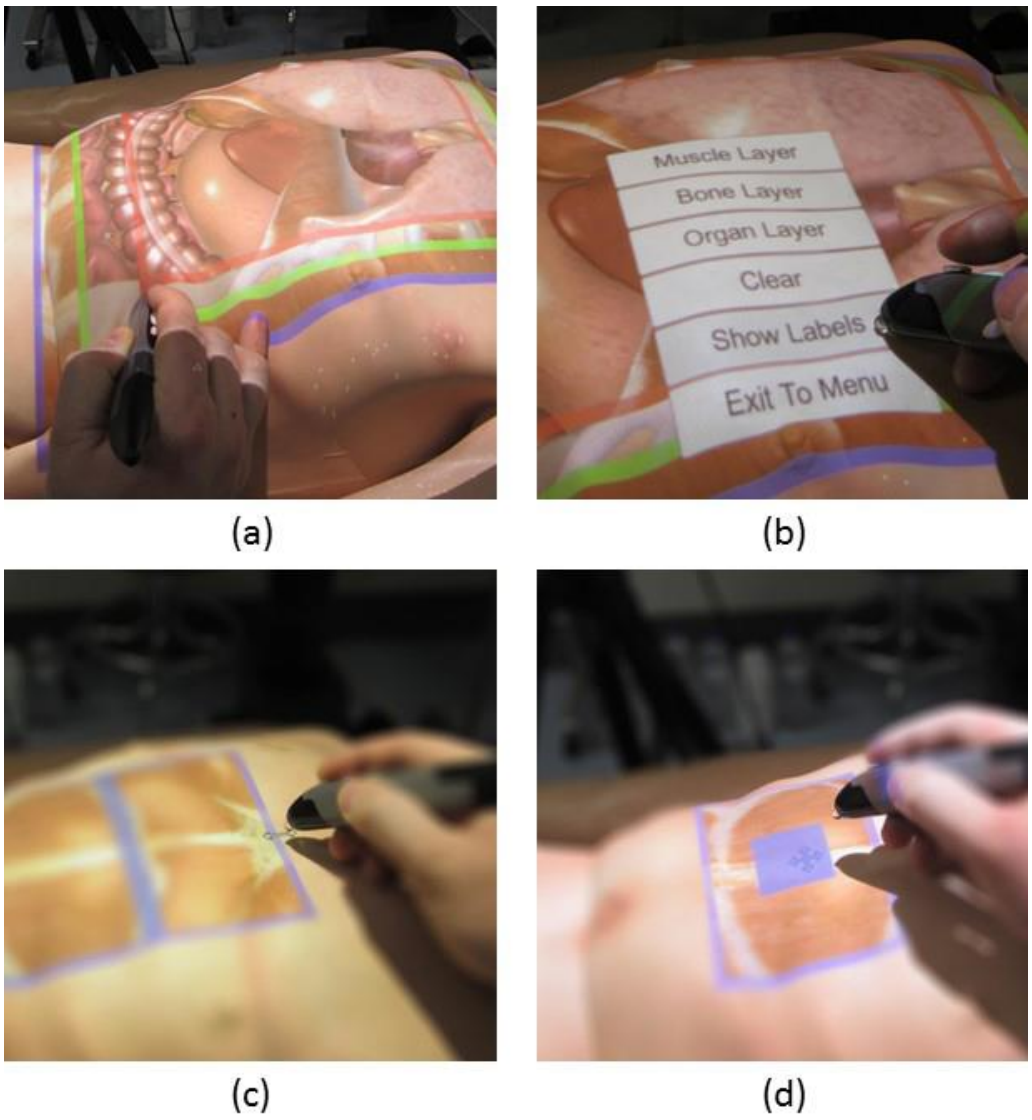


Figure 68. Affordances for manipulating anatomy windows on the mannequin: (a) borders on anatomy windows, (b) menus on the mannequin, (c) resize window hotspots, (d) move window hotspots.

The functionality of the auxiliary display was redesigned to be both a device for displaying information to the user, as well as a method to allow the user to control BodyExplorer. Previous implementations of the auxiliary display showed the user whether or not she/he was pressing on the thyroid or cricoid cartilage. The circular indicator on the right in **Figure 69** would light up red if the thyroid was pressed. The indicator on the left in **Figure 69** would change color if the cricoid

was pressed, with color mapped over a gradient from white to blue to green to red as increasing force was applied. Additionally, the numeric indicator on the left in **Figure 69** showed the user the quantified amount of force being applied in Newtons.

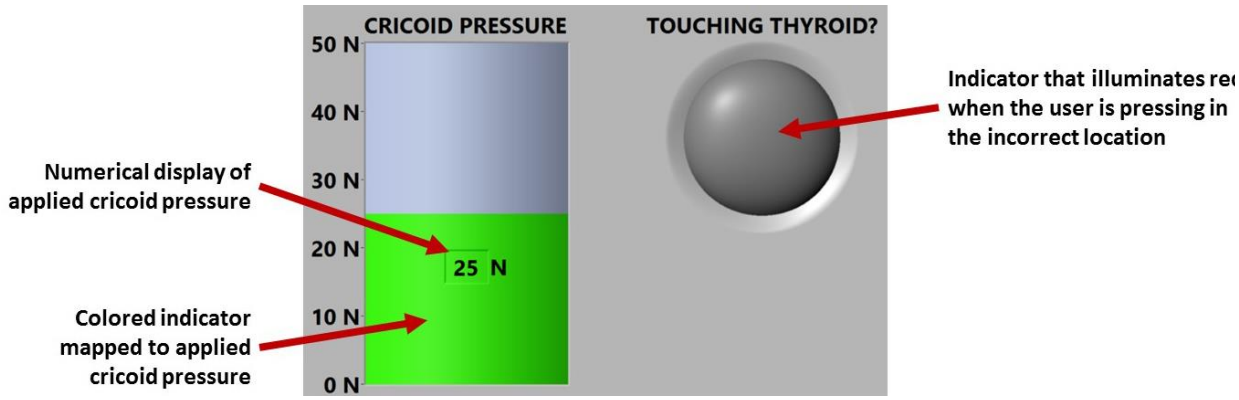


Figure 69. Auxiliary display for cricoid pressure application from UT2
(© 2016, Douglas A. Nelson Jr.)

The new redesign provided those same measurements on the auxiliary display and also linked them to animated models of the underlying anatomy. As shown in **Figure 70**, visual feedback was provided to the user via three mechanisms:

1. On-screen numerical display quantifying applied force,
2. On-screen anatomical color-mapped display quantifying applied force, and
3. On-screen animation correlating to applied force displaying the internal anatomical result of pinching the esophagus between the cricoid and spine.

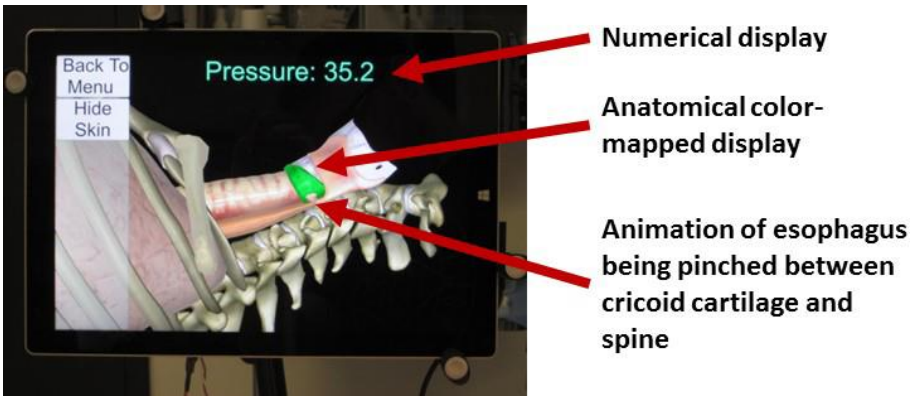


Figure 70. Auxiliary display for cricoid pressure application from UT3
(© 2016, Douglas A. Nelson Jr.)

A sequence showing a user touching the thyroid and then applying cricoid pressure is shown in **Figure 71**. The images on the right show the visual feedback mechanisms. The thyroid turns red when the user touches the thyroid cartilage on the mannequin (**Figure 71, b**). The cricoid turns blue when the user applies soft pressure to the cricoid cartilage (**Figure 71, c**). Then, it turns green when correct cricoid pressure is applied (**Figure 71, d**) and red when too much pressure is applied (**Figure 71, e**).

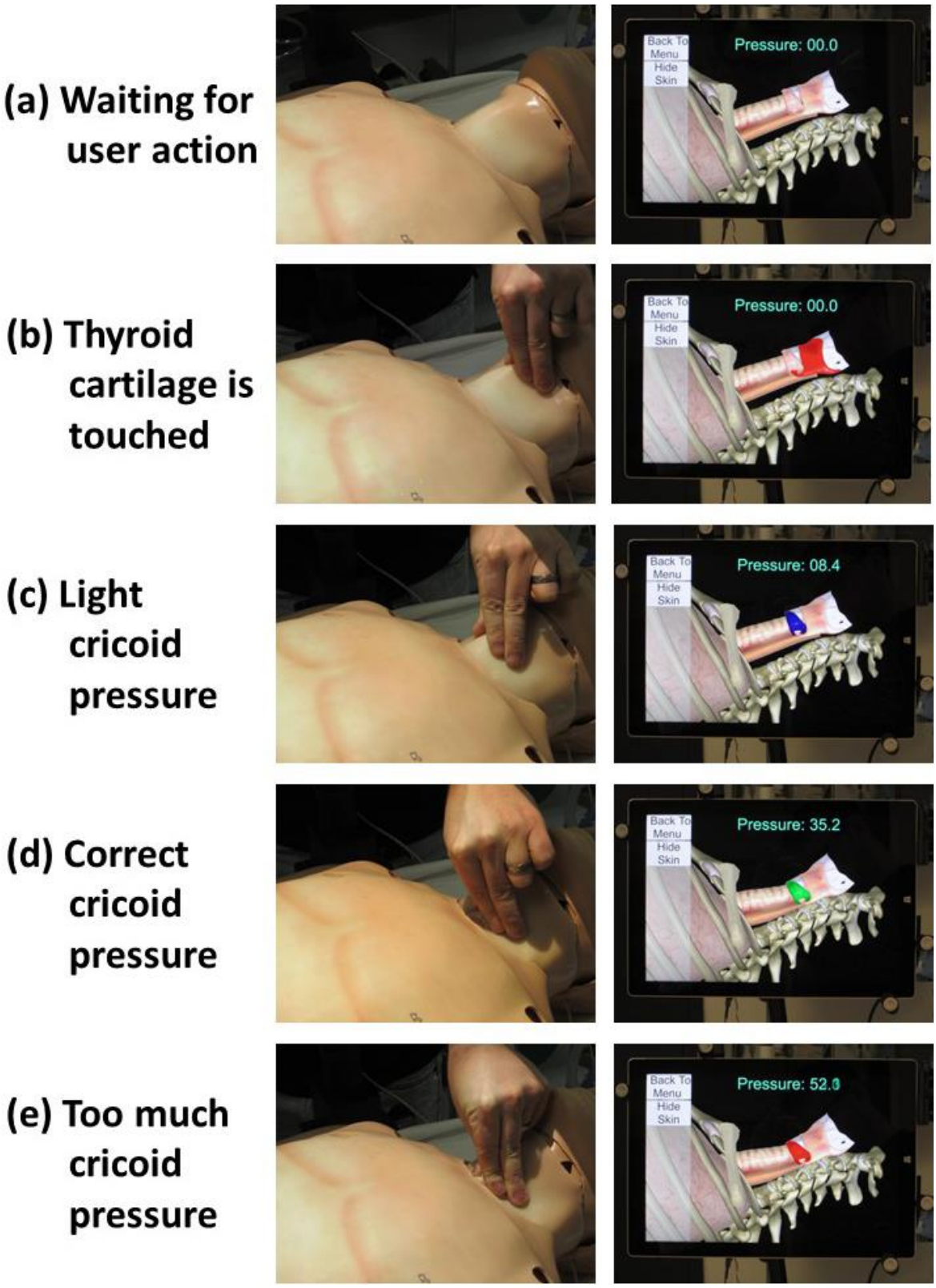


Figure 71. Auxiliary display showing feedback when applying cricoid pressure
(© 2016, Douglas A. Nelson Jr.)

The functional redesign of the auxiliary display also allowed the user to control BodyExplorer by interacting with the display's touchscreen. The user could select learning modules from a menu presented on the auxiliary display. Selecting a learning module, shown in the **Figure 72 (a) and (b)**, would launch the respective learning module on the auxiliary display as well as on the mannequin. When within a learning module, the auxiliary display provided menu options that allowed the user to make selections to modify the anatomy projections on the mannequin, options such as selecting an active anatomy layer to edit with the input device on the mannequin (**Figure 72, c**), or showing or hiding the skin (**Figure 72, d**).

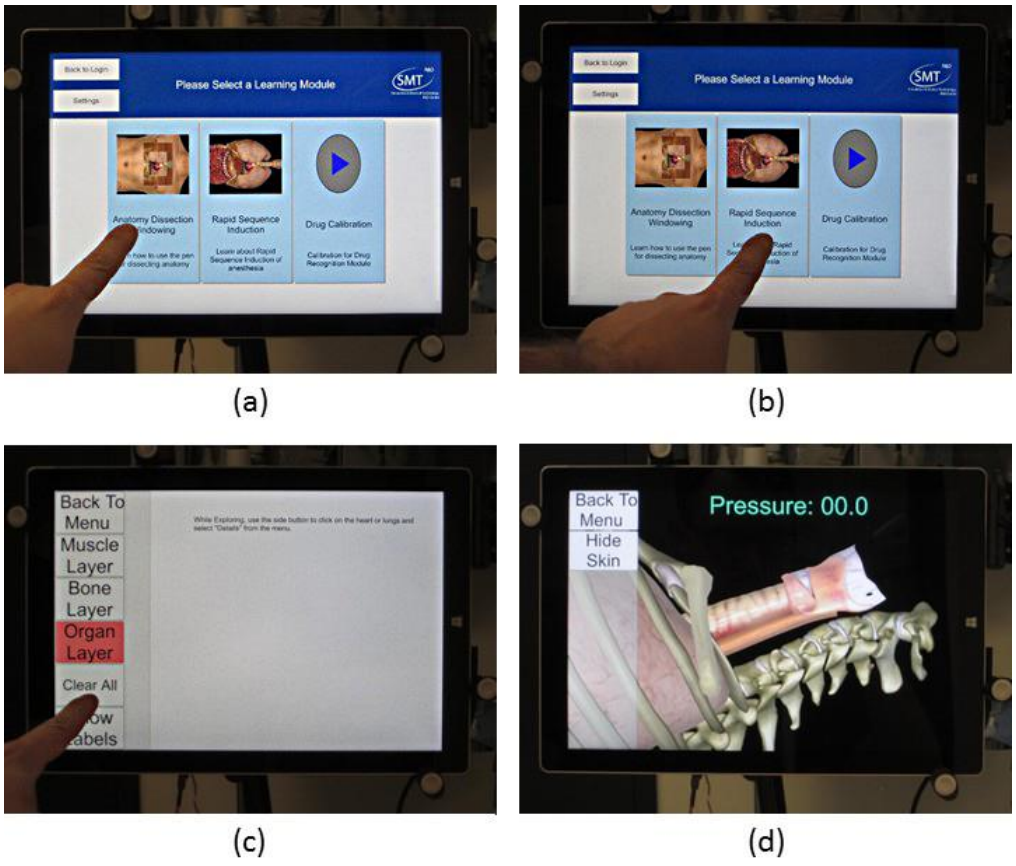


Figure 72. Menu functionality on the auxiliary display: (a) user selects the Anatomy Dissection & Windowing Module, (b) user selects Rapid Sequence Induction Module, (c) user can select on-screen menu options to modify anatomy windows, and (d) user can select to show/hide the skin.

5.4.1.2 User Login Experience

A user login experience was designed to provide the user the opportunity to walk up to BodyExplorer, have BodyExplorer auto-recognize the user's presence, be provided with login instructions, and select a learning module from the main menu. Detecting user presence was prototyped using a proximity sensor (Motion Trigger, i18, LittleBits Electronics, Inc.), and user login was prototyped using a photogate sensor (Photo Interrupter, GP1A57RJ00F, SparkFun Electronics, Inc.) to simulate the experience of using an ID to log in to BodyExplorer. The experience is shown through a storyboard approach in **Figure 73**. The user goes to the lab (**Figure 73, a**), enters the lab (**Figure 73, b**), and approaches BodyExplorer (**Figure 73, c**). BodyExplorer recognizes that the user is present and "wakes up", providing login instructions to the user on the auxiliary display (**Figure 73, d**). The user logs in by swiping their ID card (**Figure 73, e**), and then selects a learning module from the main menu (**Figure 73, f**).

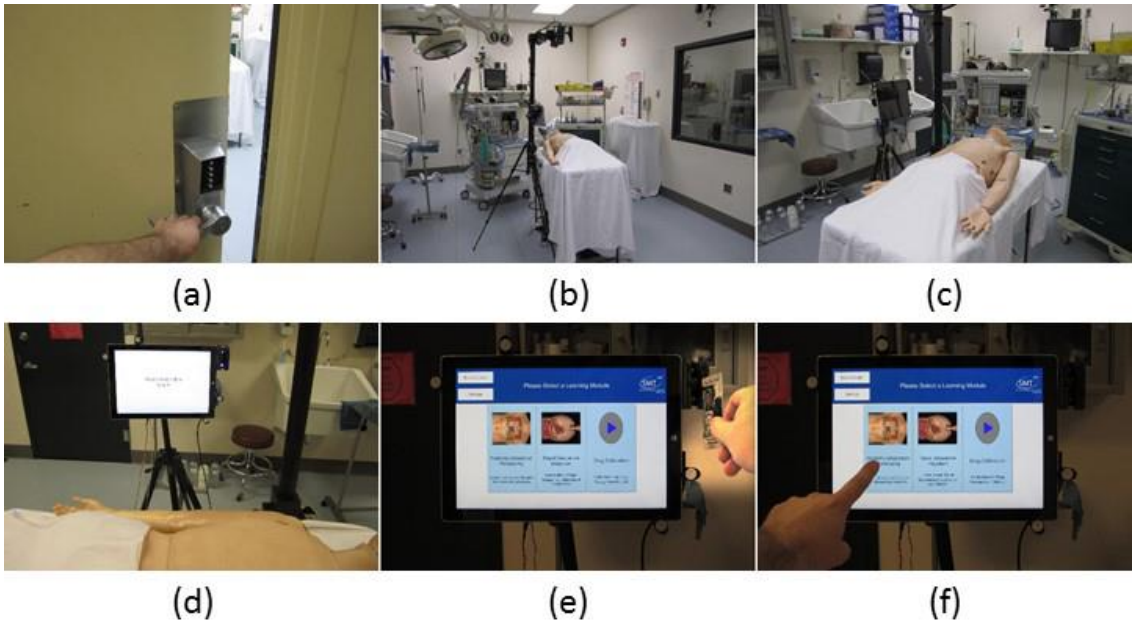


Figure 73. Storyboard showing user login experience: (a) user goes to the lab, (b) enters, (c) approaches BodyExplorer, (d) BodyExplorer detects user and wakes up, providing (d) login instructions on the auxiliary display. (e) The user logs in by swiping their ID card, (f) and then selects a learning module from the main menu. (© 2016, Douglas A. Nelson Jr.)

5.4.1.3 Input Device Tutorial

An automated instruction system was designed to guide the users through a tutorial for using the input device—the Input Device Tutorial. The automated instruction system spoke to the user through the speakers of the auxiliary display, which was designed to visually appear as if an instructor was present with the student. First, the Input Device Tutorial provided the user with a verbal description of the input device and how BodyExplorer "sees" the device. Then, the Input Device Tutorial provided instructions to the user for using the input device on the mannequin. A series of blue target boxes were displayed on the mannequin as shown in **Figure 74**.

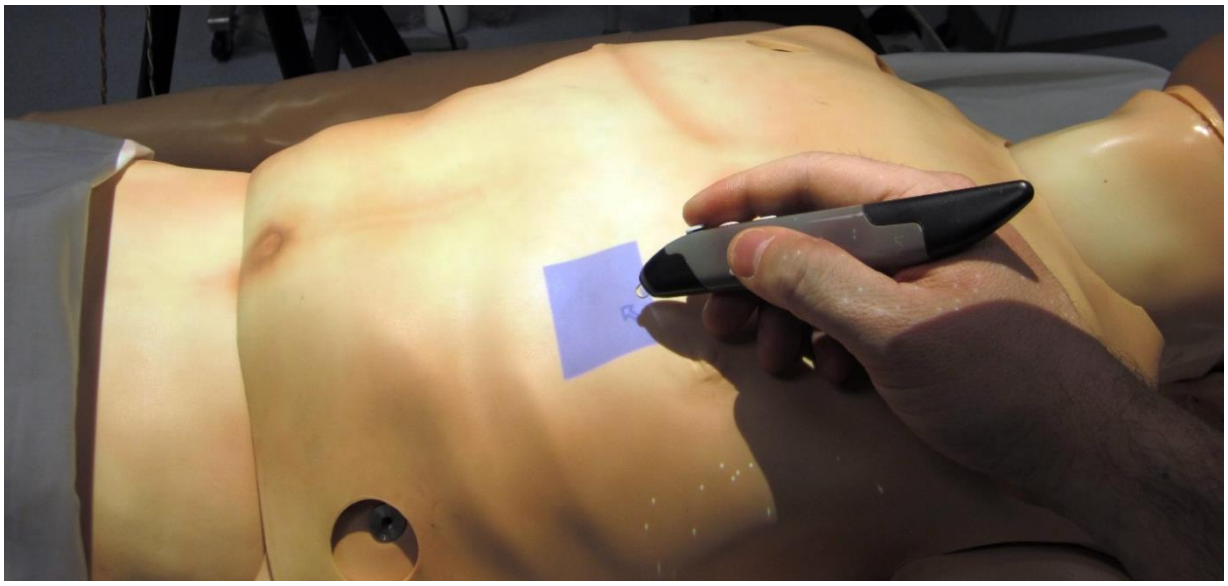


Figure 74. Blue target box displayed on the body surface during the Input Device Tutorial
(© 2016, Douglas A. Nelson Jr.)

The user was instructed to:

1. Point to the blue target boxes using the input device.
2. Point to and click on the blue target boxes using the input device.
3. Point to, click, and drag the blue target boxes to new blue target boxes using the input device. (This method opened an anatomy window on the mannequin, **Figure 75**)
4. Point to the side border of an anatomy window displaying a blue target box, click on the target box, and drag the box to a new target box location. (This method resized the anatomy window, **Figure 76**)
5. Point to the center of an anatomy window displaying a blue target box, click on the target box, and drag the box to a new target box location. (This method moved the anatomy window, **Figure 77**)

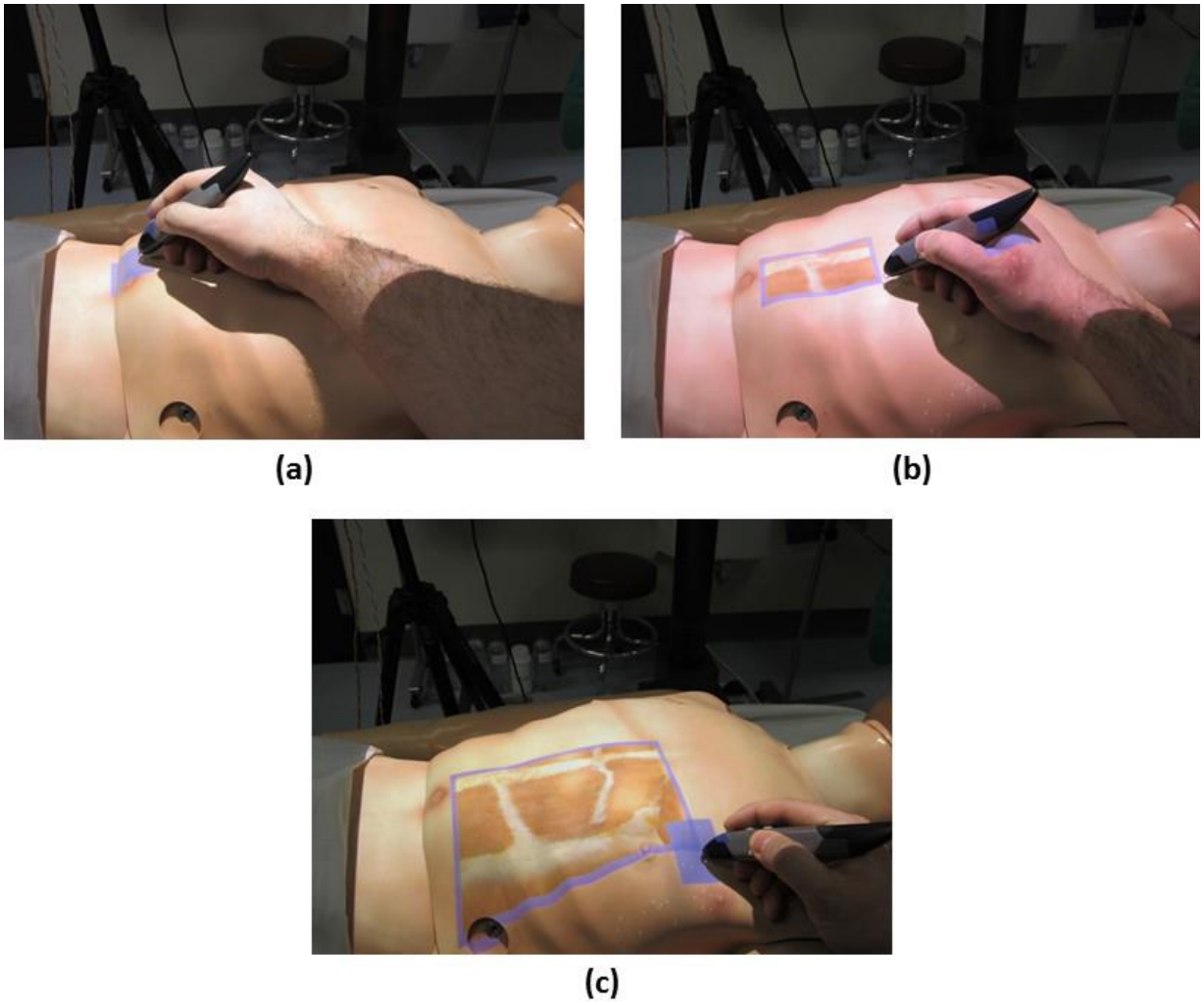


Figure 75. Instructions for opening an anatomy window: (a) point and click on target, (b) drag to new target, and (c) release over new target.

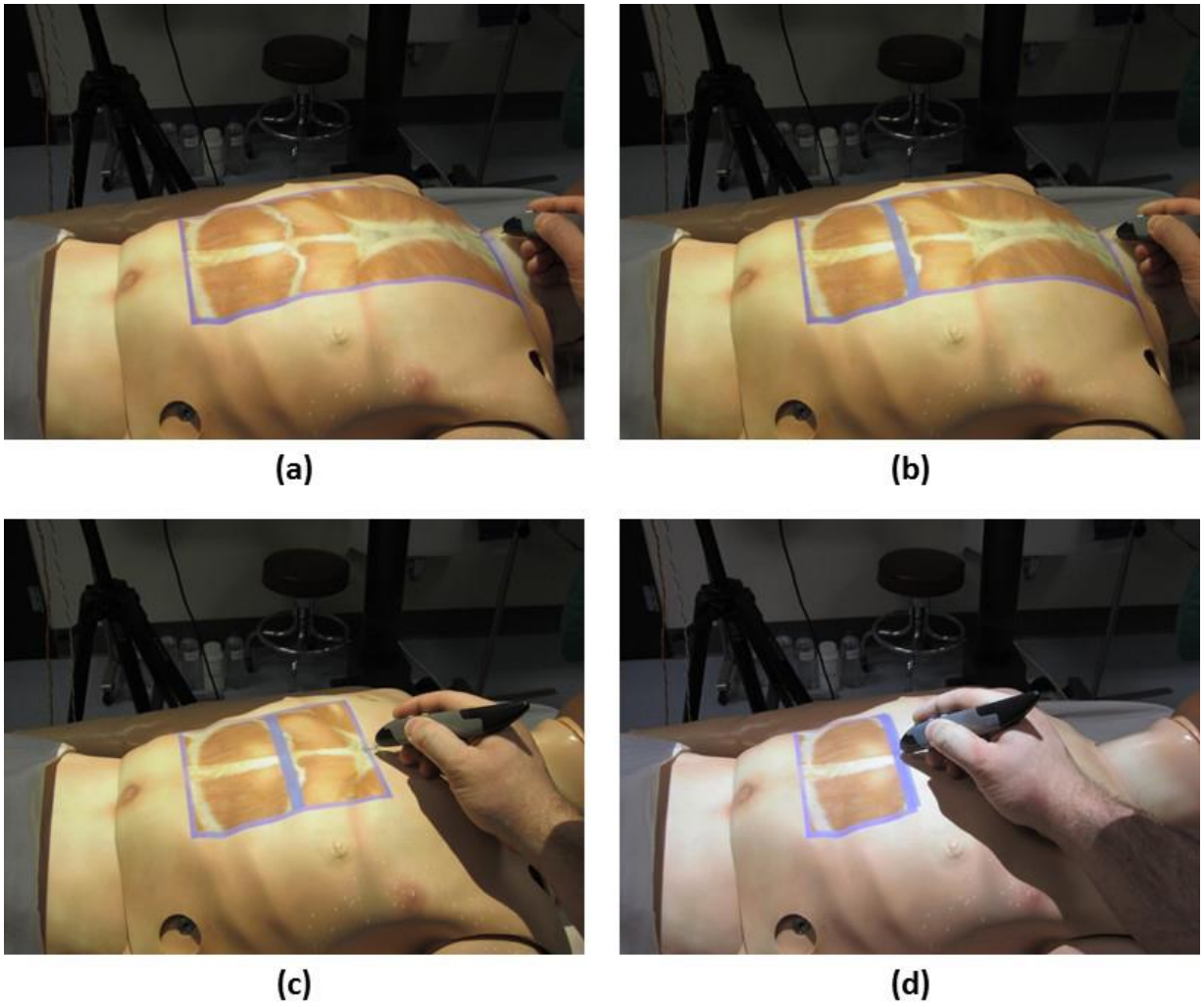


Figure 76. Instructions for resizing an anatomy window: (a) point to target, (b) click and drag to new target, (c) drag to new target, and (d) release over new target.

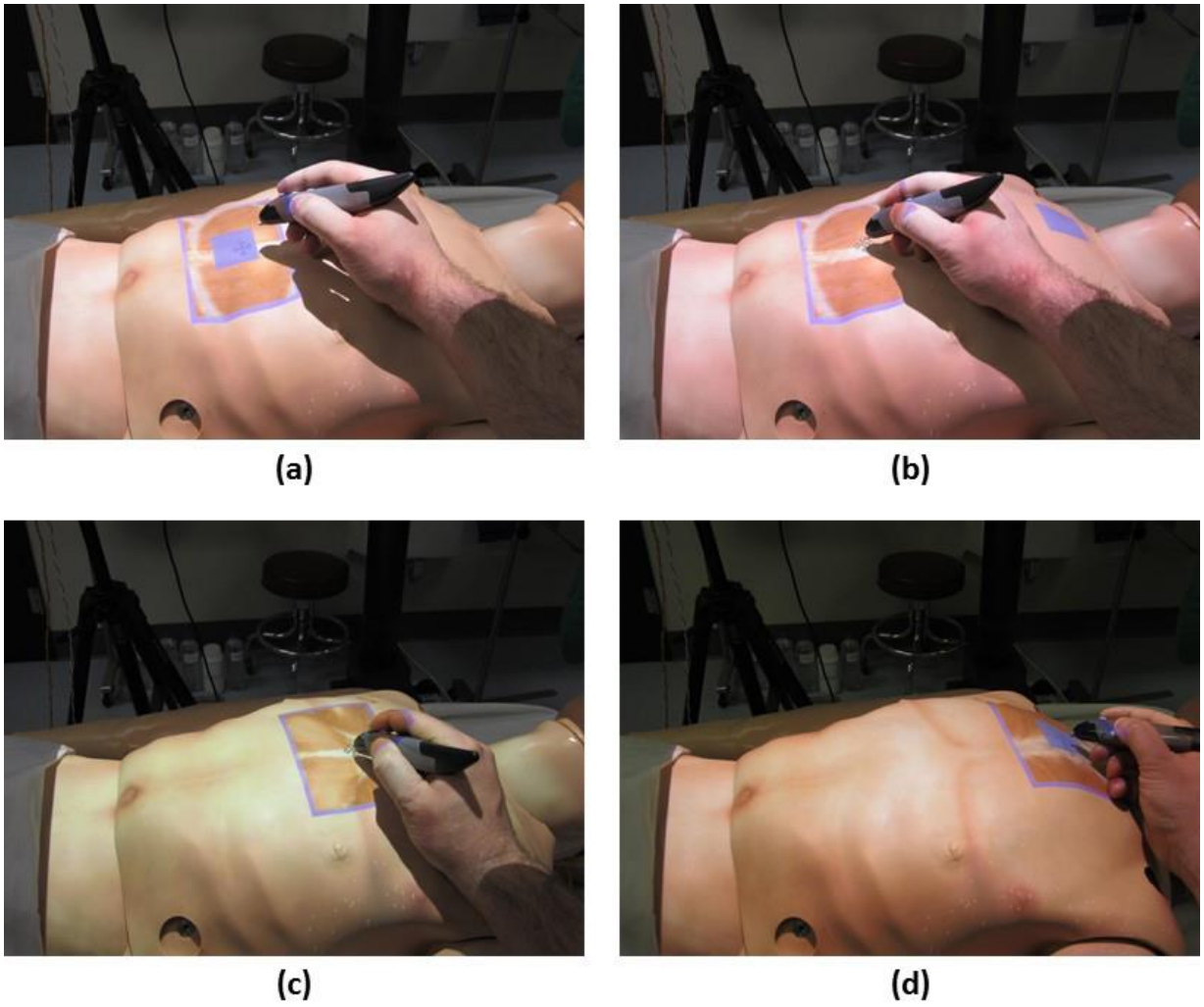


Figure 77. Instructions for moving an anatomy window: (a) point to target, (b) click and drag to new target, (c) drag to new target, and (d) release over new target.

After each individual set of instructions were given, the user was expected to perform the instructed actions before the next set of instructions were given. If the participant did not complete the instruction properly, verbal feedback was provided to the user of her/his mistake, and verbal instructions for corrective action were also provided.

5.4.1.4 Input Device Designs

Usability of the input device can be affected by poor tracking of the input device's tip moving with respect to the mannequin. Poor tracking of the tip could result from the ergonomics of the input device causing the user's hand and arm to shadow the tip of the input device from the tracking camera. Thus, a new design style for the input device was prototyped to test this hypothesis. The new design style, called the Wand-Like design due to its longer shape, extended the tip of the input device. This design enabled a user to keep the device's tip near the mannequin's surface while enabling her/his hand and arm to be further away from the mannequin, thus causing fewer shadows around the device's tip, as shown in **Figure 78**. The Wand-Like design style was compared with the previously designed input device—called the Pen-Like design for distinction between the two. The construction of the Wand-Like design had a foam-core shell wrapping the internal electronics from a second Pen-Like device. The IR-LED was extended with twisted-pair wire down to the Wand-Like design's tip, and the buttons were extended with wire leads to the outer surface of the foam-core.

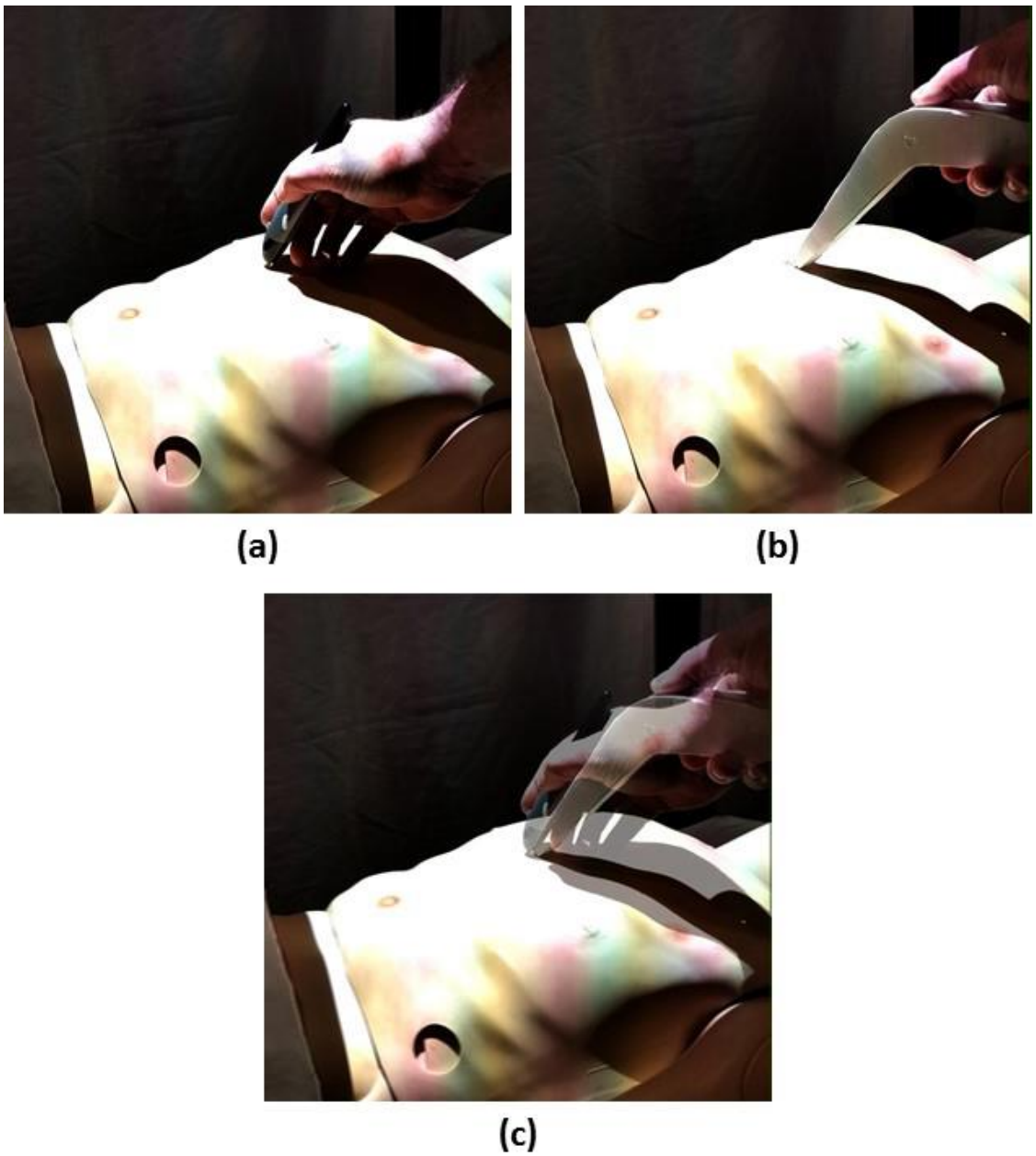


Figure 78. Pen-Like vs Wand-Like shadow comparison: (a) Pen-Like shadow, (b) Wand-Like shadow, (c) merged shadows showing less shadowing on the mannequin from Pen-Like design.

5.4.1.5 Location of Menus: On-Body vs On-Screen

The new software modifications gave the user two options for making selections from menus: on the surface of the mannequin (On-Body, **Figure 79, left**) and on the touchscreen of the auxiliary display (On-Screen, **Figure 79, right**).

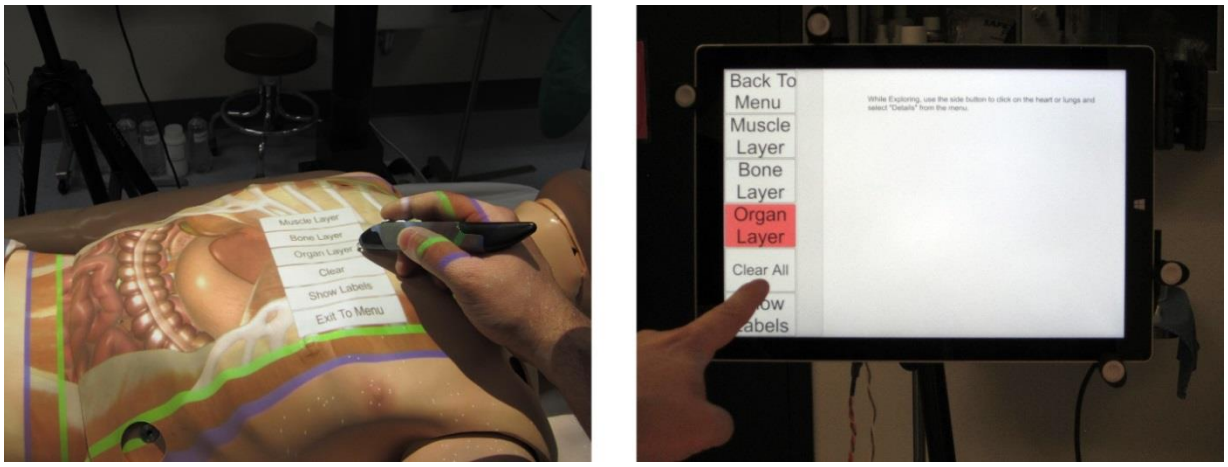


Figure 79. On-Body (left) versus On-Screen (right) menus.
(© 2016, Douglas A. Nelson Jr.)

5.4.2 Methods

5.4.2.1 Recruitment and Enrollment

After IRB approval, the third usability testing of BodyExplorer commenced. Participants were recruited using a convenience sample from the University of Pittsburgh School of Nursing. Instructors, graduate students, and undergraduate students were invited to participate in the study through mass email on Nursing School distribution lists and by word of mouth from usability

testing team members. As done in previous usability testing sessions, participants enrolled by signing up for an hour-long testing session using SignUpGenius [110].

5.4.2.2 Data Collection

Similar to UT1 and UT2, each hour-long testing session in UT3 was run by a session moderator, and at least two session observers were present to take observational notes. The observers and moderator were the same as UT1 and UT2, and sessions were recorded for post-session review and analysis. Quantitative analysis of the recordings was performed to assess usability with the two design styles for the input device and is presented in **Section 5.5**. A demographic tool (**Appendix C**) was used to collect information about the participant's background as well as the participant's previous educational and simulation experiences.

5.4.2.3 Session Protocol

The overall structure for each hour-long testing session was outlined previously in **Table 17**. The task list for Session 3 was modified from the Task List used in Sessions 1 and 2. A new task, Task 0, was added to the beginning of the Session for the user login experience and Input Device Tutorial. The modifications are outlined in **Table 21**. Task 1 was completed twice in order to evaluate both input device designs.

Table 21. Summary of tasks and subtasks performed during UT3

TASK	SUBTASK
Task 0: Login and Tutorial	A. Login to BodyExplorer
	B. Select Input Device Tutorial
	C. Complete Input Device Tutorial
Task 1: Windowing and IV Drug Administration	A. Open, move, and resize viewport windows
	B. Play heart sounds and show ECG
	C. Administer simulated medications
Task 2: Rapid Sequence Induction (RSI)	A. Administer simulated medications
	B. Apply cricoid pressure
	C. Perform endotracheal intubation

Participants performed counterbalanced testing of two design styles for the input device—Pen-Like versus Wand-Like. To account for potential ordering bias, counterbalancing was performed [106], alternating the first device design used by each participant across testing sessions. The first device was used to complete Task 0 and Task 1 Then, before moving on to Task 2, the participant was given the other device design and repeated Task 1. After repeating Task 1 with the other device design, the participant compared both device designs and was permitted to switch back and forth between using each design until the participant developed a preference. Once a device preference was achieved, the participant gave her/his rationale for the preference. If no device preference was achieved after 5 minutes, the participant provided their opinions regarding the positives and negatives about each design.

Within Task 1, the participant was required to select layers of anatomy from a menu to open sequential windows of anatomy on the mannequin. The session moderator allowed the participant to figure out how to make a menu selection on their own. If the participant was not able to figure out how to select a layer on their own, the moderator provided instruction to use the On-Screen menu. The participant was instructed by the moderator to try both On-Screen and On-Body menus if the participant did not come across both types on her/his own accord. The participant

compared both types of menus until she/he developed a preference. Once a menu preference was achieved, the participant was asked to give her/his rationale for the preference. If no menu preference was achieved after 5 minutes, the participant provided their opinions regarding the positives and negatives about each menu type.

After completing the comparison of menu types, the participant began Task 2 for RSI. If minimal time was left in the session, Task 2 was condensed to focus on application of cricoid pressure to receive user feedback on the new visualizations provided on the auxiliary display.

5.4.3 Results

5.4.3.1 Demographics

The participant demographics are shown in **Table 22**. All the undergraduate students were 3rd or 4th year students. All the graduate students were student registered nurse anesthetists (SRNA). Three of the four instructors were from the School of Nursing, while the fourth was an instructor from the School of Arts and Sciences. There were 3 participants who had previously used BodyExplorer.

Table 22. Participant demographics for UT3

User Group	N
Instructor	4
Graduate Student	4
Undergrad Student	8
TOTAL	16

Prior Experience…	N
Administering Medication	14
Endotracheal Intubation	11
Applying Cricoid Pressure	7
Using BodyExplorer	3

5.4.3.2 Feedback from Participants

User Login Experience

Overall, the user login experience was well received by the participants. Participants commented that they were expecting to see a computer workstation for login, but once the instructions appeared on the auxiliary display, login was intuitive. Some participants commented on the fidelity of the ID card reader mock-up being too flimsy. Others noted that the card reader was too far away when mounted on the auxiliary display. When questioned about alternative methods for login, participants suggested:

1. Username and password login on the auxiliary display,
2. An in-room computer workstation with either a username and password login, or a card reader login, and
3. Login on the body itself.

After completing the login, there were 3 participants that attempted to use the input device to touch the auxiliary display to select the learning module. However, they quickly reverted to using their finger once they realized the input device would not work with the auxiliary display. Despite 3 participants attempting to initially use the input device on the touchscreen menu, most participants commented that the touchscreen menu was intuitive to navigate for selecting the learning module.

Input Device Tutorial

All participants completed the Input Device Tutorial. Participants commented that the verbal instructions were well-paced and provided enough instruction for the participant to use the input device effectively. There were three primary improvements suggested by the participants:

1. functionality for repeating instructions,
2. a visual method for providing instructions, like a video showing someone else using the device, and
3. the aesthetics of the tutorial target boxes displayed on the body should be modified.

The shape of the targets should be changed from boxes, and the target object should be a different color from the initially displayed object. With the targets shaped like boxes, users tried to match the corner of the anatomy window to the outside corner of the target box, sometimes missing the target despite good intentions at reaching it.

Input Device Design and Location of Menus

Qualitative feedback was gathered from users with respect to the Pen-Like and Wand-Like input device designs. User preference for input device design was mixed. After the first 11 participants completed UT3, 2 participants preferred the Wand-Like, 3 participants preferred the Pen-Like, and 6 participants preferred features of both. The common feedback from those who preferred the Pen-Like Input design and those without preference preferred the size of the Pen-Like design over the Wand-Like design. The Wand-Like design was *"too big"* and *"too bulky"*. Those who preferred the first Wand-Like design and those without preference preferred how their arm was not in the way of the projected images when using the Wand-Like design; participants could see more of the projected anatomy and physiology on the body.

Due to the abundance of user comments regarding the size of the Wand-Like design, a second Wand-Like design was quickly prototyped and incorporated into the sessions for the final 5 study participants in order to obtain more qualitative comparisons about the different designs.

The second Wand-Like design was less bulky than the first, and the design comparison is shown in **Figure 80**, with the first design on the top and the second design on the bottom.

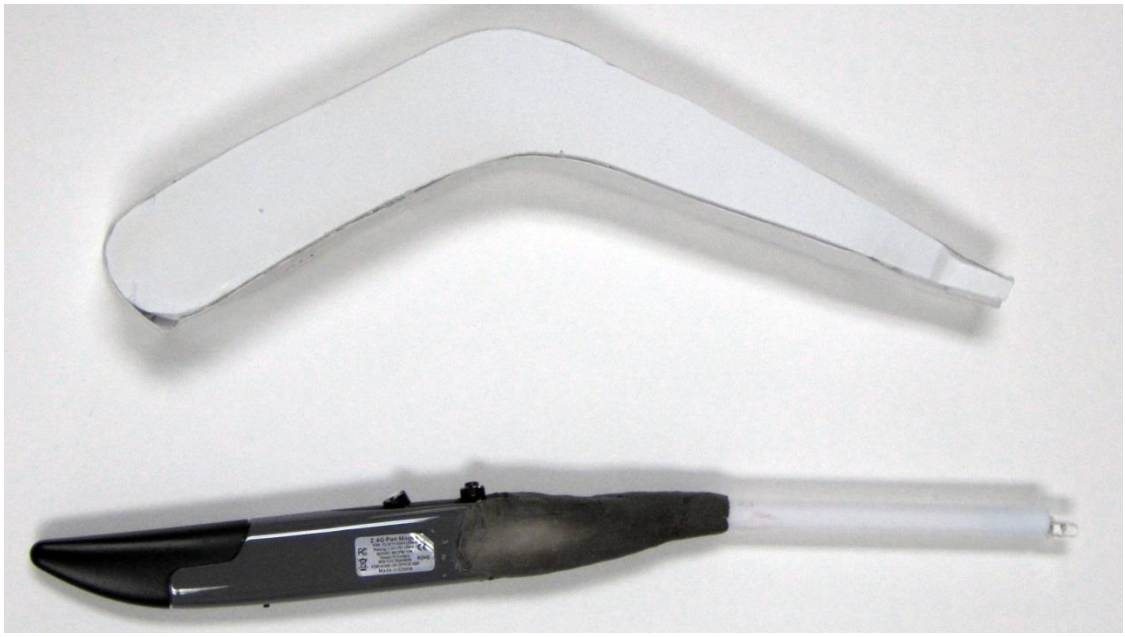


Figure 80. First Wand-Like design (top) vs second Wand-Like design (bottom).
(© 2016, Douglas A. Nelson Jr.)

Using this second Wand-Like input device design, three of the five preferred the Wand-Like, and the other two preferred the Pen-Like. The primary reason the other two participants preferred the Pen-Like design was because they felt that the second Wand-Like design was not comfortable to hold, the buttons were tougher to press, and the design felt more *"flimsy"* than the Pen-Like design. Quantification of the usability results are presented in **Section 5.5**.

The On-Body menus were preferred over the On-Screen menus by 11 out of 16 participants, as shown in **Table 23**. There were 3 participants that preferred the On-Screen menus and 2 participants were indifferent to either method.

Table 23. Summary of participant preferences for menu location

PARAMETER	PREFERENCE	COUNT
Menu Location	On-Body	11
	On-Screen	3
	Features of Both	2

Of those who preferred the On-Body menus, the primary reason was that they could keep their focus on the body and not have to look away to another screen, even when the auxiliary display was moved to other locations around the user. There were two criticisms of the On-Body menus. Some participants suggested using a button to bring up the menu on the body rather than the scroll-wheel. All participants commented on the difficulty to see the menu on the body when the menu was brought up in a location near the side of the body or on the shoulder, as shown in **Figure 81**.

Figure 81. On-Body menus in locations where the menu falls off the surface of the mannequin
(© 2016, Douglas A. Nelson Jr.)

Cricoid Pressure Application

Fewer than half of the total participants had enough time to reach Task 2 during the session. **Table 24** shows that all the graduate student SRNAs, half of the instructors, and only 1 undergraduate

student reached Task 2. These participants had previous experience applying cricoid pressure either in simulation or on human patients.

Table 24. User groups for participants reaching Task 2

User Group	N
Instructor	2
Graduate Student	4
Undergrad Student	1
TOTAL	7

Overall, participants preferred the visual feedback mechanisms provided on the auxiliary display. Participants unanimously commented that the red and green cricoid colors made sense for pressure over and within threshold, respectively. The blue cricoid color corresponding to inadequate pressure demonstrated no consensus as to whether the coloration provided meaningful feedback. Some participants suggested yellow or white as an alternative, but others did not have suggestions for improvement. The color gradient was a visual distraction for 3 of the participants, voicing preference for discrete colors, while the other 4 participants appreciated the color gradient. Participants also commented on preferring numerical ranges, like a bar-graph, rather than a number to represent the amount of force applied.

5.4.4 Discussion

The proximity detection, simulated ID-based login, and On-Screen menu selection was intuitive for most participants. These results are encouraging for allocating more resources into continued development to bring these features towards more robust implementations, including a real card reader. Practically speaking, ID-based login might be challenging to implement large-scale across

many institutions. Some institutions might use magnetic strip ID cards, others might use RFID cards, while others might not have any built-in technology for unique identification other than a picture and name. Even if all institutions used unique technologic identifiers, integration with all of the different systems could pose problems, unless a hardware/software application interface was provided so that institutions could add their own front-end ID readers for user login. It was interesting to find that some participants were still searching for a computer workstation for login, and even suggested username and password as an alternative to IDs. The affordances of the desktop computer are widespread. A computer interface with username and password would remove some of the challenges of scalability across institutions brought about by an ID-based login system. Having become available in mobile smartphones in recent years, finger print readers could be a viable alternative login method. Similarly, leveraging the wireless connectivity to BodyExplorer and the cross-platform development capabilities of the Unity environment, a personal mobile phone with a BodyExplorer app could also interface with BodyExplorer and be used for user login.

Based upon the qualitative feedback of the three input device designs (Pen-Like, first Wand-Like, second Wand-Like), there were a few common themes. The size and button-feel of the Pen-Like design was preferable over both Wand-Like designs. Participants appreciated how the length of the second Wand-Like design moved the hand away from the body and caused less shadows on the body, but the ergonomics were still not right. Namely, the buttons should be flush with the body of the device and a "less flimsy" prototype should be created—possibly a new 3D-printed shell instead of the retrofitted mock-up with clay. New interactive projectors that incorporate a built-in input device, such as the Dell Interactive Short-Throw Projector model S320wi shown in **Figure 82**, have become available in consumer markets over the time that BodyExplorer has been in development.

Figure 82. Dell interactive short-throw projector, model S320wi
(photo courtesy of Dell)

It would be worthwhile to integrate a new interactive projector, use its built-in input device with BodyExplorer, and assess usability. Using the interactive projector would lower the amount of hardware necessary for setup, and could lead to an easier setup for end-users (even though that wasn't an objective during this round of usability testing).

Participants preferred using the On-Body menus versus the On-Screen menus even when the auxiliary display was moved to different locations around the mannequin as the participant directed. Participants preferred to keep their focus of attention on the body rather than looking away. The On-Body menus should be created on projection surfaces that are mostly flat so that the options are easy to read and select. **Figure 83** shows an example of a good location for an On-Body menu which can be contrasted with the locations shown in **Figure 81** that overhang the body edges. While the users preferred the On-Body menus, the On-Screen menus are still useful for selecting learning modules from the main menu. As such, there could be a configuration option available for the user to select menu style based upon preference.

Figure 83. Menu in a good, flat location on the body
(© 2016, Douglas A. Nelson Jr.)

In regards to evaluation of the built-in automated instruction features, participants desired a combination of both verbal and visual instructions for clarity. A key aspect of instruction was not incorporated into the automated instruction features piloted—demonstration. Participants wanted to watch someone else perform actions as part of the instruction set before she/he would perform the action. Possible future implementations of demonstration could be shown by a video playing on the auxiliary display or by computer-driven animations projected directly onto the body itself. The latter method would require instruction for the participant to step back and watch so as not to obscure the projected instructions—a limitation of using projection. Incorporating graphical and pictorial methods for demonstration leads to more expressive instructional methods than

verbal alone. Furthermore, like a quality instruction manual for assembling a new piece of furniture or a child's toy, the graphical and pictorial methods could remove language barriers.

5.5 QUANTITATIVE ANALYSIS OF INPUT DEVICE USABILITY

The primary design goals for BodyExplorer require the system to be easy to use. Design iterations have incorporated feedback from end-users towards meeting the design goal. BodyExplorer includes a novel interface for interacting with images projected onto the surface of a mannequin using a custom input device. In order to quantify usability over the design iterations, analysis was performed on the video recordings from each usability testing session. Specifically, analysis was performed regarding the usability of the input device across sessions. The following sections present the methods, results, and discussion for quantifying input device usability.

5.5.1 Methods

5.5.1.1 Quantifying Input Device Usability

The video recordings systematically were analyzed to provide a quantitative measure for usability of the input device between the three usability testing sessions. The primary measures used to define input device usability were:

1. time that the user was using the input device—the Total Time,
2. time that the user was having difficulty using the input device—the Difficulty Time, and
3. time that the moderator was providing help to the user—the Help Time.

The measures, Difficulty Time and Help Time, were normalized by dividing by the Total Time to control for variations in Total Time between users. The normalization resulted in the outcome variables, Percent Difficulty and Percent Help, respectively. A standardized method for collecting and analyzing these data from the video recordings was developed and is described in the following sections.

5.5.1.2 Data Collection

Data collection from the video recordings was performed using NVivo 11 (QSR International, Doncaster, Victoria, Australia) by three trained research assistants (the "raters"). To prevent potential bias in the video data analysis, raters that were not previously involved in any aspect of BodyExplorer development, usability testing, or data collection were chosen. Raters were also not privy to the design modifications of the input device across usability testing sessions or to the ordering of the sessions.

The NVivo software is a research tool that supports quantification of qualitative and mixed methods research [112]. NVivo provides functionality for quantifying videos that have been coded within the software. Video coding is the process of tagging segments of videos with predefined classification variables, or nodes, following a coding schema. A coding schema requires standardized operational definitions to be established for each node in order to promote consistent coding. The total time covered by nodes tagged in a video is calculated within NVivo and can be used for analysis within or across multiple videos. For reference, **Figure 84** shows the NVivo user interface with an example of video coding a usability testing session. Using the interface, raters can highlight time segments from the video and code them to nodes based upon the coding schema.

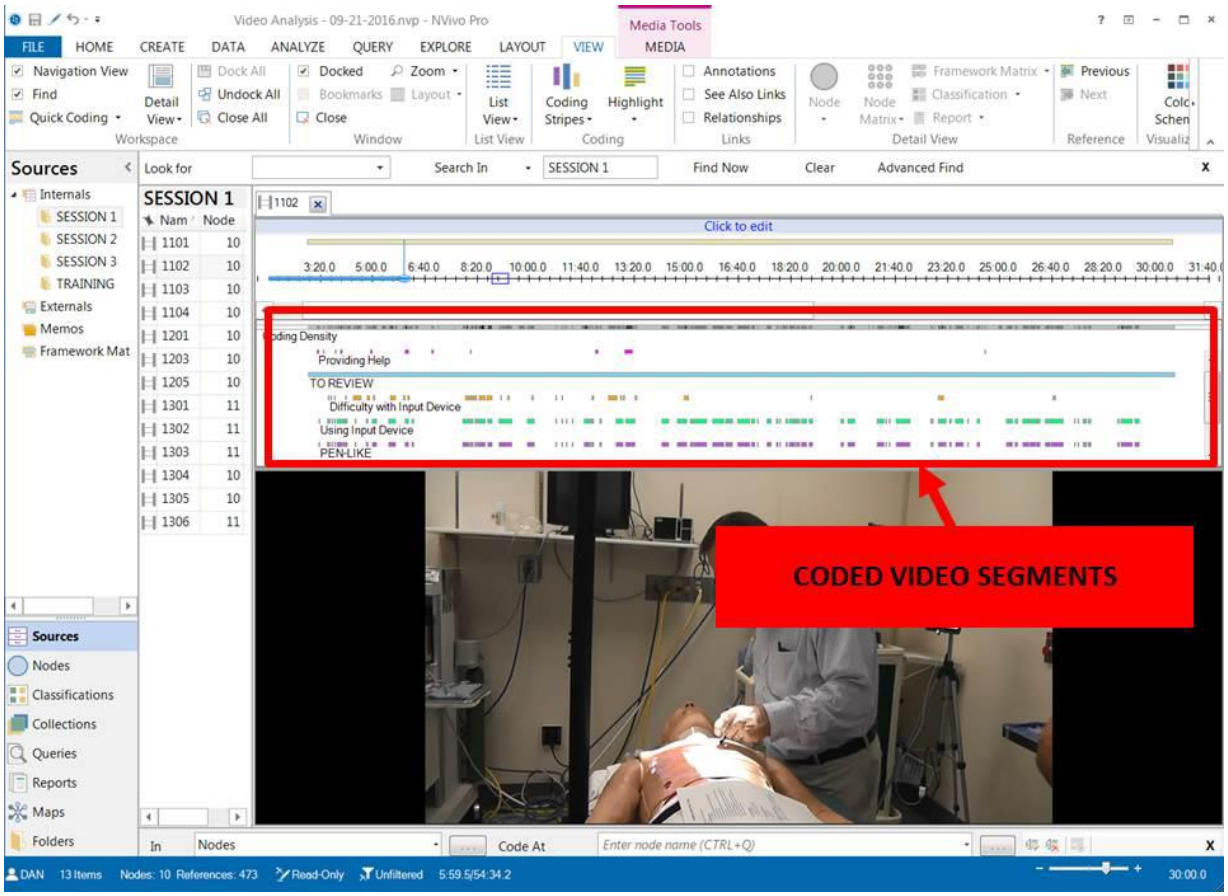


Figure 84. NVivo interface for coding Usability Testing Session videos.
(© 2016, Douglas A. Nelson Jr.)

A list of operation definitions and a coding schema for the video recordings were developed by the PI for the raters to use to standardize the coding of videos in NVivo. The operational definitions and coding schema are included in **Appendix D**. Nodes were defined for coding video segments as:

1. Using Input Device, which had further sub-nodes of Pen-Like and Wand-Like,
2. Difficulty with Input Device, and
3. Providing Help.

These nodes corresponded to the primary measures Total Time, Difficulty Time, and Help Time, respectively, and could be categorized based upon input device used.

5.5.1.3 Rater Training

Raters were trained by the PI on how to code the videos in NVivo using the operational definitions and coding schema. Excerpts from the usability testing session videos were used as examples to instruct the raters on what the operational definitions looked like in action. During training, four full-length videos from usability testing sessions were coded by all raters and the PI, and the results were compared for interrater agreement. Interrater agreement was measured according to Percent Absolute Agreement and Cohen's Kappa. The goal during training was to achieve a minimally acceptable Percent Absolute Agreement and Cohen's Kappa of 90% and 0.50, respectively. These goals were established based upon recommendations from literature [113, 114] and within NVivo [115]. Coding comparisons were performed using NVivo and exported to Excel for visualizing the comparison data.

5.5.1.4 Distribution of Videos for Coding

After training, the remaining usability testing session videos were divided among the raters for coding, as shown in **Figure 85**. The coding was performed in three stages in which videos were pseudo-randomly assigned to each rater. Pseudo-random video assignment was performed rather than fully random assignments to ensure that the videos from each of the three rounds of usability testing sessions were equally distributed among raters and across the stages. For each stage in **Figure 85**, each rater has 4 assigned videos with equal distributions of videos from UT1, UT2, and UT3 among raters at each stage. At the beginning of each stage, all raters and the PI coded a "control video" which will be described further in the next section.

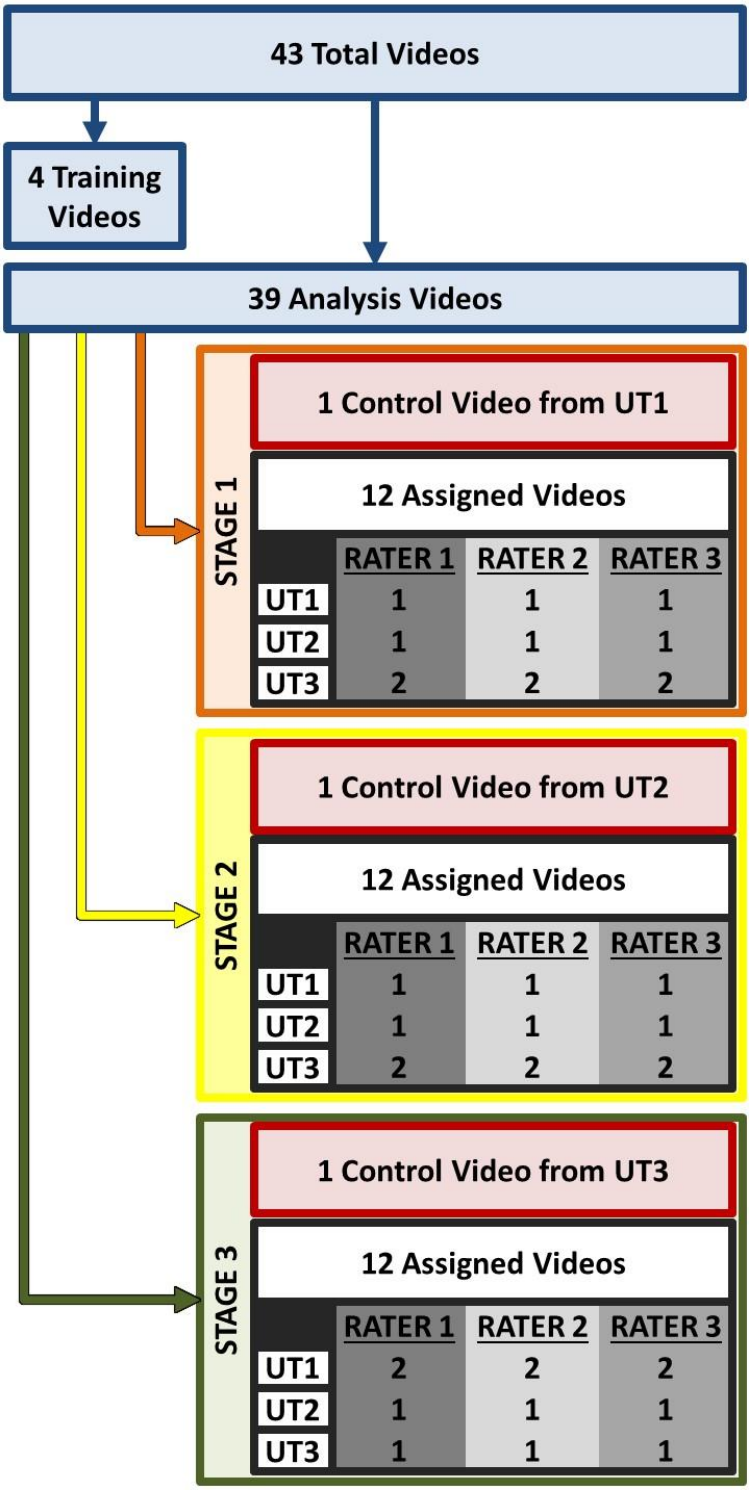


Figure 85. Distribution of videos for analysis
(© 2016, Douglas A. Nelson Jr.)

5.5.1.5 Coding Validity Checks – Control Videos

During each stage, all raters had 5 usability testing session videos to code. One of the videos during each stage was the same for all raters (the "control video"); the other four were different for each rater (the "assigned videos"). The control video was also coded by the PI, and it was used to measure interrater agreement across raters and PI during each stage. Interrater agreement for each control video between the three stages of video coding supports that there was coding validity across raters and stages, and also supports the validity of data coded within the assigned videos. At the beginning of each stage of video coding, the raters and PI were required to meet minimally acceptable interrater agreement on the control video for that stage. The minimally acceptable Percentage Absolute Agreement and Cohen's Kappa of 90% and 0.50, respectively, were required before coding the assigned videos for that stage could begin. Coding comparisons were performed using NVivo and exported to Excel for visualizing the comparison data.

5.5.1.6 Data Analysis

Coding data from all raters across all stages of the video coding were collected into a single NVivo project. The combination of the data preserved the ability to review coding from each individual rater while also allowing the data to be queried for quantification of the coding within each video. Queries were performed on the coded videos in order to extract the timing data from each video for:

1. Using Input Device, which had further sub-nodes of Pen-Like and Wand-Like,
2. Difficulty with Input Device, and
3. Providing Help.

Query results were manually transferred from NVivo into an Excel Spreadsheet to combine all the results, and then the data were imported into SPSS for statistical analysis (IBM, SPSS 24).

5.5.2 Results

5.5.2.1 Coding Validity Checks – Control Videos

The complete results from the coding validity checks performed on the control videos are shown in **Appendix D**. All values of Percent Absolute Agreement were greater than 90%. All values of Cohen's Kappa except for 7 were above 0.50, with all but 2 values greater than 0.40 (which is NVivo's recommended lower threshold for determining "fair to good agreement" and a "moderate" agreement by [114]). All cases where Cohen's Kappa fell below the threshold of 0.50 occurred at the very beginning of the video coding protocol during coding of the control video from stage 1—at the very beginning of coding, just after completing the training videos. Discrepancies in the coding were reviewed by the raters and PI against the operational definitions and coding schema before continuing with the assigned videos of stage 1. The good agreement from control videos of stages 2 and 3 (signified by Cohen's Kappa values greater than 0.50) suggest that the discrepancies from stage 1 were effectively rectified.

5.5.2.2 Quantified Measures of Usability

The primary measurements extracted from the video coding were:

1. Total Time,
2. Difficulty Time, and
3. Help Time for each type of Input Device.

From those measurements, ratios were calculated to determine Percent Difficulty and Percent Help for each type of input device, as defined in **Section 5.5.1.1**. A summary of the descriptive statistics comparing metrics of Pen-Like input device usability across all usability testing sessions is shown in **Table 25**. The complete results for all individual videos are shown in **Appendix D**.

Table 25. Descriptive statistics for Pen-Like input device usability for all usability testing sessions

Measure	Session	N	Median	Mean ± SD	95% CI Lower	95% CI Upper
Total Time	UT1	13	338	343.1 ± 139.6	258.7	427.4
	UT2	8	244.5	265.3 ± 119.3	165.5	365.0
	UT3	16	346.5	384.0 ± 222.3	265.5	502.5
Difficulty Time	UT1	13	85	95.5 ± 61.2	58.5	132.5
	UT2	8	42.5	61.1 ± 54.1	15.9	106.4
	UT3	16	18.5	39.4 ± 56.9	9.1	69.8
Help Time	UT1	13	17	28.7 ± 30.6	10.2	47.2
	UT2	8	30	29.1 ± 20.5	12.0	46.3
	UT3	16	3	8.3 ± 11.4	2.2	14.4
Percent Difficulty	UT1	13	23%	27% ± 11%	21%	33%
	UT2	8	16%	21% ± 14%	10%	33%
	UT3	16	5%	9% ± 11%	3%	15%
Percent Help	UT1	13	6%	9% ± 10%	3%	15%
	UT2	8	10%	12% ± 8%	5%	19%
	UT3	16	1%	2% ± 3%	1%	4%

* all times reported in seconds.

Boxplots were created to visualize the distributions, variability, and outliers of the data. **Figure 86** shows the boxplot for Percent Difficulty across all sessions using the Pen-Like device. An extreme outlier (>3 standard deviations from the mean) exists in the data for UT3. Likewise, there are outliers in the data for the other measures. After the video coding references and the numerical summary data were confirmed for accuracy, analysis continued without removing the outliers or performing transformations on the data. However, instead of conducting one-way ANOVAs on the data, the Kruskal-Wallis test was conducted due to its robustness to outliers when compared with ANOVA [116, 117].

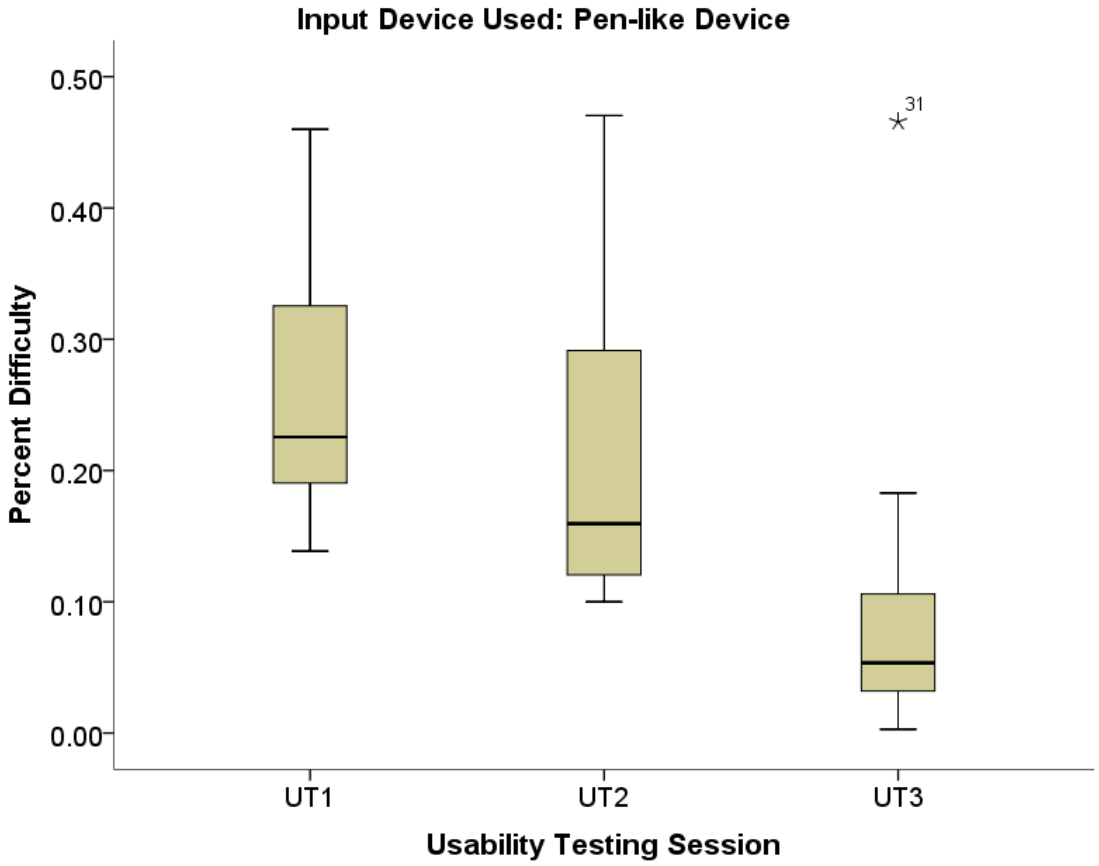


Figure 86. Boxplot for Percent Difficulty in using Pen-Like Device across Sessions.

The Kruskal-Wallis test was conducted to determine whether there were differences in the defined usability measures between sessions when participants were using the Pen-Like input device. Distributions of the usability measures were not similar for all groups, as assessed by visual inspection of their respective boxplots. Thus, mean rank of the distributions were assessed for significant differences. The usability measures of Difficulty Time, Help Time, Percent Difficulty, and Percent Help were significantly different between the different Sessions; whereas Total Time was not significantly different. Thus, the differences in Difficulty Time, Help Time, Percent

Difficulty, and Percent Help cannot be attributable to differences in Total Time using the Pen-Like device across sessions. The χ^2 results of the Kruskal-Wallis test are summarized in **Table 26**.

Table 26. Kruskal-Wallis test results for usability of Pen-Like input device across all sessions

Measure	Kruskal-Wallis Test Result	Significance
Total Time	$\chi^2(2) = 2.557$	$p = 0.278$
Difficulty Time	$\chi^2(2) = 11.320$	$p = 0.003$*
Help Time	$\chi^2(2) = 9.411$	$p = 0.009$*
Percent Difficulty	$\chi^2(2) = 18.777$	$p < 0.0005$*
Percent Help	$\chi^2(2) = 13.296$	$p = 0.001$*

*Significant at the $p < 0.05$ level

Subsequently, pairwise comparisons were performed using Dunn's procedure with a Bonferroni correction for multiple comparisons. The post hoc analysis revealed significant differences in Difficulty Time between UT3 and UT1, as well as in Help Time, Percent Difficulty, and Percent Help between UT3 and both UT1 and UT2. There were not significant differences in any of the usability measures between UT1 and UT2 or any other group combination. Because the distributions of the measures were not similar for all groups across sessions, the significant results are valid for mean rank ordering only—not quantified differences between the medians.

The results from the post hoc analysis are summarized in **Table 27**. The results from these analyses signify the mean rank ordering between significant results. Interpreting these results, participants had significantly less Difficulty Time, Help Time, Percent Difficulty, and Percent Help when using the Pen-Like Device in UT3 compared to UT1. These results were also true for UT3 when compared to UT2 for measures of Help Time, Percent Difficulty, and Percent Help.

However, there were not significant differences between any measure comparisons between UT1 and UT2.

Table 27. Pairwise comparisons across Sessions for usability measures

Measure	Pairwise Comparison (Mean Rank)	Adjusted Significance**
Difficulty Time	UT3 (12.62) and UT2 (20.12)	p = 0.328
	UT3 (12.62) and UT1 (26.15)	p = 0.002*
	UT2 (20.12) and UT1 (26.15)	p = 0.645
Help Time	UT3 (12.88) and UT2 (25.25)	p = 0.044*
	UT3 (12.88) and UT1 (22.69)	p = 0.024*
	UT2 (25.25) and UT1 (22.69)	p = 1.000
Percent Difficulty	UT3 (10.38) and UT2 (22.75)	p = 0.025*
	UT3 (10.38) and UT1 (27.31)	p < 0.0005*
	UT2 (22.75) and UT1 (27.31)	p = 1.000
Percent Help	UT3 (11.91) and UT2 (27.50)	p = 0.025*
	UT3 (11.90) and UT1 (22.50)	p = 0.002*
	UT2 (27.50) and UT1 (22.50)	p = 0.905

* Significant at the $p < 0.05$ level

** Significance values have been adjusted by the Bonferroni correction for multiple tests.

Within UT3, the same measures for usability were assessed between Pen-Like and Wand-Like input device designs when used by participants. A summary of the descriptive statistics comparing metrics of Pen-Like vs Wand-Like input device usability is shown in **Table 28**.

Table 28. Descriptive statistics for Pen-Like vs Wand-Like input device usability

Measure	Device	Median	Mean ± SD	95% CI Lower	95% CI Upper
Total Time	Pen-Like	346.5	384.0 ± 222.3	265.5	502.5
	Wand-Like	298.5	325.6 ± 174.7	232.5	418.7
Difficulty Time	Pen-Like	18.5	39.4 ± 56.9	9.1	69.8
	Wand-Like	13.0	23.2 ± 26.7	9.0	37.4
Help Time	Pen-Like	3.0	8.3 ± 11.4	2.2	14.4
	Wand-Like	1.5	10.6 ± 17.7	1.1	20.0
Percent Difficulty	Pen-Like	5%	9% ± 11%	3%	15%
	Wand-Like	4%	7% ± 8%	3%	11%
Percent Help	Pen-Like	1%	2% ± 3%	1%	4%
	Wand-Like	1%	5% ± 12%	0%	12%

Similar to the previous analysis, boxplots were created to visualize the distributions, variability, and outliers of the data. **Figure 87** shows the boxplot for Percent Difficulty across all sessions using the Pen-Like device. An extreme outlier (>3 standard deviations from the mean) exists in the data for the Pen-Like design. Likewise, there are outliers in the data for the other measures. Thus, the Mann-Whitney U test was conducted due to its robustness to outliers and because there were only two categories to compare between (Pen-Like vs Wand-Like).

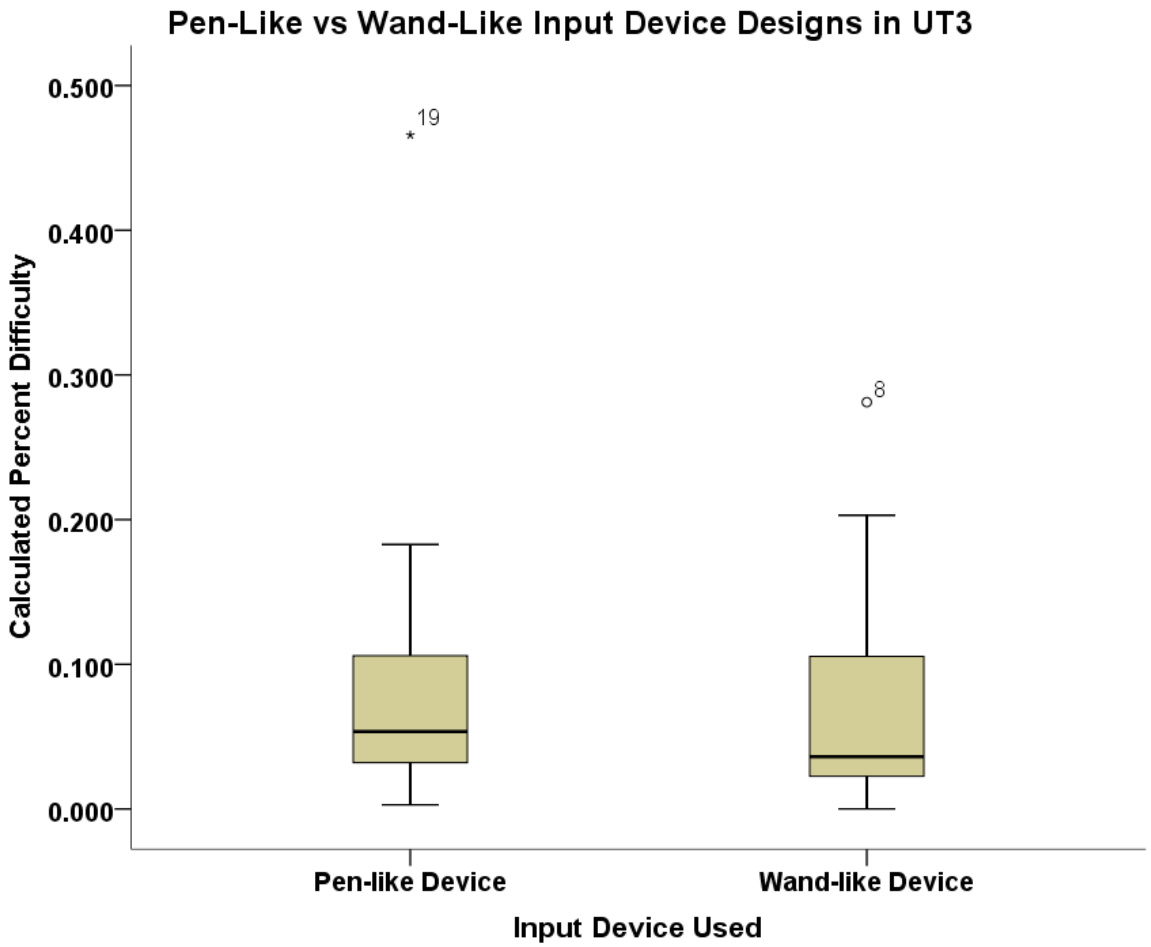


Figure 87. Boxplot for Percent Difficulty between using Pen-Like vs Wand-Like input device designs in UT3.

The boxplots illustrate that there were not significant differences in usability measures between Pen-Like and Wand-Like input device designs. The results are summarized in **Table 29**.

Table 29 Mann-Whitney U test results for usability of Pen-Like vs Wand-Like input device during UT3

Measure	Mann-Whitney U Test Result	Significance
Total Time	U = 111.0, z = -0.641	p = 0.539
Difficulty Time	U = 109.0, z =-0.717	p = 0.491
Help Time	U = 124.5, z = -0.139	p = 0.897
Percent Difficulty	U = 112.0, z =-0.603	p = 0.564
Percent Help	U = 130.0, z = 0.080	p = 0.956

Importantly, the same results hold true when the statistical analysis is performed after controlling for order of input device use and the potential effects of the input device type used first during the sessions.

5.5.3 Discussion

By comparing the defined measures of usability for the Pen-Like input device across usability testing sessions, significant differences were observed between the distributions of Difficulty Time, Help Time, Percent Difficulty, and Percent Help when comparing UT1 and UT3. The mean rank of UT3 was significantly lower than the mean rank of UT1 for Difficulty Time, Help Time, Percent Difficulty, and Percent Help (12.62 vs 26.15, 12.99 vs 22.69, 10.38 vs 27.31, and 11.91 vs 22.50, respectively). It is also noteworthy that Total Time was not significantly different across UT1, UT2, and UT3; meaning, users did not spend significantly more or less time with the Pen-Like device across usability testing sessions. These results support that the system was easier to use in UT3 than in UT1, as assessed by the defined measures related to input device.

While these results support that the system was easier to use in UT3 than in UT1, the results are not necessarily causally attributed to input device alone. The entire software was redesigned

for UT3 based upon feedback received across UT1 and UT2. Modifications were made to the affordances of the interface (**Section 5.4.1.1**) in addition to providing the Input Device Tutorial (**Section 0**). The significant results support that the modifications made for UT3 were improvements to the overall usability of the system.

One alternative possibility is that the significant results derive from the fact that two different input device designs were used in UT3, whereas only one input device design was used in UT1 and UT2. The measure of Total Time with the Pen-Like device was not significantly different between sessions. Thus, the significant difference for the other measures cannot be attributable to the time using the input device. However, it is possible that the result could derive from ordering of the input device used—if a participant used the Wand-Like device first, she/he might have learned from that experience and had less difficulty using the Pen-Like device for the remainder of the session. Therefore, further analysis was done controlling for order of input device used.

Analysis of the data was performed using the same methodology as before while excluding the cases where the Wand-Like input device was used by the user first. The one-way Kruskal-Wallis test resulted in significant differences between the means for Difficulty Time ($\chi^2(2) = 6.420$, $p = 0.04$), Percent Difficulty ($\chi^2(2) = 13.515$, $p = 0.001$), and Percent Help ($\chi^2(2) = 6.283$, $p = 0.043$), but not for Total Time ($p = 0.089$), or Help Time ($p = 0.278$)—neither group had more Total Time with the device (despite completing the tutorial in UT3), and neither group received more Total Help from the moderator. Pairwise comparisons using Dunn's procedure with Bonferroni correction for multiple comparisons resulted in significant differences between Total Difficulty across UT3 and UT2 (mean rank of 8.50 vs 19.12, $p = 0.038$), Percent Difficulty across UT3 and UT1 (mean rank of 5.14 vs 19.31), and Percent Help across UT3 and UT1 (mean rank of

8.86 vs 18.38). Thus, correcting for potential bias from using Wand-Like input device before Pen-Like input device still showed significantly lower Percent Difficulty with the Pen-Like input device in UT3 compared to UT1. A boxplot shows this result, visually, in **Figure 88**. The results from this follow-up analysis correcting for potential bias caused from using Wand-Like input device before Pen-Like input device further support that the system was easier to use in UT3 than in UT1.

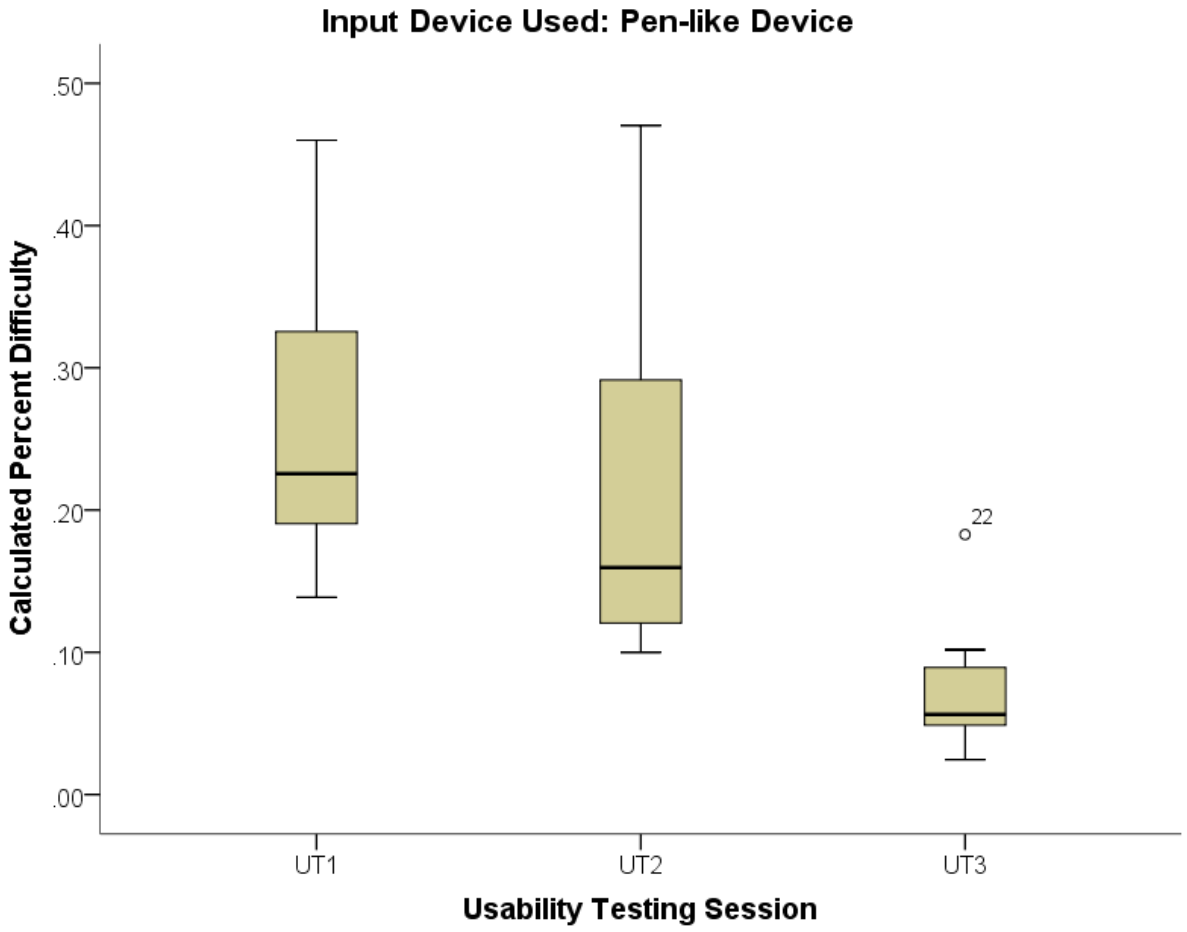


Figure 88. Boxplot for Percent Difficulty in using Pen-Like device across sessions excluding cases in UT3 where Wand-Like device was used before Pen-Like device

The results from quantifying usability across Pen-Like versus Wand-Like input device designs supported the qualitative findings reported in **Section 5.4.3**. Users reported not having a clear preference for either the Pen-Like or Wand-Like design, which is supported by the quantified results on usability not being significantly different between designs. While the quantified results did not support either design as superior, there was still valuable information gleaned from the qualitative feedback received from the users to guide future input device design decisions, as well as lessons learned to guide future comparative testing between designs. Future comparative testing should be performed where evaluation of the input device designs is the sole purpose of the experiment, using guidance from the established keystroke-level model for evaluating human-computer interactions because this model measures one performance task at a time [118].

6.0 AUTOMATED INSTRUCTION DEVELOPMENT

This chapter presents the prior art in automated instruction development within other disciplines, as well as development specifically within the healthcare field related to providing real-time feedback based upon measured user actions. This chapter also provides details about how the automated instruction within BodyExplorer is driven. Details on specific methods employed to provide automated instruction are defined. Finally, this chapter presents development of automated instruction for two clinically relevant learning modules, incorporating feedback received from end-users during usability testing sessions.

6.1 INTRODUCTION AND PRIOR ART

Automated instruction is a rapidly growing field. It is not necessarily meant to replace instructors or human tutors; rather, automated instruction is a means to multiply instructional capacity and bring efficiency and customized learning to the classroom and other educational experiences. Leveraging today's inexpensive consumer electronics that include immense computational power and memory, automated instruction platforms are becoming more widely available.

In the 1980s and 1990s, automated instruction systems were developed within numerous different educational domains. A sampling of early successful examples included tutors for students in electronic troubleshooting (SHERLOCK, [119]), mathematics (Pittsburgh Urban Mathematics Project (PUMP) Algebra Tutor, called PAT [120]), and computer programming (LISP Tutor, [121]). As concluded within the references for these systems, educational outcomes

and efficiencies were improved for students when using the tutoring systems. In a later review of tutoring systems spanning 35 years of research from 1975 to 2010, data supported that intelligent tutoring systems were nearly as effective as human tutors [122].

Historically, intelligent tutoring systems were implemented as SBSs that allowed the student to interact with the tutor using a mouse and keyboard. These systems exercised the user's cognitive abilities, but for tasks that also require hands-on skill acquisition, the model provided by these systems neglects the physical and psychomotor skills that are also necessary for mastery in some fields. When treating patients in a clinical setting, both cognitive and psychomotor skills are required for mastery and to successfully treat patients. In order to create a tutoring system that encompasses both the cognitive and psychomotor skills necessary for treating patients, new input mechanisms for measuring student interactions needed to be engineered and incorporated into a TUI, as defined in **Section 2.4**.

Previous works developing mixed cognitive-psychomotor training systems for healthcare using TUIs were developed for specialized PTTs including MIS (Computer-Enhanced Laparoscopic Training System, or "CELTS", Stylopoulos et al, [123]), female pelvic exam (E-Pelvis, Pugh et al, [124, 125]), and the clinical breast exam (CBE Simulator, Kotranza et al, [64]). CELTS added to the field of MIS training by providing a quantified model for scoring trainees that used CELTS against an expert model, was able to give the trainee feedback about what aspects of their performance could be improved to improve their score, and also distinguish between expert and novice performers [123]. The E-Pelvis added to the field of TUIs in healthcare simulation by incorporating sensors for measuring a user's applied force to internal anatomical structures during a pelvic exam and by providing quantified feedback in real-time about the user's performance [124]. The initial concepts of the E-Pelvis were combined with a mixed reality system for the

clinical breast exam. The system is defined as mixed reality because it merges interactions with a virtual patient with a physical model of a breast that the user palpates to detect lumps. The system can provide feedback to the user based upon the total coverage and adequate palpation style during examination of the breast model [63, 65]. These advanced PTTs are early examples of TUIs within healthcare that provide feedback to the user based upon measured performance and can evaluate the user's performance compared to an expert model. However, none of these systems demonstrate the ability to provide real-time automated instruction based upon those measurements. BodyExplorer builds upon previous work in the field of TUIs for healthcare simulation by incorporating real-time automated instruction with quantified feedback.

6.2 IMPLEMENTATION OF AUTOMATED INSTRUCTION

Within the BodyExplorer system architecture previously defined in **Section 3.3.2.1**, the software for automated instruction is built upon a finite state machine (FSM). The FSM stipulates that the number of states in the machine is finite and that state changes, which are called the "transitions," lead to other existing states in the machine [126]. Leveraging open-source software, the implementation of the FSM in BodyExplorer builds upon the Stateless library [127].

The Stateless library was chosen based upon its minimalist design and on its ability to seamlessly integrate within Unity. **Figure 89** can be used to visualize the following description of state machines created using Stateless. Using Stateless, a state machine contains a list of states and a list of triggers. The list of states specifies the finite number of states within the machine, and the list of triggers specifies the possible transitions to other states. In **Figure 89**, there are two possible

states (Cricoid Instructions and Detect Cricoid Touch) and two possible triggers (Success and Provide Help).

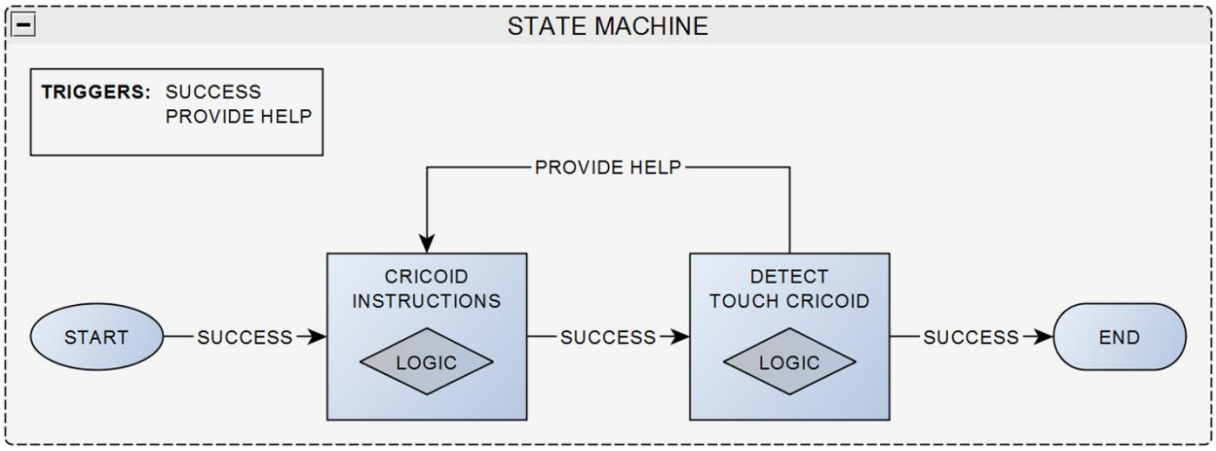


Figure 89. State machine example
(© 2016, Douglas A. Nelson Jr.)

Each state contains its own logic for performing internal functions and for deciding when to transition to another state. States transition to other existing states by "firing" a trigger. When states are initially defined within the state machine, the definitions for each state's transitions are specified. **Figure 89** shows the state machine with defined triggers between states. The Cricoid Instructions state only fires the trigger Success, which transitions to the Detect Cricoid Touch state. The Detect Cricoid Touch state can fire either the trigger Success, which ends this example, or fires the trigger Provide Help, which transitions back to the Cricoid Instruction state.

States can also perform internal functions based upon the trigger that was used to enter the state. For example, initially, Cricoid Instruction receives the Success trigger. The state's logic based upon receiving the Success trigger could be a general set of instructions telling the user how to locate and press on the cricoid cartilage. However, whenever the Cricoid Instruction state receives the Provide Help trigger, the state's logic tells the user a different set of instructions and

shows the user where to push by projecting instructional guidance onto the mannequin's neck, as shown in **Figure 90**.

Figure 90. Instructional guidance on mannequin
(© 2016, Douglas A. Nelson Jr.)

Furthermore, the triggers can be used to pass additional data between states during transitions. The implementation of triggers contains the name (e.g. Success) and also a generic object-type variable for passing data between states. A specific example of passing data with respect to the example state machine in **Figure 89** could be that the Success transition from Cricoid Instructions state could pass a boolean value (either true or false) to the Detect Cricoid Touch based upon whether or not the Cricoid Instruction state just provided the initial instructions (e.g. the state was entered by the Success trigger) or just provided the help instructions (e.g. the state was entered by the Provide Help trigger). In both cases, the Detect Cricoid Touch state is entered by a transition from Cricoid Instructions and the Success trigger. However, the logic within the Detect Cricoid Touch state could vary based upon the boolean parameter passed to the state

alongside the Success trigger. This functionality enables the general logic for detecting when the user touches the cricoid cartilage within the Detect Cricoid Touch state to be reused, while the only difference in the logic is whether the instructional guidance (as shown in **Figure 90**) remains projected on the mannequin or is removed.

In the previous example shown in **Figure 89**, the two states each had a different theme. The Cricoid Instructions state primarily instructs the user about what she/he should be doing through verbal and on-body instructions and guidance (Instruction States). The Detect Touch Cricoid state primarily waits for and measures user interactions on the body (Detect States). Building a series of Instruction and Detect States is how automated instruction within BodyExplorer is provided. Detect States can measure user actions from the Learning Module Sensor subsystems for application of cricoid pressure (Cricoid Pressure Learning Module), administration of medications (Medication Administration Learning Module), or endotracheal intubation (Endotracheal Intubation Learning Module).

Instruction States can provide instructions to users in multiple ways. As piloted in the Input Device Tutorial during UT3, verbal instructions and on-body targets can be provided to users. A script was created for the verbal instructions, and digital audio files were composed from voice recordings of the script. Programmatic sequencing of the audio files provided the automated instruction in response to user actions.

On-body targets are provided using the projector to highlight areas where the user should be focusing her/his attention and/or touching. **Figure 91** shows the on-body targets for using the input device to make anatomy windows (**Figure 91, a**), locating the thyroid cartilage on the neck (**Figure 91, b**), and locating the cricoid cartilage on the neck (**Figure 91, c**). In the example images showing the thyroid and cricoid cartilage locations on the neck, alpha channel masking is used to

highlight only the region of interest (the neck) while also providing an overlay of where the underlying cartilage structures can be located on the neck.

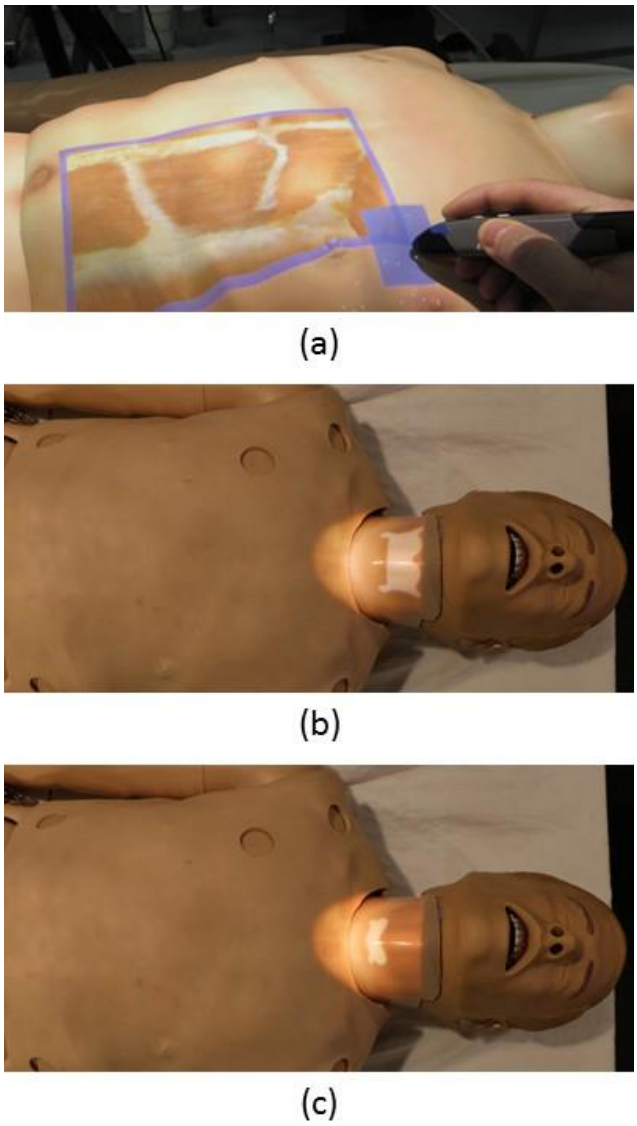


Figure 91. On-body targets: (a) for making anatomy windows with the input device, (b) for locating the thyroid cartilage, and (c) for locating the cricoid cartilage.

Based upon feedback received during the pilot of automated instruction during UT3, additional functionality for on-body guidance—specifically for providing demonstrations—was

incorporated into the automated instruction system. The on-body guidance implements an overlay system for showing recordings of instructor demonstrations on the mannequin. The recordings are made with a solid color black sheet draped over the mannequin (**Figure 92, a and b**). The black background in the video is removed by using Chroma keying in Adobe Premiere Pro (CS5, Adobe) as shown in **Figure 92, c**. The resulting video is alpha blended to show only the instructor's hand movements as opaque and the rest of the scene as transparent. Thus, the video can be overlaid onto the regularly projected anatomy to give the appearance of a virtual instructor being present with the user (**Figure 92, d**).

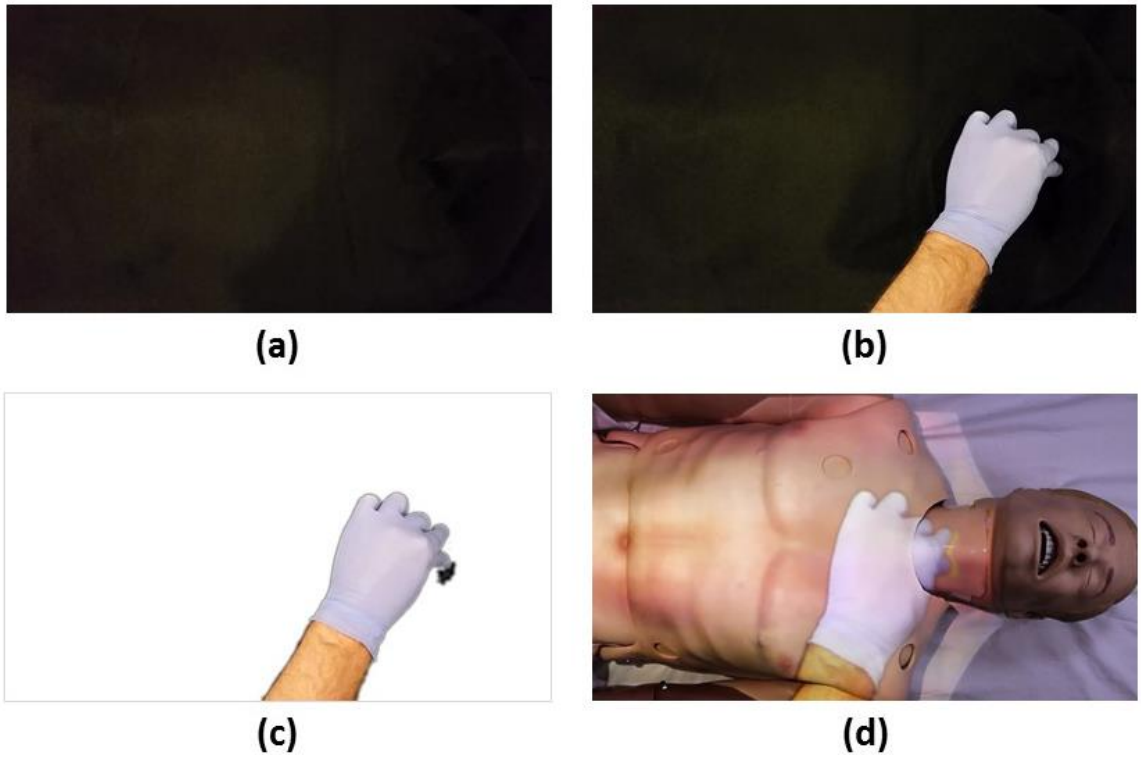


Figure 92. Creating on-body guidance demonstrations: (a) monochrome background draped over mannequin, (b) video recording is made from instructor performing actions over the mannequin, (c) black background is removed using chroma-keying, and (d) the resulting video frames are merged with the projected anatomy.

As shown in **Figure 93**, the on-body guidance from the virtual instructor shows the user where the cricoid cartilage is located on the neck (**Figure 93, a**) and how to position her/his fingers while applying cricoid pressure (**Figure 93, b and c**).

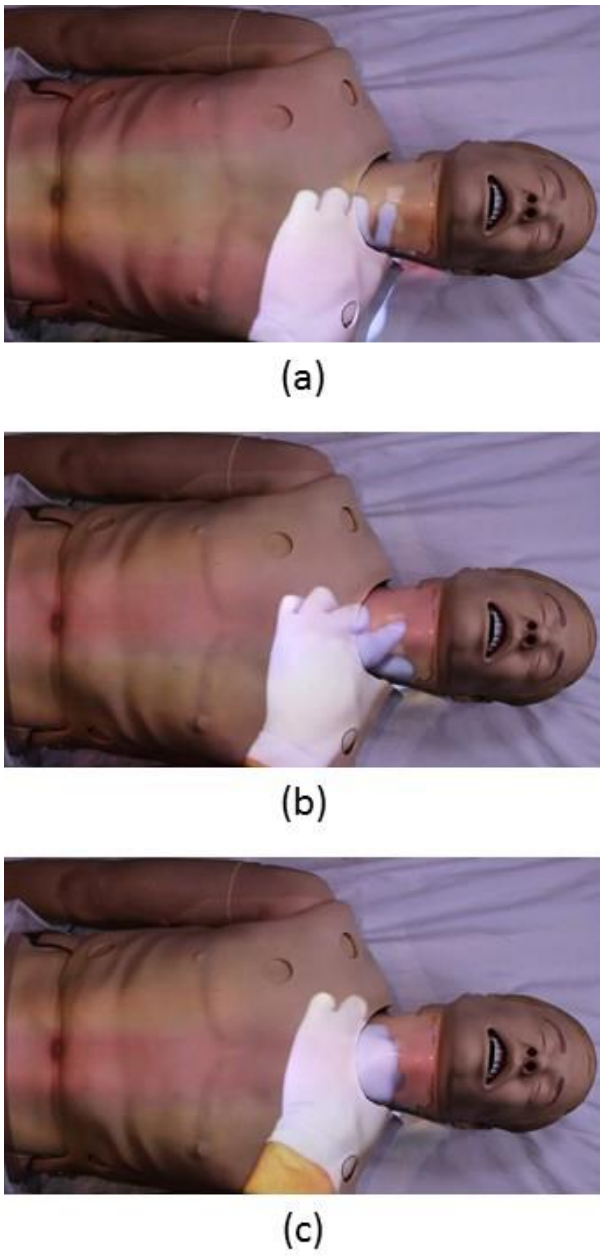


Figure 93. Instructional guidance on mannequin for applying cricoid pressure: (a) showing where to find the cricoid cartilage, (b) showing how and (c) where to position fingers to apply cricoid pressure.

Finally, the on-body guidance can remain visible after instruction to provide a visual reference for where and how the user should be positioning her/his hand. **Figure 94** shows where the user should position her/his hand. Once the user's hand is touching the correct area, the on-body guidance is removed.

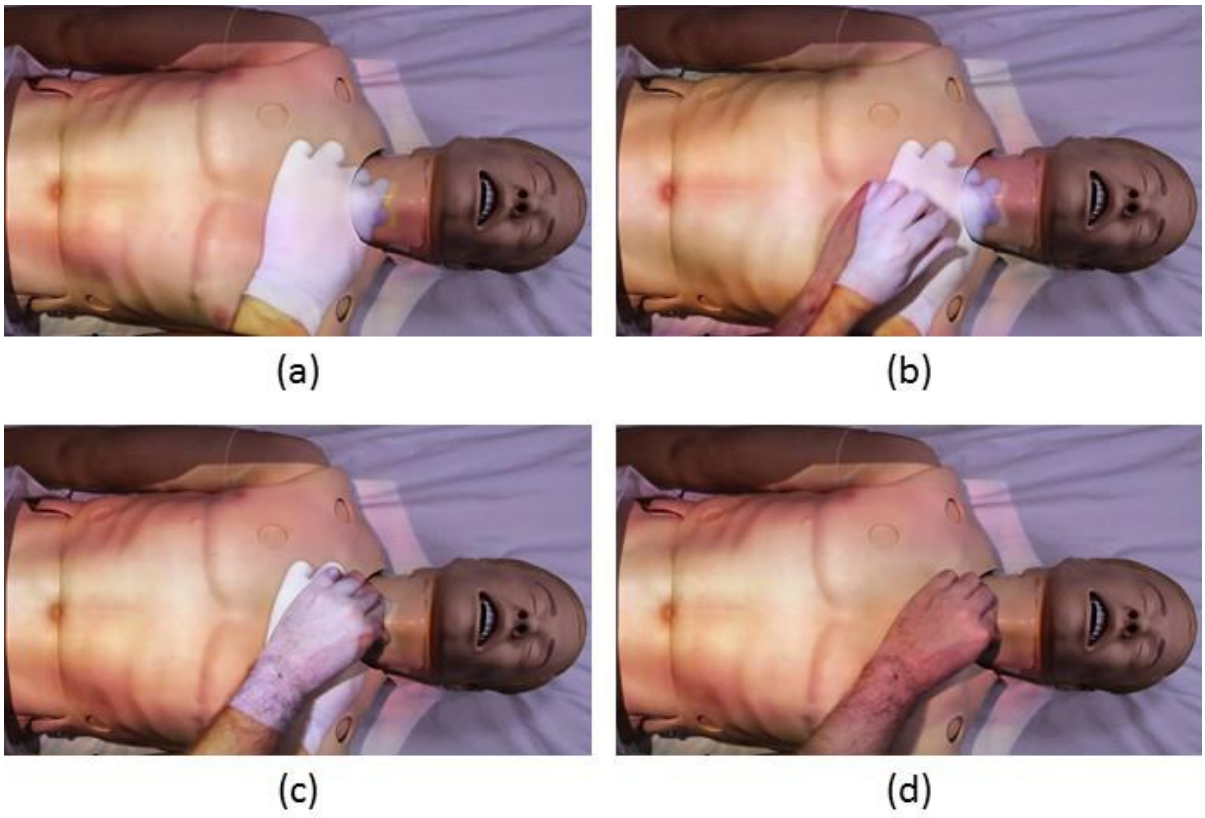


Figure 94. Instructional guidance on mannequin for user to match: (a) reference hand for where and how user should press, (b) user attempts to match reference hand, (c) user matches reference hand and applies cricoid pressure, (d) reference hand is removed.
(© 2016, Douglas A. Nelson Jr.)

Based upon feedback received during UT3, an additional on-screen visual feedback mechanism was provided during application of cricoid pressure. In addition to the color change of the cartilage structures and the animation of the esophagus getting pinched, a scale showing the range of applied pressure was shown to the user. **Figure 95** shows examples of the pressure range

scale with respect to not enough pressure (**Figure 95, a**), proper pressure (**Figure 95, b**), and too much pressure (**Figure 95, c**).

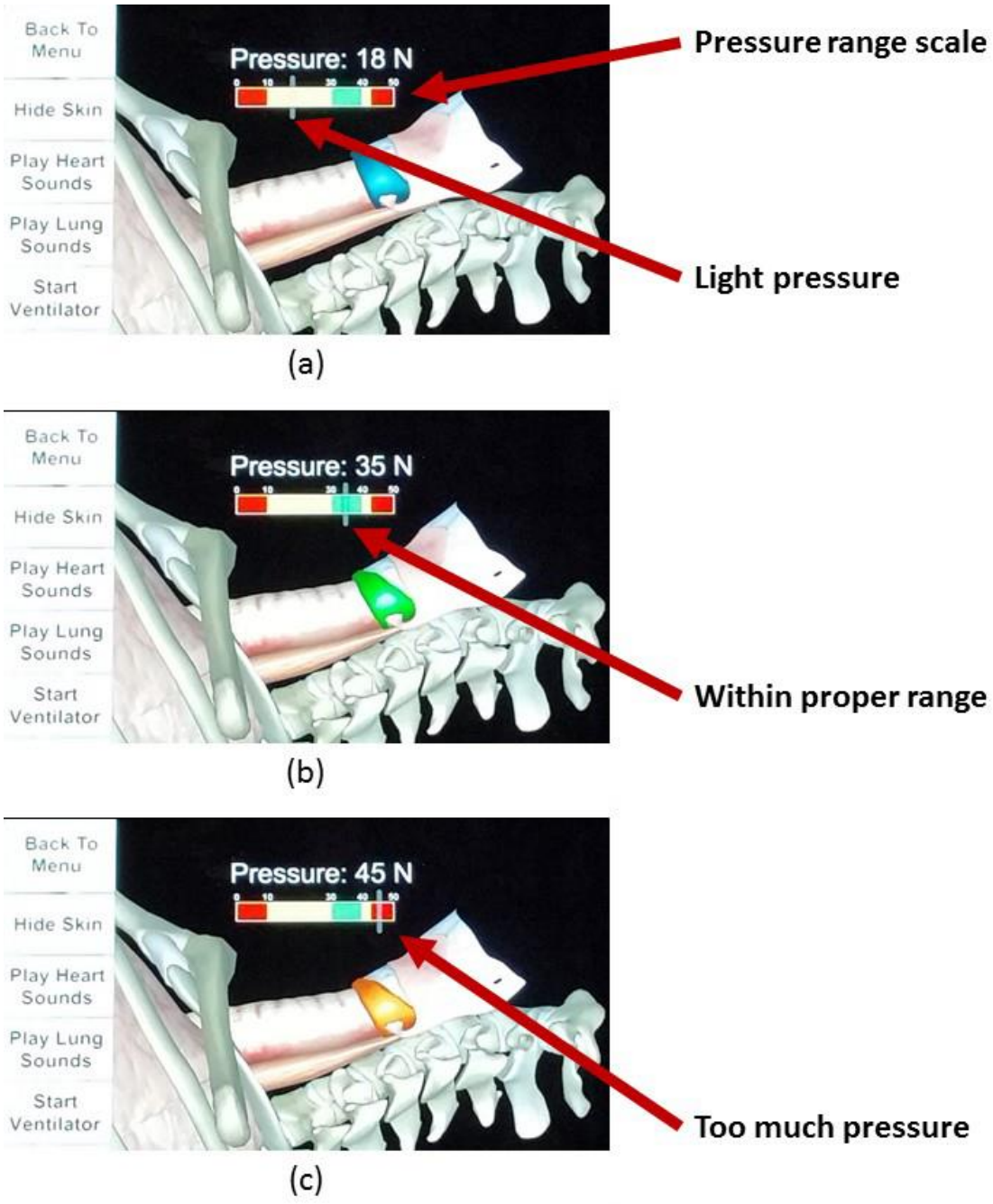


Figure 95. Cricoid pressure range scale showing applied pressure with respect to guidelines: (a) not enough pressure, (b) proper pressure, and (c) too much pressure
(© 2016, Douglas A. Nelson Jr.)

In addition to these methods for providing automated instruction, other on-screen and on-body methods were developed as part of the Medication Administration Learning Module. The on-screen methods are shown in **Figure 96**.

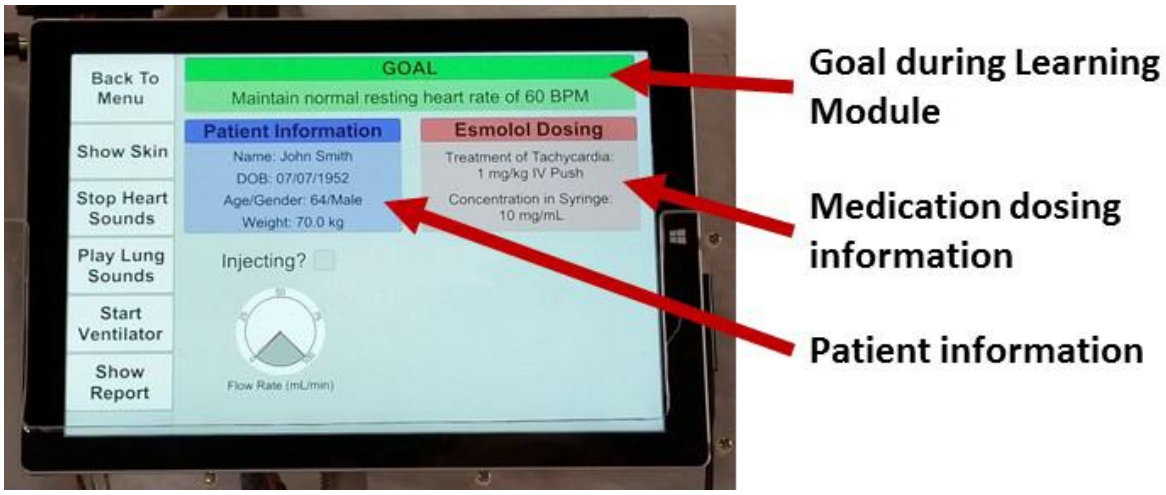


Figure 96. Auxiliary display presented during the Medication Administration Learning Module. (© 2016, Douglas A. Nelson Jr.)

A patient case is presented to the user, including the patient's name, date of birth, age, gender, and weight. Verbal instructions, as well as on-screen instructions, describe the goal of the scenario to lower the patient's heart rate to a normal resting heart rate by administering medication. In this case, the recommended medication is esmolol, and the dosing information is both verbalized by the virtual instructor and displayed on the auxiliary display. Furthermore, the virtual instructor automatically modifies the projected anatomy, removing the skin to show the beating heart and ECG (**Figure 97**).

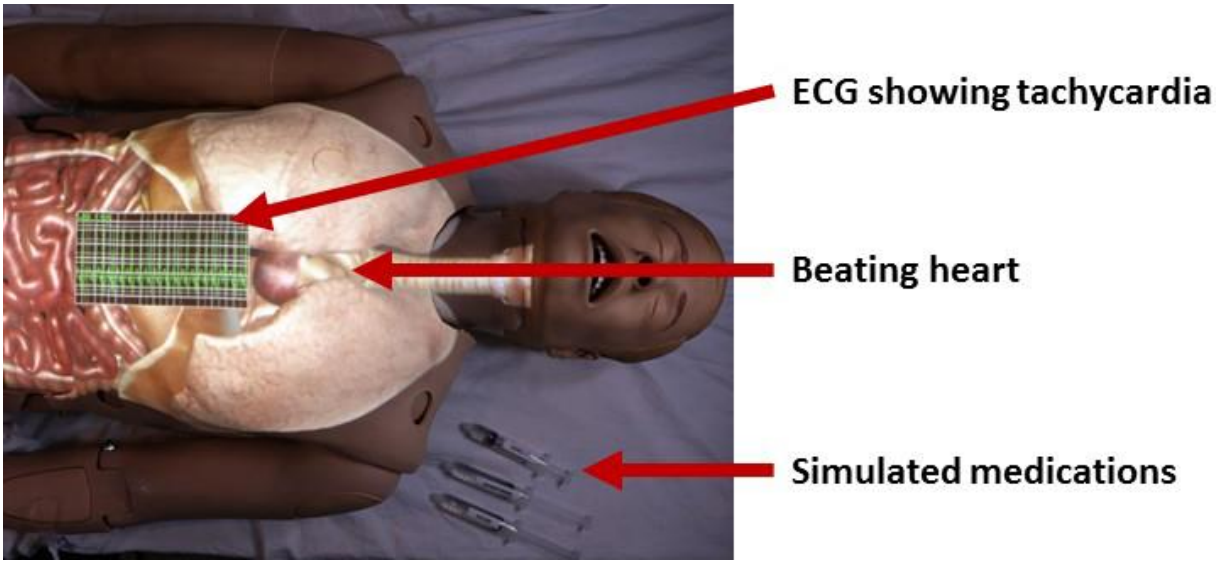


Figure 97. On-body projected anatomy during the Medication Administration Learning Module.
(© 2016, Douglas A. Nelson Jr.)

After given the instructions, the user must calculate and administer the correct dose to the patient. In this example, the 70-kg patient should receive 70 mg esmolol per the dosing instructions, which would equal 7.0 mL of the 10 mg/mL medication in the syringe. The automated instruction system detects when the user begins administering the medication, toggles the checkbox for "Injecting?" on the auxiliary display, and also displays feedback for the rate of injection (**Figure 98, top**). When the injection is completed, the auxiliary display shows the identity and volume of the administered medication (**Figure 98, bottom**).

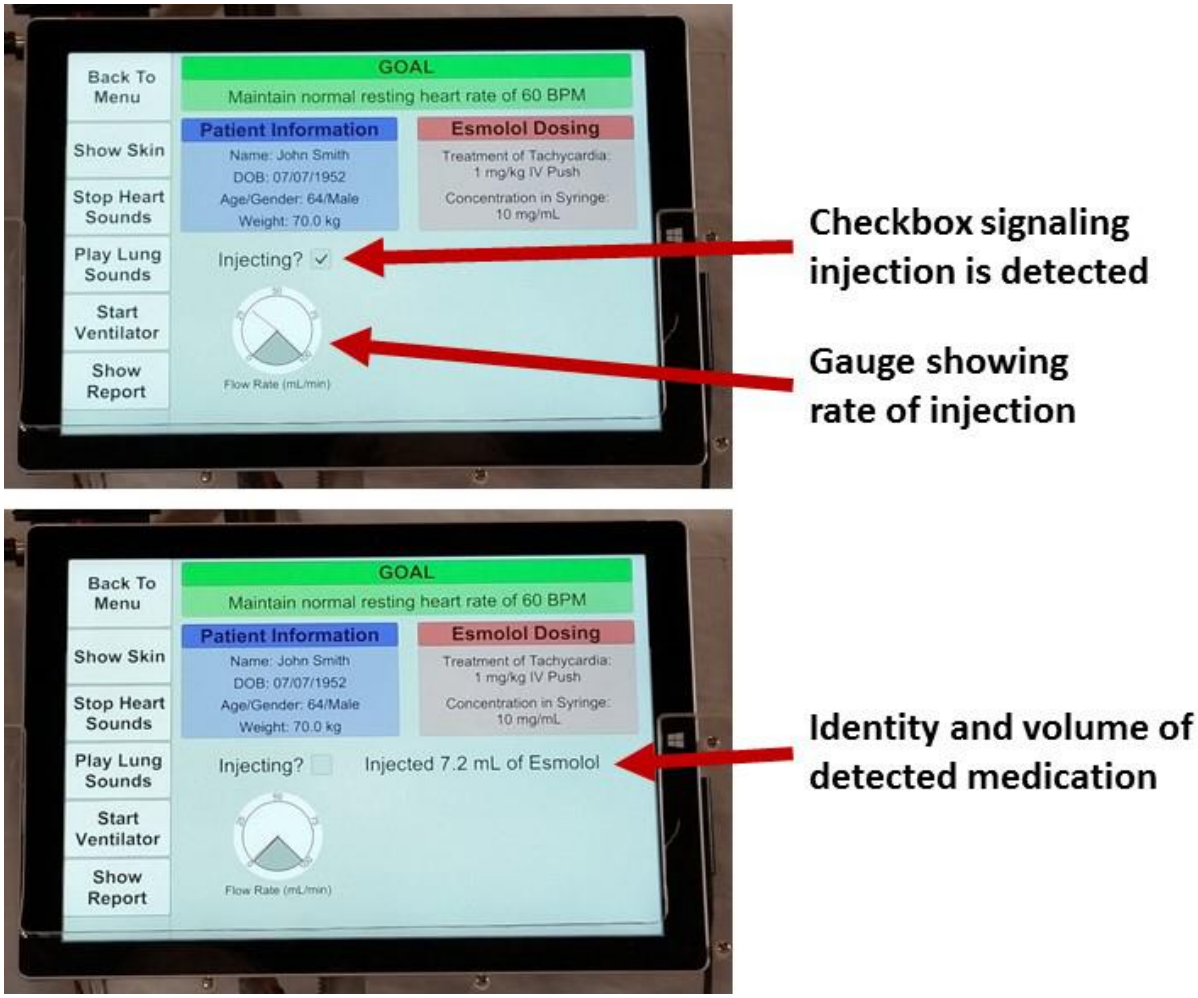


Figure 98. Updates to the auxiliary display during (top) and after (bottom) medication administration. (© 2016, Douglas A. Nelson Jr.)

After identifying the administered medication, the virtual instructor provides verbal feedback to the user. If the administered medication was esmolol, the virtual instructor provides verbal confirmation that esmolol was injected and tells the user that the physiological responses will be accelerated to show the effects of the injection. Then, the physiologic response of the patient is updated based upon the amount of esmolol injected. If the volume was less than 6 mL, the virtual instructor provides feedback that the volume was not enough, but that titrating is better than overdosing. Then, the user is instructed to administer more esmolol to reach the goal heart

rate of normal resting heart rate. If the volume was more than 8 mL, the virtual instructor provides feedback that the volume was too much and that overdosing could potentially cause severe bradycardia. At this point, depending upon the learning objectives for the module, the virtual instructor could instruct the user on how to correct for the overdose by administering a different medication. Furthermore, if the user administered a medication other than esmolol, the virtual instructor could show the physiological responses associated with the incorrectly administered medication.

In the example of the Medication Administration Learning Module, the physiology of the patient was being driven by the virtual instructor in response to the interventions performed by the user. The virtual instructor was also automatically changing the displayed anatomy, patient monitors, and status of the patient's physiology to teach the user about administering medications to return a patient's vitals towards normal values. Providing this functionality addresses user concerns related to automation, highlighted in **Section 3.1**. Much like real instructors modify the physiologic trends on the patient to meet learning objectives, the virtual instructor in BodyExplorer can accelerate physiologic responses and provide additional visualizations to the user to meet learning objectives.

6.3 APPLICATIONS TO OTHER HEALTHCARE TRAINING

The primary methods employed for automated instruction were verbal and visual. Verbal instructions were prerecorded audio files that were played through the speakers of the auxiliary display device. The instructions verbalized to the user included instructions about what action the

user should perform and how to do the action, positive feedback based upon correct user actions, and corrective error feedback based upon incorrect user actions.

Visual instructions were displayed to the user through on-body and on-screen methods. On-body visuals could be adapted for specialized methods of guiding users to targets (or anatomical landmarks) for placing her/his hands, touching with a medical tool such as a stethoscope, inserting a medical device such as a chest tube, or making an incision for inserting an emergency airway. These methods are also amenable to providing instruction on performing a physical exam and assessment of a patient, including auscultation, palpation for abnormal masses or pain sensitivity, or assessment of surface abnormalities such as wound care and cleaning. On-body positioning guidance could also be used for positioning other medical tools on the outside surface of the body such as positioning of a mask for ventilation, paddles or pads for defibrillation, ECG/EKG leads to monitoring, or even guiding the positioning and orientation of an ultrasound probe for sonography such as during prenatal, needle guidance, vascular, transthoracic, abdominal, or focused assessment with sonography for trauma (FAST) exams.

Generalization of the methods employed for measuring location and depth of insertion during endotracheal intubation could provide automated instruction for other domains where medical tools are inserted inside the body. Similar visualization techniques could be developed for instructing the insertion of other devices into the head and neck, including insertion of other airway devices, suction, or nasogastric tube insertion. Catheter interventions could also be demonstrated, such as placement of a central venous catheter or other intravenous or intra-arterial catheters such as pulmonary artery catheterization. Similarly, insertion of a Foley catheter could be demonstrated, while also incorporating a mechanism for measuring proper inflation of the catheter's cuff. While current PTTs exist for different procedures using scopes such as bronchoscopy, colonoscopy, or

other types of endoscopy, the methods employed in BodyExplorer could provide a method for automated instruction during team training, as well as providing auxiliary views of the overall organ anatomy with respect to the specific procedure being performed.

The methods for on-screen visuals were used to show the user a side-view of the neck anatomy relevant to applying cricoid pressure, visual feedback based upon if the user was touching the cricoid or thyroid cartilage, color-mapped feedback based upon how hard the user was pressing on the cricoid cartilage, and an anatomical animation correlating to the amount of cricoid pressure being applied that showed the esophagus being pinched between the cricoid cartilage and cervical spine when proper pressure was applied. These methods for on-screen visuals could be adapted for showing different viewpoints of anatomical views such as the brain or detailed views of the chambers of the heart, visual and/or color-mapped feedback based upon other measurable actions such as rate and depth of chest compressions during CPR, as well as other anatomical animations based upon a measured user action such as a side-view of a nasogastric tube being inserted through the nasal passage and down the esophagus into the stomach or a view of a Foley catheter traveling up the urethra to the bladder.

The on-screen visuals can also support traditional screen-based information related to medical devices and patient charting. As demonstrated previously by Samosky et al in [128] and leveraging networking functionality demonstrated in this dissertation with the auxiliary display, a suite of touchscreen-based devices could be networked within the BodyExplorer system architecture to incorporate automated instruction of peripheral medical devices with mannequin-based simulations.

6.4 LIMITATIONS OF BODYEXPLORER

BodyExplorer currently only measures interactions on, or inside, the mannequin, on the touchscreen of the auxiliary display, or proximity near the mannequin. From these measurements, the system can make inferences about what a user is doing on or to the mannequin. Initial work for detecting user presence has been prototyped, as described in **Section 5.4.1.2**. Expanding the measurement field to the area around the mannequin could expand the instructional possibilities by being able to provide guidance on movements before touching the simulated patient. Furthermore, by knowing the location and actions of learners around the mannequin, team training could be more individualized to each learner, instructing each learner where, when, and how they should move around the patient during a complex scenario. Possible technologies to expand the measurement field could include multiple cameras and/or depth sensors positioned around BodyExplorer to measure the 3D environment around the mannequin.

BodyExplorer is not currently designed to provide instruction for performing open surgeries. While the functionality for making anatomy windows on the mannequin does resemble the act of cutting through tissue layers, it does not provide any haptic feedback or depth-of-cut control based upon applied force of the instrument for cutting. Furthermore, since the mannequin's surface is not modified when making virtual cuts during windowing, the physical depth realized during open surgeries is lost. BodyExplorer is more amenable to training in minimally invasive surgeries than with open surgeries, but the haptic feedback of internal structures is still lacking with the current BodyExplorer system. A mixed-model of training that utilizes BodyExplorer and a pre-existing system for training minimally invasive surgeries that incorporates the haptics of internal anatomy could provide opportunities to expand team training in surgeries. The BodyExplorer system architecture aims to support modularity and expandability, and

incorporation of an existing haptic interface could be a good test of the modular nature of the architecture.

7.0 GENERAL DISCUSSION AND FUTURE DIRECTIONS

7.1 GENERAL DISCUSSION

7.1.1 Summary

The work presented in this dissertation describes the development of a novel simulator for innovations in healthcare education. Design objectives for the simulator were developed using principles of UCD by including end-users at all stages of development, from eliciting design requirements to end-user validation during usability testing. The system incorporates methods for measuring user actions while performing common medical procedures with visualizations of anatomy and physiology in response to those actions. Furthermore, automated instruction connects the measured actions and visualizations to support self-learning. The two most significant contributions of this work are:

1) the developments related to automated instruction for supporting self-learning, and
2) the modular and extensible architecture for permitting continued innovations to build upon the initial developments presented herein.

Usability testing sessions gathered feedback from end-users in order to guide iterative developments of the system. Qualitative feedback from the users provided an overview of priorities for future instructional content development. Quantitative metrics of system usability were developed, measured, and assessed to understand if improvements made the system easier to use. The results indicate that the software redesign at UT3 was easier to use than that previously

tested in UT1 and UT2. However, future research should evaluate usability of the input device independently.

Initial demonstrations of Learning Modules for applying cricoid pressure and administering medications were developed, incorporating feedback received from users on the Input Device Tutorial piloted in UT3. Verbal and visual methods were developed for providing automated instruction to users through a virtual instructor. The virtual instructor provided the automated instruction by talking to the user via the computer speakers based upon learning objectives and with respect to the actions of the user. The virtual instructor also provided on-screen and on-body visualizations as part of the automated instruction.

On-screen instructions provided quantified feedback based upon measured user actions while applying cricoid pressure and administering medications. Furthermore, on-screen visuals provided users with additional views of anatomy, such as a side view of the neck anatomy when applying cricoid pressure. The additional views of anatomy were synchronized with measured user actions driving animations of the anatomy, giving the user a visual representation of the internal consequences of their external actions.

The virtual instructor also provided on-body visualizations as part of the automated instruction. Based upon feedback received in UT3, demonstrations were included as on-body visualizations. The virtual instructor demonstrated where to touch on the neck when locating the cricoid cartilage. On-body visuals could also remain visible, providing a template for where and how a user should touch. These examples extended initial template matching that was prototyped for manipulating a laryngoscope by Nelson and Samosky [129].

The virtual instructor also demonstrated functionality for automatically manipulating the anatomy and physiology presented to the user based upon measured user actions in order to meet

learning objectives. In the Medication Administration Learning Module, the virtual instructor took control of the projected anatomy and changed the view automatically to show the user a view of the beating heart and ECG waveform during tachycardia. Then, the virtual instructor provided instructions for how to intervene using medication to lower the patient's heart rate towards normal values. Similar to how an actual instructor might modify the simulation from true reality to meet specific learning objectives, the virtual instructor can also adjust physiological responses to meet learning objectives. Instead of running a standard physiological response to the administered medication, the virtual instructor fast forwarded the response after instructing the user about the actual time response to expect when administering the medication.

These methods for providing automated instruction from the virtual instructor can be extended to other training applications; not just within interventions on patients, as outlined in **Section 6.3**, but also to learning how to use a variety of medical devices. Preliminary work has been presented by Samosky et al on virtualized medical devices [128]. Integrating training for proper use of medical devices within the BodyExplorer system architecture could teach users how to properly use the devices, as well as instruct users on how to identify and correct adverse events using the devices in the context of treating a simulated, physical patient.

7.1.2 Taxonomy for Providing Automated Instruction with BodyExplorer

As highlighted earlier in **Sections 6.3** and **6.4**, the methods employed for providing automated instruction with BodyExplorer have application to other healthcare training scenarios. While only a subset of specific healthcare training topics has been currently prototyped in this dissertation, there are a wealth of other topics amenable to the methods presented herein. **Figure 99** lists other possible healthcare training topics that can be developed using the taxonomy presented for

automated instruction with BodyExplorer. The green boxes highlight the features and applications currently implemented in this dissertation—a small subset of the possibilities! The blue boxes highlight how BodyExplorer could, with minimal effort, be extended to provide automated instruction for other healthcare training scenarios. The orange boxes highlight areas where moderate effort would be necessary to extend BodyExplorer. The primary obstacles that differentiate between implementing those in blue and those in orange are:

1. New sensor system(s) must be developed to measure the action or task of interest,
2. User action(s) with the device or during the procedure are complex and/or may be difficult to measure, and
3. New 3D CGI models of anatomy, physiology, and/or the device need to be created.

Based upon the taxonomy matrix in **Figure 99** and feedback received through usability testing, minimal effort would be needed to develop Learning Modules for expanding training scenarios for administration of IV medications, as well as developing new Learning Modules for patient assessment skills such as auscultation and patient monitoring equipment.

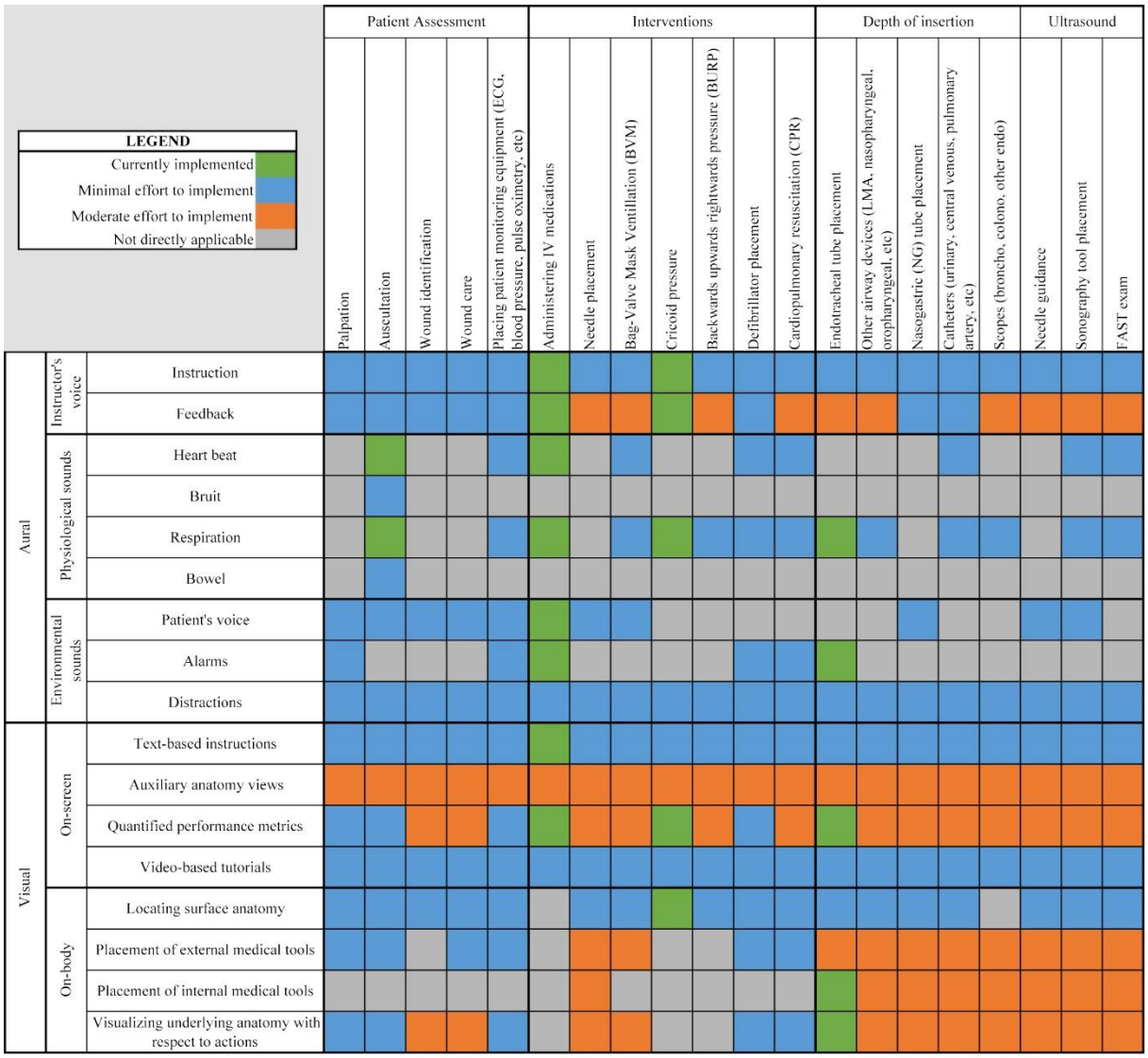


Figure 99. Examples of healthcare training scenarios amenable to automated instruction development within the BodyExplorer taxonomy
(© 2017, Douglas A. Nelson Jr.)

7.1.3 Extending the Taxonomy

The taxonomy presented in **Figure 99** has some key omissions that should be addressed in future work. First, the taxonomy only addresses two of the five senses: aural and visual. Additional work

could extend the taxonomy to include implementations for haptic and olfactory, but maybe not gustatory! Some preliminary work was performed as part of this dissertation towards developing a haptic model for palpating the thyroid and cricoid cartilage of the neck. This work could be extended for palpating other anatomical regions such as the abdomen for masses or sensitivities. Olfactory could be added through a Learning Module for caring for an infected wound or in a massive-hemorrhage scenario. Rather than requiring oozing, bloody wounds, the trauma could be simulated through projection with added realism—and visceral buy-in from the learners—by adding scents during the scenario.

Another key omission from the taxonomy that should be addressed in future work is the extension to team training. The columns from the taxonomy afford themselves to be combined for advanced team training applications. For example, a next step in BodyExplorer development should be to combine Learning Modules for IV medication administration, cricoid pressure application, and endotracheal intubation into a complex, team-training Learning Module.

Finally, BodyExplorer's taxonomy does not provide a method for verbal ***input*** from the user into the system. Verbal input could provide an opportunity for questions and answers between BodyExplorer's virtual instructor and the user(s). Extending the taxonomy to include verbal input could also provide a means for assessing team-based performance by developing a measurement of effective communication between teammates.

7.2 FUTURE DEVELOPMENT

While the work presented in this dissertation is a significant achievement, there are a plethora of opportunities for continued research and development to build upon this work.

7.2.1 Instructional Content Development

Based on feedback received during the three rounds of usability testing, additional instructional content can be developed. Content areas identified by users include:

1. comparisons between normal and abnormal anatomy and physiology,
2. addition of other patient vital signs including blood pressure, oxygen saturation, and capnography,
3. additional views of anatomy, including cross-sectional views inside organs, and
4. expanding the amount of physiologic responses to administered medications.

Items (1) and (3) will require additional artistic work to create 3D models of the abnormal and cross-sectional anatomy. Items (2) and (4) could be addressed by incorporating a comprehensive, open-source physiology engine into the BodyExplorer system architecture, such as BioGears [130]. Although, preliminary work on expanding the physiological responses to administered medications could begin before integrating a full physiology engine by scripting the physiological responses based upon guidance from educators and learning objectives.

7.2.2 Projection and Input Device Design

The quantitative results from the usability testing of the input device design were mixed. However, qualitative data from the usability testing recommend modifying the device design to reduce shadows, improve button interactions, and improve ergonomics while using the device. Since the development of BodyExplorer began, new interactive projectors have become commercially available. The Dell Interactive Short-Throw Projector model S320wi, previously shown in **Figure 82**, has built-in functionality for using an input device to interact with the projected images through

left- and right-click buttons, in addition to a programmable third button [131]. Furthermore, since the projector is short-throw, it can project a much larger field in a shorter distance. While the current projector is limited to projecting only onto the mannequin's torso when mounted 1m above the torso, the S320wi projector could project onto the entire mannequin's body, in addition to the area around the mannequin. Having a larger projection field could permit the projected images to follow the mannequin if the mannequin is moved without needing to move the projector. Additionally, projecting onto an area adjacent to the mannequin has potential for displaying other data such as:

1) blood or other bodily fluids that are not amenable to quick cleanup and turnaround using current simulators,
2) additional user-specific instructions, similar to having multiple auxiliary displays, and
3) directions for where users should move around the patient during team training scenarios.

7.2.3 Sensor Subsystem Improvements and Additions

Initial work presented in this dissertation assesses the accuracy of the Sensor subsystems for measuring touch and applied force on the thyroid and cricoid cartilage of the neck, in addition to assessing the accuracy of detecting the identity and volume of administered medication simulants. Future work could build upon the findings for detecting administered medications by optimizing the detection windows and the detection algorithm. Performing the optimization could expand the number of simulated medications recognizable by the system. Furthermore, addition of new Sensor subsystems could test the modular aspects of the BodyExplorer system architecture while also adding functionality for instruction of other medical procedures, as outlined in **Section 6.3** and in

the taxonomy presented in **Section 7.1.2**. For example, adding functionality for detecting placement of a stethoscope for instruction on auscultation could meet the needs identified by users and also be a straightforward technological implementation.

7.2.4 Automated Instruction Module Development

In its current implementation, modifications to the automated instruction system within the BodyExplorer system architecture must be accomplished through edits to procedural code in C#. It should not be expected that a clinical educator is well versed in computer programming to make edits and extend the automated instructions. Future work should address this limitation by providing an easy-to-use interface for allowing non-programmers to create new Learning Modules within BodyExplorer.

7.2.5 Team-Based Automated Instruction without a Team

Treating a patient often requires a team of clinical providers to be present. For example, during a RSI, at least three clinical providers must be present: one to administer medications, one to hold cricoid pressure, and one to intubate the patient. In training scenarios where less than three clinical providers are present, it may not be possible to practice a lifelike patient care scenario. In order to overcome this barrier to practicing steps of a team-based scenario when a full team is not able to be present, a virtual teammate could be added to BodyExplorer's architecture, similar to how one's ally in a video game might be controlled by an autonomous intelligent agent within the game.

Continuing the previous example, if two clinical providers are trying to practice RSI, the virtual teammate could virtually administer the medications while the two clinical providers apply

cricoid pressure and intubate BodyExplorer. The scenario could work, albeit with less physical realism, with only one clinical provider. For example, the single provider could take the role of holding cricoid pressure while virtual teammates administer medications and intubate. While the virtual teammates could signal the physiological changes through virtual administration of medications and virtual intubation, the physical components would be lacking—a virtual teammate will not be able to physically place the endotracheal tube, and the physical clinician will not feel resistance on the neck while holding cricoid pressure because there will not be someone physically trying to intubate BodyExplorer.

These physical limitations to virtual teammates could be overcome with robotics. Continuing the RSI example, actuators could be incorporated within the neck or outside the neck as a physical overlay (similar to the load testing apparatus from **Section 4.1.1**, only including active actuators for applying the loads). With one clinician, the virtual teammate could administer the medications, the robotic teammate could hold cricoid pressure, and the clinician could perform the intubation. Significant technological and architectural changes would need to occur to BodyExplorer for this virtual-robotic teammate scenario to come to fruition, but it could provide a method for team-based automated instruction when a full team is not present. As robotic surgeries are emerging, perhaps this idea is not too far off.

7.2.6 BodyExplorer System Architecture

A significant contribution of this dissertation is the creation of the BodyExplorer system architecture which allows rapid innovations in SMBE. Future work should assess usability of the architecture by others who did not contribute to the initial development of the architecture.

Additional work could also explore packaging the architecture into a distributable software library available to other researchers.

7.2.7 Evaluation of Learning using BodyExplorer

While it was not a primary objective within this dissertation, the eventual goal of BodyExplorer is to efficiently support learning, both in the development of learning materials and deployment of simulation, as well as teaching users how to provide safe healthcare to patients. Future work should design Learning Modules to supplement current curriculum and supplant costly or high-risk training models. Assessment of BodyExplorer should first show how the system can replicate current curriculum while achieving comparable academic outcomes. In addition, a cost-benefit analysis could evaluate the efficiency of using BodyExplorer in terms of material and time costs associated with lab resources, simulation preparation time and setup, and instructor and technician time to run simulation scenarios.

7.2.8 Oscilloscope for Informing Best Practices

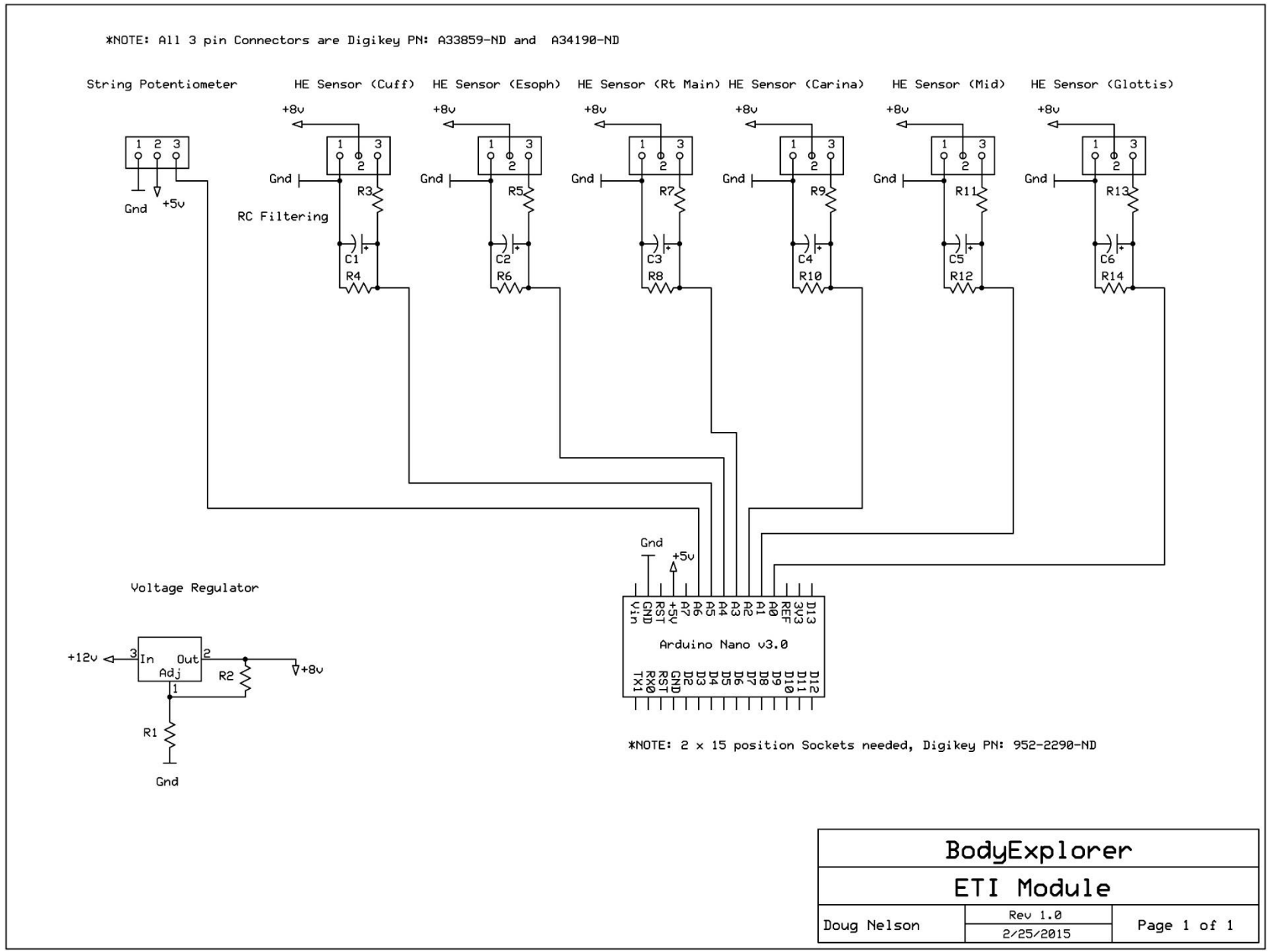
In electronics, a device called an oscilloscope is used to troubleshoot electronic signals, allowing the user to measure what cannot be seen by the human eye. In healthcare, medical errors can be observed, but the root cause of the errors may not always be apparent. BodyExplorer is a system that can measure user actions while treating a simulated patient. Some measurements are not readily visible to the human eye, such as rate of medication injection, how much force is being applied, or whether devices inserted inside the body are properly placed. Using BodyExplorer as

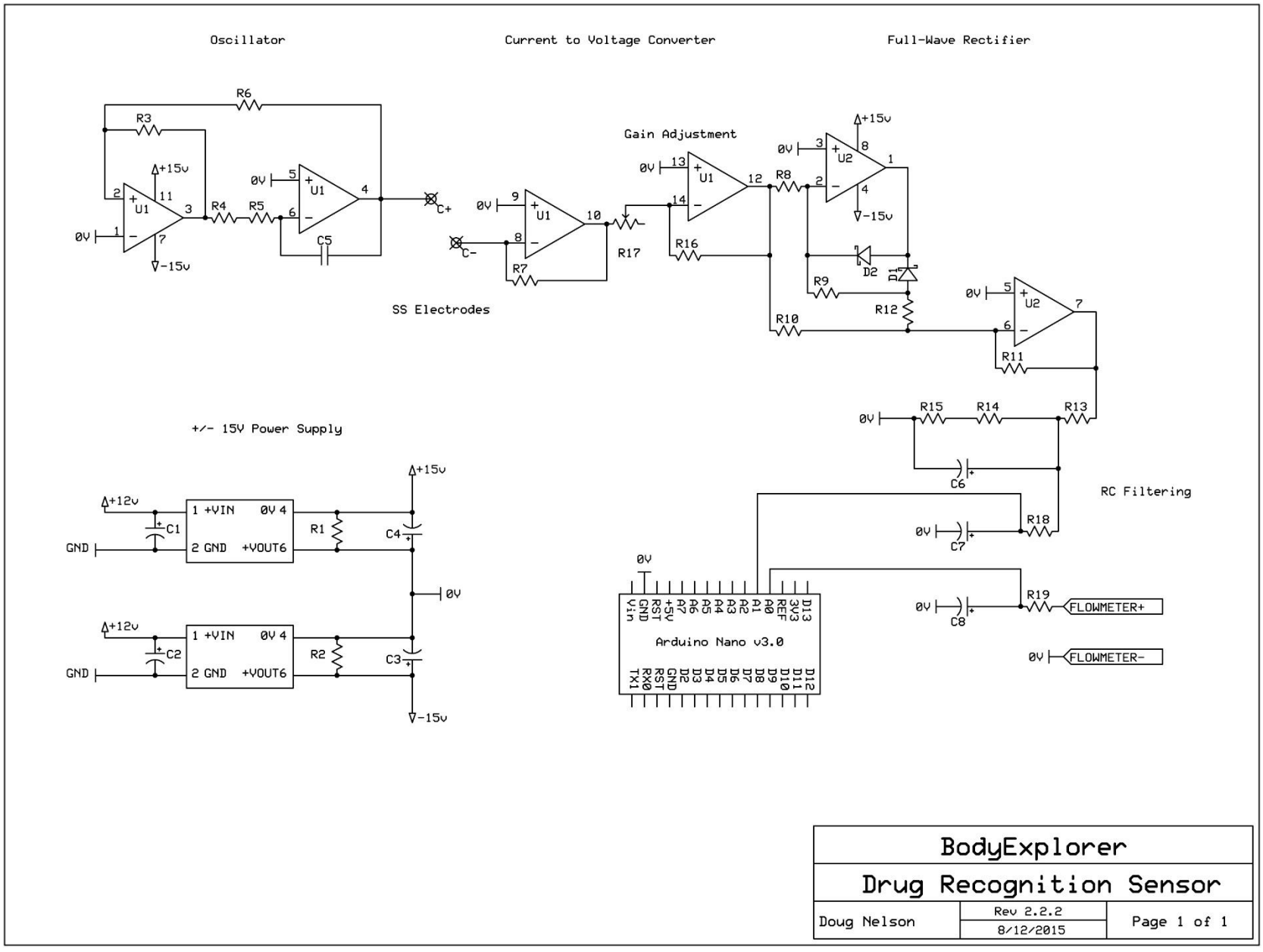
an oscilloscope for measuring the non-visible actions while treating a simulated patient can support:

1) measurement of standards for expert actions,
2) identification of competency and variability among experts, and
3) development of mastery standards and best practices in procedural skills using a data driven approach.

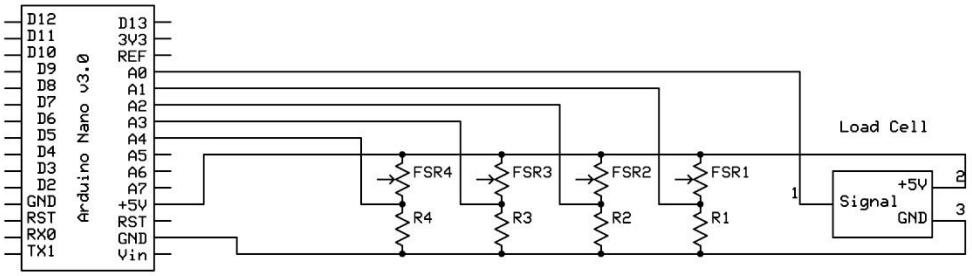
APPENDIX A

CIRCUIT DIAGRAMS


*NOTE: All 3 pin Connectors are Digikey PN: A33859-ND and A34190-ND
String Potentiometer
HE Sensor (Cuff)
HE Sensor (Esoph)
HE Sensor (Rt Main)
HE Sensor (Carina)
HE Sensor (Mid)
HE Sensor (Glottis)
+8v
Gnd
+5v
RC Filtering
R3
R4
C1
R5
R6
C2
R7
R8
C3
R9
R10
C4
R11
R12
C5
R13
R14
C6
Voltage Regulator
+12v
In
Out
Adj
R1
R2
Arduino Nano v3.0
*NOTE: 2 x 15 position Sockets needed, Digikey PN: 952-2290-ND
BodyExplorer
ETI Module
Doug Nelson
Rev 1.0
2/25/2015
Page 1 of 1


Oscillator
Current to Voltage Converter
Full-Wave Rectifier
Gain Adjustment
SS Electrodes
+/- 15V Power Supply
RC Filtering
R6
R3
R4
R5
R7
R8
R9
R10
R11
R12
R13
R14
R15
R16
R17
R18
R19
C1
C2
C3
C4
C5
C6
C7
C8
D1
D2
U1
U2
+15v
-15v
0V
C+
C-
+12v
GND
1 +VIN
2 GND
0V 4
+VOUT6
Arduino Nano v3.0
FLOWMETER+
FLOWMETER-
BodyExplorer
Drug Recognition Sensor
Doug Nelson
Rev 2.2.2
8/12/2015
Page 1 of 1

*NOTE: 2 pin Connectors for FSRs are Digikey PN: A33857-ND and A34187-ND



*NOTE: 2 x 15 position Sockets needed, Digikey PN: 952-2290-ND

*NOTE: 3 pin Connectors for Load Cell are Digikey PN: A33859-ND and A34190-ND

SMT R&D Center		
Cricoid Pressure Schematic		
Doug Nelson	Rev 3.0 3/16/2015	1 of 1

APPENDIX B

RATIONALE FOR ASSUMPTIONS IN FACTORIAL EXPERIMENT

Temperature corrections to the density of water and mass corrections due to the added salt in drug simulants were not performed for multiple reasons. First, the density of water deviates between 0.99907 and 0.99669 g/mL over a range of ambient, climate-controlled building temperatures (60-80 F)—the range of possible temperatures for the salt solutions during testing. Assuming the extremes, if an over-injection of 10.0 mL was performed during a trial, the correction for density would be 0.0331 g, which is outside the measurement resolution of the scale. Since the salt solutions are all at the same ambient temperature, the correction would affect all solutions equally. Mass correction due to the added salt in the solutions is also negligible. Again, assuming the extremes, if an over-injection of 10.0 mL was performed during an injection of the 3.0 g/L solution, the mass correction for the injection would be 0.03 g. Adding both of these scenarios, an overestimation of 0.0631 g would be expected at measurements from a 10.0 mL injection, which is at the rounding threshold for the scale. Thus, mass measurements could be inflated by as much as 0.1 g for 10.0 mL injections—a +1% error.

The reported accuracy of the flow meter is ±1%. Adding these contributions of error, we might assume a maximal error of +2% (+1% over-estimation on mass and -1% under-estimation on flow rate).

Mass correction would be more complicated by the fact that exact input mass from the salt in the solution at the site of injection is not exact output mass from the salt in the solution at the

measurement scale. The length of tubing between the site of injection and the point of measurement on the scale provides a delay before the input mass from the salt reaches the scale.

While it would be possible to account for all of these variables, the corrections would be negligible due to the measurement resolution of the scale, as discussed.

APPENDIX C

USABILITY TESTING PROTOCOLS

BodyExplorer Usability Testing: Demographics Tool BE1

Name:_____ **Email**:_____

Please mark your current status (mark all that apply):

- ❑ Undergraduate Student (expected graduation year:_____)
- ❑ Graduate Student (expected graduation year:_____)
- ❑ RN (years experience:_____)
- ❑ Didactic Instructor (years experience:_____)
- ❑ Clinical Instructor (years experience:_____)

What is your primary clinical and/or educational role? _____

Have you previously… (mark all that apply)

	In: Simulation	Humans	Comment(s)
...administered oral medications (PO)			
...administered IV medications (IV push)			
...performed bag-valve mask ventilation			
...performed intubation			
...performed cricoid pressure			

If you have previous experience with simulators, what kind have you used in the past? (mark all that apply)

- ❑ Screen-based systems (such as an online CPR course or a computer-based anatomy/physiology lab)
- ❑ Partial-task trainer (such as an IV insertion arm, intubation head/airway, etc)

Full-body mannequins:

- ❑ Static mannequins (such as a CPR mannequin or a plastic dummy)
- ❑ Computer controlled mannequin (such as SimMan)

Please mark whether you have had previous experience in an anatomy lab that was:

- ❑ Animal-based
- ❑ Cadaver-based
- ❑ Computer-based
- ❑ Mannequin-based

Have you previously used the BodyExplorer? YES NO

Are you interested in evaluating future BodyExplorer modules? YES NO

BodyExplorer Usability Testing - UT0 Tasks

I. Windowing and IV Drug Administration **DATE/TIME: _____**

Tasks:

A. Edit the Windows:

1. Enter "Edit Mode".
2. Open a window showing the muscle using the pen (aka Muscle Window).
3. Move the Muscle Window.
4. Resize the Muscle Window.
5. Move and Resize the Muscle Window to show only a small border of Skin around the Muscle Window.
6. Open a window showing the ribcage and abdominal organs (aka Ribcage Window).
7. Move and Resize the Ribcage Window to show only a small border of Muscle Window around the Ribcage Window.
8. Open a window showing the beating heart (aka Heart Window).
9. Move and Resize the Heart Window to show only a small border of Ribcage Window around the Heart Window.

TIME	NOTES

BodyExplorer Usability Testing Study: Questions to Participant

Example Open-Ended Questions:

I noticed… can you tell me more about that?
… how did that feel?
… how does that compare to a human?
that you were frustrated when … can you talk more about that?

Did you encounter anything else that was …
...frustrating?
...exciting?
...surprising?

Was the level of anatomy/physiology detail what you would have expected?

Were the responses that you saw when you injected the IV medications what you would have expected?

Did the response seem of the correct magnitude? timing? as quickly as you would have expected?

Tell me about .... feature? (IV drug, Intubation, Cricoid Pressure, Windowing)

What were you thinking when you opened that window?

Can you elaborate more?

That's really a valuable statement, can you talk more about that?

Tell me more about ____? (ask probing questions)

BodyExplorer Usability Testing - UT1 Tasks

I. Windowing and IV Drug Administration

Tasks:

A. Edit the Windows:

1. Enter "Edit Mode".
2. Open a window showing the muscle using the pen (aka Muscle Window).
3. Move the Muscle Window.
4. Resize the Muscle Window.
5. Move and Resize the Muscle Window to show only a small border of Skin around the Muscle Window.
6. Open a window showing the ribcage and abdominal organs (aka Ribcage Window).
7. Move and Resize the Ribcage Window to show only a small border of Muscle Window around the Ribcage Window.
8. Open a window showing the beating heart (aka Heart Window).
9. Move and Resize the Heart Window to show only a small border of Ribcage Window around the Heart Window.

B. Explore the Anatomy:

1. Enter "Explore Mode".
2. Point to different organs with the pen.
3. Click on different organs with the pen.

C. Explore Heart Physiology:

1. Click on the Heart.
2. Play heart sounds.
3. Show an ECG.

D. Explore Cardiac Pharmacology:

1. Administer 5cc of simulated Epinephrine into the IV port of the BodyExplorer.
2. Administer 5cc of simulated Esmolol into the IV port of the BodyExplorer.
3. Administer 5cc of simulated Saline into the IV port of the BodyExplorer ***at a rapid rate***.

BodyExplorer Usability Testing - UT1 Tasks

II. Rapid Sequence Induction

Tasks:

A. Put the BodyExplorer into Anesthetized State

1. Administer 16cc of simulated Propofol into the IV port of the BodyExplorer.
2. Administer 3cc of simulated SuccinylCholine into the IV port of the BodyExplorer.
3. Administer 4cc of simulated Saline into the IV port of the BodyExplorer.

B. Explore Application of Cricoid Pressure:

1. Find and press on the Thyroid Cartilage.
2. Find and press on the Cricoid Cartilage.

C. Intubate the BodyExplorer:

1. Insert the Endotracheal (ET) Tube into the Trachea using Direct Vision Laryngoscopy.
2. Inflate the ET tube's cuff with air using the syringe near the BodyExplorer's head.
3. If you were able to successfully intubate the patient on your first attempt, deflate the ET tube's cuff, remove the cuff, and purposely insert the ET tube into the BodyExplorer's esophagus.

**If you, the participant, have never intubated before, the moderator will demonstrate.

APPENDIX D

VIDEO CODING ANALYSIS

DEFINITIONS AND CODINGSCHEMA

DEFINITIONS:

“ANATOMY WINDOW” – an image projected onto the body that shows a layer of anatomy. The user can click on HOTSPOTS to move or resize an ANATOMY WINDOW

“HOTSPOTS” – a location within an ANATOMY WINDOW that, when clicked and dragged, allows a user to move or resize an ANATOMY WINDOW. The center of the ANATOMY WINDOW is a HOTSPOT that allows a user to move the ANATOMY WINDOW. The corners of the ANATOMY WINDOW are HOTSPOTS that allow a user to resize the ANATOMY WINDOW from that corner.

“INPUT DEVICE” – the device used by the user to interact with images projected onto the body. The user uses the INPUT DEVICE to open/move/resize ANATOMY WINDOWS, point to organs, and interact with ON BODY MENUS.

“MENU” – a collection of buttons or selectable objects displayed ON AUXILIARY DISPLAY or ON BODY Examples: The menu for the layers on the AUXILIARY DISPLAY (tablet) or ON BODY (projected on the mannequin). The menu that comes up ON BODY when the user clicks on an organ while in EXPLORE MODE.

“EDIT MODE” – the mode that allows the user to manipulate the ANATOMY WINDOWS, ie open/move/resize ANATOMY WINDOWS.

“EXPLORE MODE” – the mode that allows the user to point to organs with the INPUT DEVICE and see the name of the organ being pointed to. The user can also press the button on the INPUT DEVICE while pointing to an organ to bring up that organ’s ON BODY MENU.

CODING SCHEMA:

User/Using Input Device

The user had the INPUT DEVICE in his or her hand and was pointing the INPUT DEVICE at the mannequin with intent to manipulate the visuals on the mannequin. Times when the INPUT DEVICE was in the user's hand but not pointed at the mannequin should not be included under this code. Times when the INPUT DEVICE was in the user's hand and pointed at the mannequin but the user was not actively interacting with the visuals on the mannequin should not be included under this code.

User/Using Input Device/Using PEN-LIKE Input Device

The videos named with a "1" and "2" at the beginning should be coded as PEN-LIKE for all times while the user is "Using Input Device". The videos with a "3" at the beginning should be coded as PEN-LIKE for all times while the user is using the INPUT DEVICE that is identical to the INPUT DEVICE used in videos beginning with "1" and "2".

User/Using Input Device/Using WAND-LIKE Input Device

The videos named with a "3" at the beginning should be coded as WAND-LIKE for all times while the user is using the INPUT DEVICE that is different from the INPUT DEVICE used in videos beginning with "1" and "2". This INPUT DEVICE will either be a larger white device or a device with a long white tip.

*****Note: Difficulty with the INPUT DEVICE should not be confused with difficulty understanding the instructions from the task list. For example, it was noted that users didn't understand the instructions from the task list to "Move and resize the XXXXX window to show only a small border of YYYY window around the XXXX window."**

User/Difficulty with Input Device

The user was observed having difficulty with the INPUT DEVICE to open/move/resize an anatomy window. Examples could include:

1) An ANATOMY WINDOW disappeared without the user intending to remove it.
2) The user repeated the same action with the INPUT DEVICE multiple times because his or her intended response by the system was not observed. For example, multiple attempts to move or resize a window while doing the same action. While resizing, a user attempting to click in the same corner to resize multiple times in a row on the same corner can be classified as "Difficulty with Input Device". However, this is different than a user resizing a window from multiple corners to get the window to be the size that he or she desires. In this second example, the action should not be classified as "Difficulty with Input Device". When moving, a user attempting to click in the middle of the window and drag to a new location multiple times in a row without the window moving between attempts can be classified as "Difficulty with Input Device". However, this is different than a user moving a window from spot to spot in succession. In this second moving example, the action should not be classified as "Difficulty with Input Device".
3) The ANATOMY WINDOW "jumps" to the corner of the projected anatomy while the user is moving or resizing the window. This will be visible by the window "jumping" to the mannequin's lower right abdominal area.
4) The user clicking the button on the INPUT DEVICE multiple times while the device is in the same location. You should be able to hear the button "clicks" on the audio playback. This can be interpreted as the button not responding to the user.

User/Difficulty with Input Device/Verbalized Difficulty with Input Device
The user verbally stated that he or she was having difficulty using the INPUT DEVICE. This is a subset of the "Difficulty with Input Device" coding scheme.

Moderator/Providing Help
The moderator was PROVIDING HELP to the user ***specifically*** about using the INPUT DEVICE. Typically, this is associated with the moderator either providing instructions about how to properly use the pen, or when the moderator provides comments to the user after observing the user have DIFFICULTY with the INPUT DEVICE.

Moderator/Providing Help/Task List
The moderator was providing help to the user ***specifically*** about following the TASK LIST. Typically, this is associated with the moderator either providing instructions for the next step, or when the moderator provides comments to the user when the user doesn't understand the instructions.

Differentiating PROVIDING HELP versus TASK LIST:
PROVIDING HELP is differentiated from help about the TASK LIST by the following examples:

1) <PROVIDING HELP> "Keep the pen close to the body surface"
2) <TASK LIST> "Move the window and make it really big"
3) <PROVIDING HELP> "You don't need to touch the body surface with the INPUT DEVICE"
4) <TASK LIST> "Point to one of the corners of the window and click-and-drag to resize the window"

I can see how example (4) could be coded as both PROVIDING HELP and TASK LIST, but it depends upon the context. If the user had not yet moved on to the step about moving/resizing a window yet, and the moderator was prompting the user to move on to that step, then example (4) should be coded as TASK LIST. However, if the user already had window(s) open, and the moderator was noticing the user having difficulty with the INPUT DEVICE, and THEN the moderator told the user (4), then the example could be coded as PROVIDING HELP.

Generally, if the moderator is observing the user having difficulty using the INPUT DEVICE, the segment should be coded as PROVIDING HELP. If the moderator is observing the user having difficulty understanding the instructions, the segment should be coded as TASK LIST.

Interrater Reliability Results from Control Videos

Coding Comparison	Moderator			User						
	Providing Help			Using Pen-Like Device			Using Wand-Like Device	Difficulty with Input Device		
	UT1	UT2	UT3	UT1	UT2	UT3	UT3**	UT1	UT2	UT3
PI v Rater1	(98%, 0.53)	(99%, 0.79)	(98%, 0.87)	(97%, 0.89)	(98%, 0.88)	(99%, 0.91)	(98%, 0.90)	(94%, 0.49)	(99%, 0.62)	(99%, 0.70)
PI v Rater2	(98%, 0.46)	(99%, 0.86)	(99%, 0.86)	(93%, 0.74)	(94%, 0.65)	(97%, 0.78)	(97%, 0.79)	(94%, 0.44)	(99%, 0.63)	(99%, 0.66)
PI v Rater3	(96%, 0.17)	(99%, 0.65)	(98%, 0.58)	(92%, 0.74)	(95%, 0.75)	(97%, 0.83)	(94%, 0.63)	(93%, 0.41)	(99%, 0.57)	(99%, 0.81)
Rater1 v Rater2	(99%, 0.63)	(99%, 0.77)	(99%, 0.86)	(94%, 0.76)	(94%, 0.65)	(97%, 0.80)	(97%, 0.81)	(95%, 0.63)	(99%, 0.51)	(99%, 0.61)
Rater1 v Rater3	(97%, 0.44)	(99%, 0.70)	(98%, 0.57)	(93%, 0.75)	(95%, 0.76)	(98%, 0.83)	(93%, 0.57)	(93%, 0.53)	(99%, 0.53)	(99%, 0.69)
Rater2 v Rater3	(97%, 0.35)	(99%, 0.63)	(98%, 0.62)	(91%, 0.66)	(92%, 0.53)	(97%, 0.76)	(93%, 0.48)	(94%, 0.57)	(99%, 0.53)	(99%, 0.58)

*Data displayed in the table as (Percent Absolute Agreement, Cohen's Kappa)

**UT3 was the only session where the Wand-Like device was used.

LEGEND
K < 0.40
0.40 < K < 0.50
0.50 < K

Quantified Measures of Usability

Video	UT_Session	User_Group	Video_Type	Coding_Stage	Rater	Total_Time (s)	PL (%)	PL (s)	PL_Diff (%)	PL_Diff (s)	PL_Diff_Ratio (%)	WL (%)	WL (s)	WL_Diff (%)	WL_Diff (s)	WL_Diff_Ratio (%)	Help (%)	Help (s)	Help_Ratio (%)
1101	1	I	C	1	ALL	2319	23.18%	538	10.51%	244	45.34%	0.00%	0	0.00%	0	0	1.36%	32	5.87%
1102	1	I	A	1	2	3334	18.40%	613	2.56%	85	13.91%	0.00%	0	0.00%	0	0	1.15%	38	6.25%
1103	1	G	A	1	1	2434	19.76%	481	6.60%	161	33.40%	0.00%	0	0.00%	0	0	0.09%	2	0.46%
1104	1	G	A	1	3	2366	14.29%	338	4.64%	110	32.47%	0.00%	0	0.00%	0	0	1.23%	29	8.61%
2107	1	G	A	2	1	2813	9.45%	266	2.13%	60	22.54%	0.00%	0	0.00%	0	0	0.35%	10	3.70%
2108	1	I	A	2	2	1729	16.78%	290	5.26%	91	31.35%	0.00%	0	0.00%	0	0	6.27%	108	37.37%
2109	1	U	A	2	3	1940	17.38%	337	7.99%	155	45.97%	0.00%	0	0.00%	0	0	0.21%	4	1.21%
3110	1	G	A	3	3	3073	10.99%	338	3.29%	101	29.94%	0.00%	0	0.00%	0	0	2.35%	72	21.38%
3111	1	I	A	3	1	2856	12.71%	363	2.26%	65	17.78%	0.00%	0	0.00%	0	0	0.58%	17	4.56%
3112	1	U	A	3	1	2327	10.85%	252	1.90%	44	17.51%	0.00%	0	0.00%	0	0	0.71%	17	6.54%
3113	1	G	A	3	2	1867	6.74%	126	1.27%	24	18.84%	0.00%	0	0.00%	0	0	0.79%	15	11.72%
3114	1	G	A	3	2	2554	5.75%	147	1.20%	31	20.87%	0.00%	0	0.00%	0	0	0.00%	0	0.00%
3115	1	U	A	3	3	2339	15.85%	371	3.02%	71	19.05%	0.00%	0	0.00%	0	0	1.24%	29	7.82%
1201	2	U	A	1	2	2753	7.97%	219	1.48%	41	18.57%	0.00%	0	0.00%	0	0	1.73%	48	21.71%
1203	2	I	A	1	1	2615	10.37%	271	3.92%	103	37.80%	0.00%	0	0.00%	0	0	2.31%	60	22.28%
1205	2	I	A	1	3	2480	15.01%	372	7.06%	175	47.04%	0.00%	0	0.00%	0	0	1.42%	35	9.46%
2206	2	I	A	2	3	1938	11.21%	217	2.25%	44	20.07%	0.00%	0	0.00%	0	0	0.24%	5	2.14%
2207	2	U	C	2	ALL	3225	15.30%	493	2.00%	65	13.07%	0.00%	0	0.00%	0	0	1.27%	41	8.30%
2208	2	U	A	2	2	2520	4.91%	124	0.54%	14	11.00%	0.00%	0	0.00%	0	0	0.99%	25	20.16%
2209	2	U	A	2	1	2224	7.00%	156	0.88%	20	12.57%	0.00%	0	0.00%	0	0	0.72%	16	10.29%
3210	2	U	A	3	1	2092	12.90%	270	1.29%	27	10.00%	0.00%	0	0.00%	0	0	0.14%	3	1.09%
3211	2	FG	A	3	2	2548	9.85%	251	1.38%	35	14.01%	0.00%	0	0.00%	0	0	0.73%	19	7.41%
3212	2	FG	A	3	3	3057	12.50%	382	0.93%	28	7.44%	0.00%	0	0.00%	0	0	1.41%	43	11.28%
1301	3	U	A	1	3	3544	29.19%	1034	5.34%	189	18.29%	8.84%	313	2.49%	88	0.2817	1.50%	53	3.94%
1302	3	U	A	1	1	2626	20.09%	528	2.23%	59	11.10%	9.36%	246	0.53%	14	0.0566	0.11%	3	0.37%
1303	3	U	A	1	2	2877	12.36%	356	0.62%	18	5.02%	8.23%	237	0.03%	1	0.0036	2.19%	63	10.64%
1304	3	U	A	1	3	3367	6.58%	222	0.31%	10	4.71%	8.41%	283	1.08%	36	0.1284	2.19%	74	14.61%
1305	3	U	A	1	1	2873	15.74%	452	1.60%	46	10.17%	4.73%	136	0.64%	18	0.1353	0.16%	5	0.78%
1306	3	U	A	1	2	3044	9.13%	278	0.37%	11	4.05%	13.10%	399	2.66%	81	0.2031	1.24%	38	5.58%
2307	3	G	A	2	2	3352	8.26%	277	0.39%	13	4.72%	5.33%	179	0.09%	3	0.0169	1.23%	41	9.05%
2308	3	G	A	2	1	3287	7.18%	236	0.80%	26	11.14%	8.38%	275	0.21%	7	0.0251	0.40%	13	2.57%
2309	3	G	A	2	2	3055	12.77%	390	0.98%	30	7.67%	1.88%	57	0.07%	2	0.0372	2.92%	89	19.93%
2311	3	I	A	2	3	3072	11.89%	365	5.52%	170	46.43%	10.86%	334	0.91%	28	0.0838	3.37%	104	14.81%
2312	3	I	A	2	3	3567	9.45%	337	0.54%	19	5.71%	22.81%	814	0.76%	27	0.0333	1.36%	49	4.22%
2313	3	I	A	2	1	3063	6.18%	189	0.14%	4	2.27%	15.16%	464	1.16%	36	0.0765	0.80%	25	3.75%
3310	3	G	C	3	ALL	2304	9.65%	222	0.95%	22	9.84%	12.34%	284	0.30%	7	0.0243	0.00%	0	0.00%
3314	3	U	A	3	3	2266	18.01%	408	0.42%	10	2.33%	14.54%	329	0.00%	0	0	0.55%	12	1.69%
3315	3	U	A	3	2	3015	4.98%	150	0.06%	2	1.20%	17.82%	537	0.37%	11	0.0208	0.00%	0	0.00%
3316	3	I	A	3	1	3897	17.97%	700	0.05%	2	0.28%	8.25%	322	0.32%	12	0.0388	0.30%	12	1.14%

* All percent values (%)—with the exception of the Ratios, which are defined below—are %-coded of Total_Time.

Nomenclature:

PL: Using pen-like device
PL_Diff: Difficulty while using pen-like device
PL_Diff_Ratio: PL_Diff / PL
Help: Moderator was providing help

WL: Using wand-like device
WL_Diff: Difficulty while using wand-like device
PL_Diff_Ratio: WL_Diff / WL
Help_Ratio: Help / (PL + WL)

BIBLIOGRAPHY

[1] L. T. Kohn, J. Corrigan, and M. S. Donaldson, *To err is human : building a safer health system*, Washington, D.C.: National Academy Press, 2000.

[2] J. T. James, "A New, Evidence-based Estimate of Patient Harms Associated with Hospital Care," *Journal of Patient Safety*, vol. 9, no. 3, pp. 122-128, 2013.

[3] S. L. Dawson, and J. A. Kaufman, "The imperative for medical simulation," *Proceedings of the IEEE,* vol. 86, no. 3, pp. 479-483, 1998.

[4] A. Ziv, P. R. Wolpe, S. D. Small, and S. Glick, "Simulation-based medical education: an ethical imperative," *Acad Med,* vol. 78, no. 8, pp. 783-8, Aug, 2003.

[5] E. A. Hunt, M. Fiedor-Hamilton, and W. J. Eppich, "Resuscitation Education: Narrowing the Gap Between Evidence-Based Resuscitation Guidelines and Performance Using Best Educational Practices," *Pediatric Clinics of North America,* vol. 55, no. 4, pp. 1025-1050, 2008.

[6] K. Rosen, "The history of simulation," *The comprehensive textbook of healthcare simulation*, pp. 5-49: Springer, 2013.

[7] H. Samia, S. Khan, J. Lawrence, and C. P. Delaney, "Simulation and Its Role in Training," *Clinics in Colon and Rectal Surgery,* vol. 26, no. 1, pp. 47-55, 2013.

[8] R. M. Satava, "The revolution in medical education-the role of simulation," *J Grad Med Educ,* vol. 1, no. 2, pp. 172-5, Dec, 2009.

[9] R. M. Satava, "Historical Review of Surgical Simulation—A Personal Perspective," *World Journal of Surgery,* vol. 32, no. 2, pp. 141-148, 2008.

[10] Society of American Gastrointestinal and Endoscopic Surgeons. "Fundamentals of Laproscopic Surgery Program Description," November 1, 2016; http://www.flsprogram.org/index/fls-program-description/.

[11] Society of American Gastrointestinal and Endoscopic Surgeons. "The Fundamentals of Endoscopic Surgey Program Description," November 1, 2016; http://www.fesprogram.org/about/program-description-2/.

[12] The American Board of Anesthesiology. "About MOCA 2.0," November 1. 2016; http://www.theaba.org/MOCA/About-MOCA-2-0.

[13] J. K. Hayden, R. A. Smiley, M. Alexander, S. Kardong-Edgren, and P. R. Jeffries, "Supplement: The NCSBN National Simulation Study: A Longitudinal, Randomized,

Controlled Study Replacing Clinical Hours with Simulation in Prelicensure Nursing Education," *Journal of Nursing Regulation,* vol. 5, no. 2, pp. C1-S64, 2014.

[14] C. McIntosh, A. Macario, B. Flanagan, and D. M. Gaba, "Simulation: What does it really cost?," *Simulation in Healthcare,* vol. 1, no. 2, pp. 109, 2006.

[15] J. B. Cooper, and V. R. Taqueti, "A brief history of the development of mannequin simulators for clinical education and training," *Qual Saf Health Care,* vol. 13 Suppl 1, pp. i11-8, Oct, 2004.

[16] M. L. Good, "Patient simulation for training basic and advanced clinical skills," *Medical Education,* vol. 37, pp. 14-21, 2003.

[17] P. Bradley, "The history of simulation in medical education and possible future directions," *Med Educ,* vol. 40, no. 3, pp. 254-62, Mar, 2006.

[18] A. Grenvik, and J. Schaefer, "From Resusci-Anne to Sim-Man: the evolution of simulators in medicine," *Crit Care Med,* vol. 32, no. 2 Suppl, pp. S56-7, Feb, 2004.

[19] S. Abrahamson, and J. S. Denson, "A Developmental Study of Medical Training Simulators for Anesthesiologists," 1968.

[20] A. R. Molnar, and B. Sherman, "U.S. Office of Education Support of Computer Activities," 1969, p. 13.

[21] "CPI Inflation Calculator," U.S Bureau of Labor Statistics, 2014.

[22] S. Abrahamson, "Human Simulation for Training in Anesthesiology," *Medical Engineering,* pp. 370-374, Chicago, IL: Year Book Medical Publishers, 1974.

[23] S. Abrahamson, J. S. Denson, and R. M. Wolf, "Effectiveness of a simulator in training anesthesiology residents," *J Med Educ,* vol. 44, no. 6, pp. 515-9, Jun, 1969.

[24] M. S. Gordon, "Cardiology patient simulator: Development of an animated manikin to teach cardiovascular disease," *The American Journal of Cardiology,* vol. 34, no. 3, pp. 350-355, 1974.

[25] M. S. Gordon, G. A. Ewy, A. C. DeLeon Jr, R. A. Waugh, J. M. Felner, A. D. Forker, I. H. Gessner, J. W. Mayer, and D. Patterson, ""Harvey," the cardiology patient simulator: Pilot studies on teaching effectiveness," *The American Journal of Cardiology,* vol. 45, no. 4, pp. 791-796, 1980.

[26] M. L. Good, S. Lampotang, G. Gibby, and J. Gravenstein, "Critical events simulation for training in anesthesiology," *J Clin Monit Comput,* vol. 4, no. 140, 1988.

[27] D. M. Gaba, and A. DeAnda, "A comprehensive anesthesia simulation environment: re-creating the operating room for research and training," *Anesthesiology,* vol. 69, no. 3, pp. 387-394, 1988.

[28] N. J. Maran, and R. J. Glavin, "Low- to high-fidelity simulation – a continuum of medical education?," *Medical Education,* vol. 37, pp. 22-28, 2003.

[29] R. Scalese, V. Obeso, and S. B. Issenberg, "Simulation Technology for Skills Training and Competency Assessment in Medical Education," *Journal of General Internal Medicine,* vol. 23, no. 1, pp. 46-49, 2008.

[30] K. M. Ventre, and H. A. Schwid, "Computer and Web Based Simulators," *The comprehensive textbook of healthcare simulation,* pp. 191-208: Springer, 2013.

[31] "Virtual Dissection Table," August 9, 2013; http://medical.anatomage.com/.

[32] "Medrills," August 9, 2013; www.archiemd.com.

[33] C. Epps, M. L. White, and N. Tofil, "Mannequin Based Simulators," *The comprehensive textbook of healthcare simulation,* pp. 209-232: Springer, 2013.

[34] R. Owens, and J. M. Taekman, "Virtual Reality, Haptic Simulators, and Virtual Environments," *The comprehensive textbook of healthcare simulation,* pp. 233-253: Springer, 2013.

[35] W. Buxton. "Multi-Touch Systems that I Have Known and Loved," January 18, 2015; http://www.billbuxton.com/multitouchOverview.html.

[36] SimVentures. "SimStore," November 19, 2013; http://www.mysimcenter.com/en-US/SimStoreHome.aspx.

[37] B. Bray, C. R. Schwartz, D. L. Weeks, and S. Kardong-Edgren, "Human Patient Simulation Technology: Perceptions From a Multidisciplinary Sample of Health Care Educators," *Clinical Simulation in Nursing,* vol. 5, no. 4, pp. e145-e150, 2009.

[38] D. A. Jansen, N. Johnson, G. Larson, C. Berry, and G. H. Brenner, "Nursing Faculty Perceptions of Obstacles to Utilizing Manikin-based Simulations and Proposed Solutions," *Clinical Simulation in Nursing,* vol. 5, no. 1, pp. e9-e16, 2009.

[39] B. Zendejas, R. Brydges, A. Wang, and D. Cook, "Patient Outcomes in Simulation-Based Medical Education: A Systematic Review," *Journal of General Internal Medicine,* vol. 28, no. 8, pp. 1078-1089, 2013/08/01, 2013.

[40] Y. Okuda, E. O. Bryson, S. DeMaria, L. Jacobson, J. Quinones, B. Shen, and A. I. Levine, "The utility of simulation in medical education: what is the evidence?," *Mount Sinai Journal of Medicine: A Journal of Translational and Personalized Medicine,* vol. 76, no. 4, pp. 330-343, 2009.

[41] D. Nestel, J. Groom, S. Eikeland-Husebø, and J. M. O'Donnell, "Simulation for learning and teaching procedural skills: the state of the science," *Simulation in Healthcare,* vol. 6, no. 7, pp. S10-S13, 2011.

[42] W. C. McGaghie, S. B. Issenberg, M. E. R. Cohen, J. H. Barsuk, and D. B. Wayne, "Does simulation-based medical education with deliberate practice yield better results than traditional clinical education? A meta-analytic comparative review of the evidence," *Academic medicine: journal of the Association of American Medical Colleges,* vol. 86, no. 6, pp. 706, 2011.

[43] D. A. Cook, R. Hatala, R. Brydges, B. Zendejas, J. H. Szostek, A. T. Wang, P. J. Erwin, and S. J. Hamstra, "Technology-enhanced simulation for health professions education: a systematic review and meta-analysis," *Jama,* vol. 306, no. 9, pp. 978-988, 2011.

[44] D. A. Cook, R. Brydges, S. J. Hamstra, B. Zendejas, J. H. Szostek, A. T. Wang, P. J. Erwin, and R. Hatala, "Comparative Effectiveness of Technology-Enhanced Simulation Versus Other Instructional Methods: A Systematic Review and Meta-Analysis," *Simulation in Healthcare,* vol. 7, no. 5, pp. 308-320 10.1097/SIH.0b013e3182614f95, 2012.

[45] D. A. Cook, S. J. Hamstra, R. Brydges, B. Zendejas, J. H. Szostek, A. T. Wang, P. J. Erwin, and R. Hatala, "Comparative effectiveness of instructional design features in simulation-based education: Systematic review and meta-analysis," *Medical Teacher,* vol. 35, no. 1, pp. e867-e898, 2013.

[46] M. Wenk, R. Waurick, D. Schotes, M. Wenk, C. Gerdes, H. K. Van Aken, and D. M. Pöpping, "Simulation-based medical education is no better than problem-based discussions and induces misjudgment in self-assessment," *Advances in health sciences education,* vol. 14, no. 2, pp. 159-171, 2009.

[47] L. R. Schwartz, R. Fernandez, S. R. Kouyoumjian, K. A. Jones, and S. Compton, "A Randomized Comparison Trial of Case-based Learning versus Human Patient Simulation in Medical Student Education," *Academic Emergency Medicine,* vol. 14, no. 2, pp. 130-137, 2007.

[48] H. Ishii, and B. Ullmer, "Tangible bits: towards seamless interfaces between people, bits and atoms," in Proceedings of the ACM SIGCHI Conference on Human factors in computing systems, Atlanta, Georgia, USA, 1997, pp. 234-241.

[49] M. Weiser, "The computer for the 21st century," *SIGMOBILE Mob. Comput. Commun. Rev.,* vol. 3, no. 3, pp. 3-11, 1999.

[50] P. Wellner, W. Mackay, and R. Gold, "Back to the real world," *Commun. ACM,* vol. 36, no. 7, pp. 24-26, 1993.

[51] "VIMEDIX Ultrasound Simulator," August 9, 2013; http://caehealthcare.com/eng/ultrasound-simulators/vimedix.

[52] "Arthro Mentor," August 4, 2015; http://simbionix.com/simulators/arthro-mentor/.

[53] R. T. Azuma, "A survey of augmented reality," *Presence-Teleoperators and Virtual Environments,* vol. 6, no. 4, pp. 355-385, Aug, 1997.

[54] F. Sauer, S. Vogt, and A. Khamene, "Augmented Reality," *Image-Guided Interventions*, T. Peters and K. Cleary, eds., pp. 81-119: Springer US, 2008.

[55] M. Blackwell, C. Nikou, A. M. DiGioia, and T. Kanade, "An Image Overlay system for medical data visualization," *Medical Image Analysis,* vol. 4, no. 1, pp. 67-72, 3//, 2000.

[56] A. M. I. DiGioia, B. Jaramaz, and B. D. Colgan, "Computer Assisted Orthopaedic Surgery: Image Guided and Robotic Assistive Technologies," *Clinical Orthopaedics and Related Research,* vol. 354, pp. 8-16, 1998.

[57] D. Shelton, G. Stetten, and W. Chang, "Ultrasound visualization with the sonic flashlight," in ACM SIGGRAPH 2002 conference abstracts and applications, San Antonio, Texas, 2002, pp. 82-82.

[58] R. L. Klatzky, B. Wu, D. Shelton, and G. Stetten, "Effectiveness of augmented-reality visualization versus cognitive mediation for learning actions in near space," *ACM Trans. Appl. Percept.,* vol. 5, no. 1, pp. 1-23, 2008.

[59] N. Glossop, C. Wedlake, J. Moore, T. Peters, and Z. Wang, "Laser projection augmented reality system for computer assisted surgery," *Medical Image Computing and Computer-Assisted Intervention-MICCAI 2003,* pp. 239-246: Springer, 2003.

[60] R. Krempien, H. Hoppe, L. Kahrs, S. Daeuber, O. Schorr, G. Eggers, M. Bischof, M. W. Munter, J. Debus, and W. Harms, "Projector-Based Augmented Reality for Intuitive Intraoperative Guidance in Image-Guided 3D Interstitial Brachytherapy," *International Journal of Radiation Oncology*Biology*Physics,* vol. 70, no. 3, pp. 944-952, 3/1/, 2008.

[61] T. Ni, A. K. Karlson, and D. Wigdor, "AnatOnMe: facilitating doctor-patient communication using a projection-based handheld device." pp. 3333-3342.

[62] A. Kotranza, D. S. Lind, and B. Lok, "Real-time evaluation and visualization of learner performance in a mixed-reality environment for clinical breast examination," *IEEE Trans Vis Comput Graph,* vol. 18, no. 7, pp. 1101-14, Jul, 2012.

[63] A. Kotranza, D. S. Lind, C. M. Pugh, and B. Lok, "Real-time in-situ visual feedback of task performance in mixed environments for learning joint psychomotor-cognitive tasks." pp. 125-134.

[64] A. Kotranza, and B. Lok, "Virtual human+ tangible interface= mixed reality human an initial exploration with a virtual breast exam patient." pp. 99-106.

[65] A. Kotranza, B. Lok, A. Deladisma, C. M. Pugh, and D. S. Lind, "Mixed reality humans: evaluating behavior, usability, and acceptability," *IEEE Trans Vis Comput Graph,* vol. 15, no. 3, pp. 369-82, May-Jun, 2009.

[66] C. Bichlmeier, F. Wimme, S. M. Heining, and N. Navab, "Contextual anatomic mimesis hybrid in-situ visualization method for improving multi-sensory depth perception in medical augmented reality." pp. 129-138.

[67] D. Kondo, and R. Kijima, “A study on perception and operation using free form projection display,” *Interactive Technologies and Sociotechnical Systems,* vol. 4270, pp. 103-109, 2006.

[68] D. Kondo, Y. Shiwaku, and R. Kijima, "Free form projection display and application." p. 8.

[69] J. T. Samosky, B. Mikulis, R. Bregman, and D. A. Nelson, “A novel automated drug simulant recognition system for naturalistic real-time medical simulation,” *Stud Health Technol Inform,* vol. 173, pp. 430-2, 2012.

[70] J. T. Samosky, E. Baillargeon, R. Bregman, A. Brown, A. Chaya, L. Enders, D. A. Nelson, E. Robinson, A. L. Sukits, and R. A. Weaver, “Real-time "x-ray vision" for healthcare simulation: an interactive projective overlay system to enhance intubation training and other procedural training,” *Stud Health Technol Inform,* vol. 163, pp. 549-51, 2011.

[71] J. T. Samosky, D. A. Nelson, B. Wang, R. Bregman, A. Hosmer, B. Mikulis, and R. Weaver, “BodyExplorerAR: enhancing a mannequin medical simulator with sensing and projective augmented reality for exploring dynamic anatomy and physiology,” in Proceedings of the Sixth International Conference on Tangible, Embedded and Embodied Interaction, Kingston, Ontario, Canada, 2012, pp. 263-270.

[72] J. R. Maltby, and M. T. Beriault, “Science, pseudoscience and Sellick,” *Can J Anaesth,* vol. 49, no. 5, pp. 443-7, May, 2002.

[73] N. Holmes, D. Martin, and A. M. Begley, “Cricoid pressure: a review of the literature,” *J Perioper Pract,* vol. 21, no. 7, pp. 234-8, Jul, 2011.

[74] B. A. Sellick, “Cricoid pressure to control regurgitation of stomach contents during induction of anaesthesia,” *The Lancet,* vol. 278, no. 7199, pp. 404-406, 8/19/, 1961.

[75] R. Vanner, and T. Asai, “Safe use of cricoid pressure,” *Anaesthesia,* vol. 54, no. 1, pp. 1-3, 1999.

[76] R. Vanner, and B. Pryle, “Regurgitation and oesophageal rupture with cricoid pressure: a cadaver study,” *Anaesthesia,* vol. 47, no. 9, pp. 732-735, 1992.

[77] R. L. Johnson, E. K. Cannon, C. B. Mantilla, and D. A. Cook, “Cricoid pressure training using simulation: a systematic review and meta-analysis,” *Br J Anaesth,* vol. 111, no. 3, pp. 338-46, Sep, 2013.

[78] K. J. Smith, J. Dobranowski, G. Yip, A. Dauphin, and P. T. Choi, “Cricoid pressure displaces the esophagus: an observational study using magnetic resonance imaging,” *Anesthesiology,* vol. 99, no. 1, pp. 60-4, Jul, 2003.

[79] K. G. Allman, “The effect of cricoid pressure application on airway patency,” *J Clin Anesth,* vol. 7, no. 3, pp. 197-9, May, 1995.

[80] C. J. Flucker, E. Hart, M. Weisz, R. Griffiths, and M. Ruth, "The 50-millilitre syringe as an inexpensive training aid in the application of cricoid pressure," *Eur J Anaesthesiol,* vol. 17, no. 7, pp. 443-7, Jul, 2000.

[81] A. Kopka, and D. Robinson, "The 50 ml syringe training aid should be utilized immediately before cricoid pressure application," *Eur J Emerg Med,* vol. 12, no. 4, pp. 155-8, Aug, 2005.

[82] N. L. Herman, B. Carter, and T. K. Van Decar, "Cricoid pressure: teaching the recommended level," *Anesthesia & Analgesia,* vol. 83, no. 4, pp. 859-863, 1996.

[83] A. Kopka, and J. Crawford, "Cricoid pressure: a simple, yet effective biofeedback trainer," *Eur J Anaesthesiol,* vol. 21, no. 6, pp. 443-7, Jun, 2004.

[84] H. Owen, V. Follows, K. J. Reynolds, G. Burgess, and J. Plummer, "Learning to apply effective cricoid pressure using a part task trainer," *Anaesthesia,* vol. 57, no. 11, pp. 1098-101, Nov, 2002.

[85] P. Aspden, and Institute of Medicine (U.S.). Committee on Identifying and Preventing Medication Errors., *Preventing medication errors,* Washington, DC: National Academies Press, 2007.

[86] V. Wirtz, K. Taxis, and N. D. Barber, "An observational study of intravenous medication errors in the United Kingdom and in Germany," *Pharm World Sci,* vol. 25, no. 3, pp. 104-11, Jun, 2003.

[87] K. Taxis, and N. Barber, "Causes of intravenous medication errors: an ethnographic study," *Qual Saf Health Care,* vol. 12, no. 5, pp. 343-7, Oct, 2003.

[88] K. C. Nanji, A. Patel, S. Shaikh, D. L. Seger, and D. W. Bates, "Evaluation of Perioperative Medication Errors and Adverse Drug Events," *Anesthesiology,* vol. 124, no. 1, pp. 25-34, Jan, 2016.

[89] J. Bonkowski, C. Carnes, J. Melucci, J. Mirtallo, B. Prier, E. Reichert, S. Moffatt-Bruce, and R. Weber, "Effect of barcode-assisted medication administration on emergency department medication errors," *Acad Emerg Med,* vol. 20, no. 8, pp. 801-6, Aug, 2013.

[90] P. J. Helmons, L. N. Wargel, and C. E. Daniels, "Effect of bar-code-assisted medication administration on medication administration errors and accuracy in multiple patient care areas," *Am J Health Syst Pharm,* vol. 66, no. 13, pp. 1202-10, Jul 1, 2009.

[91] R. D. Paoletti, T. M. Suess, M. G. Lesko, A. A. Feroli, J. A. Kennel, J. M. Mahler, and T. Sauders, "Using bar-code technology and medication observation methodology for safer medication administration," *Am J Health Syst Pharm,* vol. 64, no. 5, pp. 536-43, Mar 1, 2007.

[92] E. G. Poon, C. A. Keohane, C. S. Yoon, M. Ditmore, A. Bane, O. Levtzion-Korach, T. Moniz, J. M. Rothschild, A. B. Kachalia, J. Hayes, W. W. Churchill, S. Lipsitz, A. D.

Whittemore, D. W. Bates, and T. K. Gandhi, “Effect of bar-code technology on the safety of medication administration,” *N Engl J Med,* vol. 362, no. 18, pp. 1698-707, May 6, 2010.

[93] D. G. Ford, A. L. Seybert, P. L. Smithburger, L. R. Kobulinsky, J. T. Samosky, and S. L. Kane-Gill, “Impact of simulation-based learning on medication error rates in critically ill patients,” *Intensive care medicine,* vol. 36, no. 9, pp. 1526-1531, 2010.

[94] D. A. Norman, *The psychology of everyday things*, [London: BasicBooks, 1988.

[95] D. A. Norman, and S. W. Draper, *User Centered System Design; New Perspectives on Human-Computer Interaction*: L. Erlbaum Associates Inc., 1986.

[96] G. Bregman, Lehocky, Scharl, and Samosky, "Continuous Color Visual Feedback for Safe and Accurate Device Operation: The Smart Syringe Injection Training System," University of Pittsburgh, 2009, unpublished.

[97] J. T. Samosky, B. Wang, D. A. Nelson, R. Bregman, A. Hosmer, and R. A. Weaver, “BodyWindows: enhancing a mannequin with projective augmented reality for exploring anatomy, physiology and medical procedures,” *Stud Health Technol Inform,* vol. 173, pp. 433-9, 2012.

[98] M. Lee, and Samosky, "Applying Cricoid Pressure," University of Pittsburgh, 2012, unpublished.

[99] W. M. Haynes, D. R. Lide, T. J. Bruno, and CRC Press, *CRC handbook of chemistry and physics : a ready-reference book of chemical and physical data*, 94th edition (2013-2014) ed., Boca Raton, Florida: CRC Press Taylor & Francis Group, 2013.

[100] R. T. da Rocha, I. G. R. Gutz, and C. L. do Lago, “A Low-Cost and High-Performance Conductivity Meter,” *Journal of Chemical Education,* vol. 74, no. 5, pp. 572, 1997/05/01, 1997.

[101] L. EDWON, "Uniduino: Adruino for Unity," 2013.

[102] Zygote Media Group, Inc. "Zygote: 3D Male Collection," February 9, 2015; https://www.zygote.com/poly-models/3d-human-collections/3d-male-anatomy-collection.

[103] J. Tabor. "Jeremy Tabor: Character Modeler and Texture Painter," February 9, 2015; http://tabor3d.com/.

[104] "StrangeIoC," November 1, 2016; https://strangeioc.github.io/strangeioc/.

[105] M. Fowler. "Inversion of Control Containers and the Dependency Injection Pattern," November 1, 2016; http://www.martinfowler.com/articles/injection.html.

[106] T. Tullis, and B. Albert, *Measuring the user experience : collecting, analyzing, and presenting usability metrics*, Amsterdam ; Boston: Elsevier/Morgan Kaufmann, 2008.

[107] J. Rubin, and D. Chisnell, *Handbook of Usability Testing: How to Plan, Design, and Conduct Effective Tests*: Wiley Publishing, 2008.

[108] J. Nielsen, and T. K. Landauer, "A mathematical model of the finding of usability problems," in Proceedings of the INTERACT '93 and CHI '93 Conference on Human Factors in Computing Systems, Amsterdam, The Netherlands, 1993, pp. 206-213.

[109] R. A. Virzi, "Refining the Test Phase of Usability Evaluation: How Many Subjects Is Enough?," *Human Factors: The Journal of the Human Factors and Ergonomics Society*, vol. 34, no. 4, pp. 457-468, August 1, 1992, 1992.

[110] SignUpGenius. "SignUp Genius: About Us," November 1, 2016; http://www.signupgenius.com/about/signupgenius.cfm.

[111] P. E. Phrampus, and J. M. O'Donnell, "Debriefing using a structured and supported approach," *The Comprehensive Textbook of Healthcare Simulation*, A. I. Levine, S. DeMaria, Jr., A. D. Schwartz and A. J. Sim, eds., pp. 73-84: Springer, 2014.

[112] QSR International. "What is NVivo," November 1, 2016; http://www.qsrinternational.com/what-is-nvivo.

[113] K. L. Gwet, *Handbook of inter-rater reliability: The definitive guide to measuring the extent of agreement among raters*: Advanced Analytics, LLC, 2014.

[114] J. R. Landis, and G. G. Koch, "The measurement of observer agreement for categorical data," *Biometrics*, vol. 33, no. 1, pp. 159-74, Mar, 1977.

[115] QSR International. "Run a Coding Comparison Query," November 1, 2016; http://help-nv11.qsrinternational.com/desktop/procedures/run_a_coding_comparison_query.htm.

[116] Laerd_Statistics. "Kruskal-Wallis H test using SPSS statistics," November 1, 2016; https://statistics.laerd.com/.

[117] Laerd_Statistics. "Statistical tutorials and software guides," November 1, 2016; https://statistics.laerd.com/.

[118] S. K. Card, A. Newell, and T. P. Moran, "The psychology of human-computer interaction," 1983.

[119] A. Lesgold, S. Lajole, M. Bunzo, and G. Eggan, *Sherlock: A coached practice environment for an electronics troubleshooting job*, ED299450, Learning Research and Development Center, University of Pittsburgh, Retrieved from ERIC database, 1988.

[120] K. R. Koedinger, J. R. Anderson, W. H. Hadley, and M. A. Mark, "Intelligent Tutoring Goes to School in the Big City," *International Journal of Artificial Intelligence in Education*, vol. 8, pp. 30-43, 1997.

[121] J. R. Anderson, R. Farrell, and R. Sauers, "Learning to Program in LISP," *Cognitive Science,* vol. 8, no. 2, pp. 87-129, 1984.

[122] K. VanLehn, "The relative effectiveness of human tutoring, intelligent tutoring systems, and other tutoring systems," *Educational Psychologist,* vol. 46, no. 4, pp. 197-221, 2011.

[123] N. Stylopoulos, S. Cotin, S. K. Maithel, M. Ottensmeyer, P. G. Jackson, R. S. Bardsley, P. F. Neumann, D. W. Rattner, and S. L. Dawson, "Computer-enhanced laparoscopic training system (CELTS): bridging the gap," *Surgical Endoscopy And Other Interventional Techniques,* vol. 18, no. 5, pp. 782-789, 2004.

[124] C. M. Pugh, and P. Youngblood, "Development and Validation of Assessment Measures for a Newly Developed Physical Examination Simulator," *Journal of the American Medical Informatics Association,* vol. 9, no. 5, pp. 448-460, 2002-09-01 00:00:00, 2002.

[125] T. R. Mackel, J. Rosen, and C. M. Pugh, "Markov Model Assessment of Subjects' Clinical Skill Using the E-Pelvis Physical Simulator," *Biomedical Engineering, IEEE Transactions on,* vol. 54, no. 12, pp. 2133-2141, 2007.

[126] V. S. Alagar, and K. Periyasamy, *Specification of Software Systems*: Springer Publishing Company, Incorporated, 2011.

[127] N. Blumhardt. "Stateless library," https://github.com/dotnet-state-machine/stateless.

[128] J. T. Samosky, A. Thornburg, T. Karkhanis, F. Petraglia, E. Strickler, D. A. Nelson, R. A. Weaver, and E. Robinson, "Enhancing medical device training with hybrid physical-virtual simulators: smart peripherals for virtual devices," *Stud Health Technol Inform,* vol. 184, pp. 377-9, 2013.

[129] D. A. Nelson, and J. T. Samosky, "The tool positioning tutor: a target-pose tracking and display system for learning correct placement of a medical device," *Stud Health Technol Inform,* vol. 163, pp. 400-2, 2011.

[130] "Biogears Open-Source Physiology Engine," November 1, 2016; https://biogearsengine.com/.

[131] "Dell Interactive Projector - S320wi," December 17, 2015; http://accessories.us.dell.com/sna/productdetail.aspx?c=us&l=en&s=bsd&cs=04&sku=225-3948.