University of Alberta

USING VIRTUAL WORLDS FOR SCENARIO-BASED TRAINING

by

David Chodos

A thesis submitted to the Faculty of Graduate Studies and Research
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

Department of Computing Science

Fall 2012
Edmonton, Alberta

Abstract

In this thesis, we present a novel, robust virtual world-based educational platform. The framework provides scenario definition and execution capabilities that support the creation and delivery of a wide range of virtual world-based training scenarios. The components built upon this framework, meanwhile, include a set of recording and analysis tools that enable the instructors to assess students' behaviour in a comprehensive manner.

We also present several empirical evaluations that provide evidence of the framework's utility and educational relevance. Through a case study in an EMT training context, we provide support for the functionality of the framework in delivering a collaboration-focused training scenario. In a marketing case study, we provide a demonstration of the range of analysis capabilities incorporated into the system. Finally, working within an ER context, we provide evidence of the educational impact of a scenario developed with the framework.

Acknowledgements

I am grateful for the assistance and support of many people who have helped me in various ways over the last several years.

First, my colleagues in the Software Systems Research Group, who have lent a hand with technical questions, an ear for discussions, and provided me with lots of thought-provoking questions and suggestions. I would particularly like to thank Mike Smit, Ken Bauer, and Barry Gergel for their friendship and support over the past five years.

I would like to thank the students who have worked with me on the MeRiTS project: Pawel Kuras, Matthew Delaney, and Lucio Gutierrez. You have all contributed to the project in various ways, and I really appreciate the hard work, dedication, and creativity that you have put into it.

I am also grateful for the support provided by researchers both within the University of Alberta and other educational institutions. I have been fortunate to be able to collaborate with people from a variety of disciplines, including health sciences education, educational psychology, paramedic training, and nursing. Mike Carbonaro, Sharla King, Tracey Stephen, Paul Messinger, Patricia Boechler, Kristen Smirnov, Erik de Jong, Eva Wasniewski, and Elaine Greidanus have collaborated on writing research papers, designing experiments, and generally helping advance my research. The studies and experiments that I have run have also been supported by the efforts of many others at a range of institutions across Edmonton: Mike East, Lisa Torres, Norbert Werner, Mark MacKenzie, Andrew Reid, Ken Brisbin, Katherine Bowman, and Jennifer Kwan have all contributed to the successful design, implementation and execution of the scenarios and experiments that I have run over the course of my research.

On this note, I would like to extend my thanks to all of the students who have participated in the experiments, from the faculties of Nursing and Business at the University of Alberta, the Paramedic program at NAIT, and everyone who participated in two “Save Stan Saturday” events in 2011 and 2012. I am grateful for your enthusiasm, your keen interest, and your willingness to try something new, which has given me many insights into the experience of actually *participating* in a virtual world-based training scenario.

I could not have completed this research without the guidance and support of my supervisor, Eleni Stroulia. She has provided me with the space to be creative, while unfailingly pointing out the places where this creativity needed to be supported by technical rigour and depth. At the same time, she has time and again shown a great deal of compassion and understanding when I have come up against the kind of hurdles that raising a young family while pursuing a degree can throw in one's path.

Finally, I am profoundly grateful for the unwavering support of my family: my parents, Bob Chodos and Andrea Leis; my in-laws, Rob and Ellen Krahn; my children, Sam and Rachel; and especially my wife, Natasha Krahn, who has shown extraordinary patience, understanding, and support as I have pursued this research.

Table of Contents

List of Tables

List of Figures

List of Acronyms

ABM	Avatar Behaviour Model
API	Application Programming Interface
AVA	Actionable Video Annotation
BPEL	Business Process Execution Language
BPM	Beats Per Minute
CBP	Case-Based Planning
CVE	Collaborative Virtual Environment
CRPG	Computer Role Playing Game
ELM	Exploratory Learning Model
EMT	Emergency Medical Technician
ER	Emergency Room
HMD	Head-mounted display
GDP	Generative Design Pattern
IV	Intravenous line
LMS	Learning Management System
LSL	Linden Scripting Language
MeRiTS	Mixed Reality Training System
MUVE	Multi-User Virtual Environment
NAIT	Northern Alberta Institute of Technology
NWN	Neverwinter Nights
ODE	Orchestration Director Engine
PPE	Personal Protective Equipment
RTS	Real-Time Strategy game
STAT	Scenario Trace Analysis Toolkit
UI	User Interface
VOIP	Voice Over IP
VR	Virtual Reality
VUE	Visual Understanding Environment
XML	eXtensible Markup Language

Chapter 1

Introduction

Scenario-based training is an experience-based approach to learning that is compelling on several levels. It is motivated by established educational theories, and is part of a growing competency-based approach to teaching. Moreover, it provides students with realistic, relevant opportunities to practice their skills. These educational opportunities are valuable for teaching situational assessment and problem solving skills, providing opportunities for applying relevant knowledge, and are especially important for training in professional disciplines where the tasks being taught involve communication among participants in a complex environment.

However, the real-world delivery of this kind of training is challenging for educators, for several reasons. First, it requires that students and educators come together in a common physical space, therefore excluding distance learners. Second, there are significant limitations on how many students can participate in a given training session. Third, the learning that occurs through this training is often solely experiential, since educators usually have limited resources with which to analyze the behaviour of multiple students.

A virtual world is an online platform that offers solutions to these challenges, as well as additional benefits. Broadly speaking, a virtual world offers its users a common online space to meet, interact, and share experiences. The capabilities of each virtual world may vary somewhat, but there are several common core characteristics. First, each user interacts in a virtual world through a representation of the user called an *avatar*. The avatar may be customized by the user, so that a user can "be himself" in the virtual world, or take on a new role or persona. Second, while in these shared virtual spaces, users can interact with objects, which can realistically simulate the behaviour of their real-world counterparts. Finally, the virtual world offers users a variety of methods of instant communication (such as text chat, voice chat, and instant messaging), which allow users to interact with each other in a realistic way. Taken together, these features enable users to be able to have realistic, meaningful and interactive experiences in the virtual world.

The capabilities provided by a virtual world offer educators an opportunity to create compelling scenario-based training programs. The use of avatars allows users to take on real-world roles in a convincing manner. The interactive, immersive capabilities of the virtual world allow students to participate in realistic, collaborative learning experiences. Indeed, several programs have begun exploring the use of virtual worlds in their curricula, and have developed scenario-based training programs.

1.1 Research Problem

Although the prospect of using virtual worlds for scenario-based training is quite promising, the current state-of-the-art in virtual world-based scenario training has three fundamental shortcomings:

a. a lack of a systematic development methodology,

b. a lack of support for recording and analyzing student actions and providing feedback, and

c. a lack of systematic evaluation of the scenario-based training programs.

First, the scenario-based training programs that have been created thus far are not systematic or reusable. That is, they have been created with a particular audience in mind (e.g., emergency responders), and for a specific educational purpose (e.g., flu pandemic response training). From a software engineering point of view, these systems have not been designed with reuse in mind. Any significant change in the audience, educational goals, or underlying technology (e.g., virtual world platform) requires an overhaul of a large portion of the system.

Second, the programs do not provide instructors with an adequate set of tools for analyzing and understanding the students' behaviour as the students participate in the training scenarios. That is, while instructors are able to observe students while they go through the scenario in the virtual world (as they would in real life), there has been little work put into recording students' actions, analyzing those actions, or presenting instructors with any kind of summary of this analysis that could lead to a better understanding of the students' experiences and their educational outcomes.

Third, while some effort has been put into *developing* scenario-based training programs, relatively little has gone into enabling instructors to *assess the effectiveness* of these programs. To this end, assessment capabilities should be developed to enable instructors to address a few basic questions about the educational impact of virtual world-based training scenarios. Do the students learn anything from their experience? Are they any better off, educationally, after the training scenario than they were before? What are the differences, in terms of both educational outcomes and student attitudes, between a virtual world-based scenario and a real-life scenario? Equipped with answers to these questions, instructors would be able to make informed decisions about the appropriateness of virtual world-based training scenarios for their students, classrooms, and educational environments.

In this work, I present three thesis statements, to address the aforementioned shortcomings. First, I have developed a software framework to support the development and enactment of virtual world-based training scenarios. The use of the framework has been demonstrated with a particular virtual world, but its design makes it possible to integrate other virtual worlds in a straightforward manner. The framework provides auxiliary services that supersede the limitations of current virtual worlds, providing educationally-relevant capabilities that would not otherwise be available.

Second, the framework enables instructors to (a) record the students actions while participating in the scenario, and (b) analyze these actions after the scenario has been completed. This capability enables instructors to examine the relationships between the students' behaviour and the educational goals of the scenario and to better assess the students' knowledge and competence.

Third, through empirical studies, I have gathered qualitative and quantitative evidence of the impact of virtual world-based training scenarios. The framework supports the relatively rapid development of a range of scenarios, while the recording and analysis tools supply the data required to draw conclusions about the impact of the scenario, both in terms of educational outcomes and behavioural and attitudinal changes.

1.2 Contributions

I support the core thesis statements with four key contributions, which are described in the following paragraphs.

First, I present the Mixed Reality Training System (MeRiTS) framework, which enables the development of virtual world-based training scenarios in a discipline- and platform-independent manner. Using this development framework, scenarios can be – and have been – created for a variety of disciplines. Two such scenarios are described in Sections 5.1 and 5.3. This has the potential to reduce the time required to develop the technical components of a scenario, and therefore allow educators and researchers to focus on the educational goals and content of the scenario. Moreover, this framework uses a modular, component-based design, and its reliance on a general model for describing avatar behaviour in any virtual world means that it is not limited by the restrictions of a particular virtual world. In principle, the framework can use any one of several virtual worlds to deliver the scenario to the users. Finally, the framework is supported by a model of participant behaviour, object interactions, and normative action sequences that, taken together, provide robust, platform-independent mechanisms for scenario definition and enactment.

Second, I present an action recording and analysis toolkit, which records and analyzes each student's actions as they participate in a scenario. This set of recording and analysis tools serves three key purposes. First, it allows educators to parse, analyze, and ultimately understand the actions taken by students while participating in a scenario. Through these capabilities, the student's actions can be evaluated in a detailed, rigorous manner, and the educational impact of the scenario on that student can be assessed in a meaningful way. Section 5.1 describes an experiment which made use of these action analysis capabilities to assess students' learning outcomes. On a broader level, by enabling educational assessment of individual students, educators can then compare a student's performance against that of other students, and compare the performance of one or more students over time. Finally, the performance of an entire class of students can be aggregated and analyzed in order to assess the educational effectiveness of the scenario as a whole, or to compare one scenario to another. Therefore, through the toolkit one can assess the educational impact of virtual world-based scenarios in general, and determine when and where they can be used most effectively.

Third, in Chapter 5 I present a series of empirical studies that begin to address the issue of the educational effectiveness of virtual world-based training scenarios. These evaluations cover a range of disciplines, provide validation for components of the MeRiTS framework, and assess the educational impact of scenarios created using MeRiTS. Specifically, in Section 5.1 I describe an experiment that provides validation for the action recording component, and also provides evidence of the scenario's educational effectiveness. In the experiment described in Section 5.2, I make extensive use of the framework's recorded action analysis capabilities, in order to answer specific research questions. Finally, in Section 5.3, I present an experiment that directly addresses the issue of educational impact, by offering several assessments of the impact of students' participation in a MeRiTS-based scenario. Thus, through these evaluations, the effectiveness of the framework is demonstrated, and the educational impact of the "virtual world-based scenario" approach is supported.

Finally, through the process of conducting rigorous background research, designing the MeRiTS framework, and implementing numerous scenarios, I have come up with a set of recommendations (described in Section 5.5) for designing, implementing, and delivering educational scenarios. These recommendations address issues such as managing the iterative development process, ensuring the scenario is educationally appropriate and sufficiently validated, and incorporating adequate training opportunities into the scenario-based training experience. Using this model along with the MeRiTS framework, researchers and educators can collaborate to develop educational scenarios quickly and effectively.

1.3 Outline

The rest of this dissertation is organized as follows.

In Chapter 2, I present a thorough survey of the literature related to my research. In this chapter, I cover a wide range of existing virtual world-based education projects, discuss relevant theories from educational psychology, and highlight relevant research from a variety of fields that informed the design of the MeRiTS framework.

In Chapter 3, I present my Avatar Behaviour Model (ABM), which supports the MeRiTS framework. This model encapsulates, in a virtual world-independent manner, the range of educationally-relevant actions that can be taken by an avatar in a virtual world. In this chapter, I also describe a recording component that I have developed, for use in a specific virtual world, based on this model.

In Chapter 4, I present a detailed description of the MeRiTS framework. This includes a description of the three-tiered architecture of the framework and each of its constituent components, and some of the design decisions that went into implementing the virtual world client. In this chapter, I also present the Scenario Trace Analysis Toolkit (STAT), which is crucial in terms of the educational effectiveness of the framework as a whole.

In Chapter 5, I present a description of the empirical evaluations that were conducted using the MeRiTS framework. These evaluations serve two purposes: first, they provide validation and evaluation for various components of the MeRiTS framework. Second, the last evaluation provides educational validation for the framework as a whole, providing empirical evidence of the educational impact of a particular scenario. Following these evaluations, I present guidelines for developing scenarios based upon my experience with the aforementioned evaluations.

Finally, in Chapter 6, I present my contributions, and describe several promising areas for future work. Based on these experiences, I present a set of recommendations for MeRiTS-based virtual world scenario development and deployment.

Chapter 2

Related Work

My research spans several fields of study in computer science and education.

First, I have been guided in my research by principles from educational psychology. I have based my work on a solid foundation of existing educational psychology research, both in terms of relevant pedagogical theories and empirical studies supporting the use of virtual worlds for education. Additionally, there are several broader trends in education that help motivate and support the use of virtual world-based education. These trends are discussed in some detail, with a particular focus on their impact on the viability of virtual world-based education.

Second, I am developing a general framework for authoring, enacting, and recording simulations in a virtual world. Motivated by these goals, I have conducted a survey of existing virtual world-based training projects, with a particular focus on those projects that are similar to ours in one or more key areas – that is, those that use simulation-based training, have been validated educationally, or are extensible in some way. Although I have considered a few projects from the games research area, I have chosen not to rigorously analyze this area, since its aims differ from ours in a few key ways. First, I am focusing specifically on programs developed within a virtual world environment, while games research spans a broad range of platforms, including on-line, console, and single-player contexts. Second, I am focused specifically on programs that provide some educational benefit or skill training, while games research is often focused on areas such as player enjoyment, immersion, or interactivity with the underlying goal of improving of gameplay quality, rather than educational impact. Finally, my research is reliant on the use of a modifiable platform – which can then be integrated with a broader set of web services-based capabilities – while games research often uses commodity or commercial platforms.

Third, I have consulted literature on behaviour modelling and analysis, to ensure that the framework captures relevant user behaviour, and provides appropriate capabilities for analyzing that behaviour. At a high level, therefore, I have looked at various process modelling methods, to ensure that I am able to capture the process behind any scenario in a robust, effective manner. Next, I consulted literature on modelling virtual worlds – and specifically avatar behaviour – to ensure that I am considering an appropriate range of user actions. On a pedagogical level, I have studied the educational psychology literature to ensure that this recording is educationally relevant and technically effective, and that the analysis tools that are provided are appropriate for the assessment of the students' educational outcomes.

2.1 Simulation and Virtual Worlds in Education

My work is motivated by the growing use of virtual worlds for education, and the desire to understand the educational impact of having students participate in virtual world-based educational programs. My research, therefore, had to be informed by established theories from educational psychology. These theories fall into two categories. First, there are theories that support the educational effectiveness of interactive and scenario-based training, in general. Second, there are studies that validate the use of virtual worlds as a means of delivering education using this approach.

2.1.1 Educational Trends

There are three broad trends in the field of education that are helping foster a climate where this kind of training can flourish.

One of the most readily apparent of these trends is the growing prominence of **distance learning** in higher education. As online technology continues to improve, educational institutions are able to offer distance learning that has the potential to be "just as good, or even better in some cases, than face-to-face traditional learning" [5]. Moreover, as shown by the steadily increasing number of students taking courses by distance (from just over 1.5 million in 1997-1998, to 3 million in 2000-2001, to 3.9 million in 2007 [85]), and the success of institutions such as the University of Phoenix (380,000 students in 2009 [1]) and Athabasca University (38,000 students in Canada and abroad[2]), there is an increasingly important distinction between traditional "brick and mortar" institutions, "virtual" institutions that deliver their content entirely through distance learning, and blended "brick-and-click" institutions [5].

Another factor supporting the development of virtual world-based training is the **emerging "Net" generation**. This term refers to people for whom "computer games, emails, the Internet, cell phones and instant messaging are integral parts of their lives", and thus "think and process information differently than their predecessors" [67], [7]. Most pedagogically relevant, however, is the observation by Creanor that members of this generation tend "to be highly skilled networkers [who] often used technology to pull support when needed" [15]. Oblinger, similarly, described members of the Net generation as people who are "comfortable with new technologies, want information quickly, multi-task well, and [who] use several channels to retrieve information and communicate" [65]. Similarly, Small described these students as "cradled in technology; they are intuitively tech-competent, exploring and trying things out" [74]. Efforts to adapt educational approaches to these students have ranged from emphasizing student-centered learning, to incorporating mobile and "Web 2.0" technology, to developing educational platforms that emphasize knowledge seeking, social interaction, and collaboratively-defined "folksonomies" [80]. Given these descriptions, the socially-oriented, technologically cutting-edge virtual world platform is a good fit with the characteristics of this demographic group. Further supporting the use of simulation-based training is the familiarity that this group has with video games. According to a recent survey of undergraduate students by Kapralos et al., the majority of respondents played video games, and of those more than half of them played more than 10 hours a week [48]. In this survey, the researchers also assessed the students' attitude towards virtual simulations and

[1] See http://nces.ed.gov/fastfacts/display.asp?id=74

[2] See http://www2.athabascau.ca/aboutau/glance.php

serious games, and found that a large majority (83%) felt that including virtual simulations into the curriculum would improve their education [48]. In a similar study, a recent survey of medical students in the U.S. found that students felt that video games could have educational value (80%), and were quite likely to use multiplayer simulations if they helped develop patient interaction skills (90%) [51]. More broadly, the respondents were almost unanimous in advocating the use of technology to enhance healthcare education (98%), and specifically using new media technologies (96%).

Along with changes in educational institutions and the demographics of students, there have also been changes in the prevailing approaches to education itself. Many fields of study, and especially the medical field, have been moving recently towards **competency-based education**. Voorhees described the shift towards competency-based education as part of "a learning revolution" emphasizing skill acquisition, intensely focused learning experiences, and a range of delivery options [83]. However, this shift is accompanied by an urgent need to develop appropriate methods of measuring learning outcomes, to ensure that students are, in fact, acquiring the necessary knowledge and skills. Similarly, ten Cate identified this trend as "a major advancement", but highlights the need to address "critical issues of curricular implementation" [78]. The most important of these issues is ensuring that a student's mastery of general competency areas is relevant to real-world clinical practice. Frankel cited competency-based training as a useful tool in addressing a disconnect between intensivist students' testing – which focuses on critical care medicine – and their training, in which this subject is addressed inconsistently [24]. Similarly, at the University of Washington School of Medicine, educators have responded to a decrease in students' clinical skills training by developing an integrated curriculum based around a well-defined set of competency domains [28]. Specifically, five domains are identified in this curriculum, and within each domain students are expected to show proficiency at increasingly advanced levels, measured by developmental benchmarks.

> *Competency-based education is part of a "learning revolution" emphasizing skill acquisition and intensely focused learning experiences* [83]

2.1.2 Educational Theories

Simulation-based education is motivated, both implicitly and explicitly, by a number of pedagogical theories. These theories are summarized in Table 2.1.

At a very high level, this kind of training is supported by constructivism, which was described by Duffy et al. [23]. Constructivism is based on Dewey's foundational theory of situated learning, which views learning as an interaction between the environment and the learner [20]. The theory of constructivism sees learning as an active process of *constructing*, rather than acquiring, knowledge. Instruction within a constructivist context, therefore, focuses on supporting a student's construction of knowledge, rather than simply presenting knowledge for the student to absorb. A crucial tenet of constructivism is that "there is no ultimate, shared reality, but rather, reality that is the outcome of constructive processes." [23]. This is a sharp contrast to objectivism, which holds that there is a structure to the world, independent of an individual's experience, and thus the goal of education is to understand this structure. From the constructivist perspective, the goal of instruction is to show students how to "construct plausible interpretations" of the world around them, rather than imparting some existing, "correct" structure [23]. From this perspective, an interactive,

Table 2.1: Overview of Educational Theories

Educational Theory	**Key Papers**	**Key idea**
Constructivism	Dewey (1916) [20], Duffy et al. (1992) [23]	Knowledge is constructed, not imparted
Experiential Learning	Kolb et al. (1975) [50]	Learning occurs through experience, reflection, and concept formation
Situated Cognition (evolved from Constructivism)	Brown et al. (1989) [4]	Learning is influenced by the context in which it occurs
Cognitive Apprenticeship (evolved from Situated Cognition)	Collins (1991) [14], Järvelä (1995) [44]	Learning can be viewed as an apprenticeship process
Rich Environments for Active Learning	Grabinger et al. (2000) [29]	Learning is most effective when supported by authentic context and assessment
Exploratory Learning (evolved from Experiential Learning)	de Freitas et al. (2009) [17]	Learning is an exploratory process, influenced by real and virtual experiences
Hierarchy of Educational Engagement Types	Messinger (2010) [59]	Educational engagement can be categorized according to the type of communication used by students and teachers

immersive approach to education is encouraged, since it engages students in the knowledge construction process.

> *An interactive, immersive approach to education engages students in the knowledge construction process.* [23]

Situated cognition, proposed by Brown et al., starts from a constructivist viewpoint, and applies these principles to the issue of the context in which learning occurs [4]. According to situated cognition, knowledge is not a set of abstract concepts to be absorbed by the student; instead, it is dependent on the context and culture in which it is used. Learning, therefore, is most effective when it occurs in a context that closely replicates that in which the knowledge will be applied. Conventional classroom-based education, on the other hand, “defeats its own goal of providing useable, robust knowledge” by ignoring the context and culture of the content being taught. The authors also discuss the issue of *authentic* vs. *hybrid* activities. *Authentic* activities are derive their meaning from their cultural context, and are defined as “the ordinary practices of the culture”. *Hybrid* activities, on the other hand, are presented as being authentic, but are in fact defined by another, external culture. Classroom activities are often hybrid, being influenced by the school culture in which they are being taught, and bearing little similarity to the actual activities that would be undertaken by real-world practitioners. This distinction is of particular importance when considering the experience that results from these activities; authentic activities provide students with the understanding of the relevant actions and context necessary to “gain access to the standpoint that enables practitioners to act meaningfully and purposefully”. Simulation-based learning, therefore, can support the use of authentic activities to provide students with appropriately situated, meaningful learning experiences.

Further refining the theory of situated cognition, Collins developed the cognitive apprenticeship model of educational practice, which incorporates the situated nature of the knowledge being conveyed to students [14]. This model views each student as an apprentice, and the instructor as a master, in the collaborative task of learning a particular concept or skill. As with situated cognition, the goal is to provide students with authentic, culturally appropriate experiences, which they can use to gradually gain mastery of a skill or knowledge area. Broadly speaking, the instructor's job is first to model the skill being taught, then to help the student acquire the skill, and finally to "fade out" as the student masters the skill. This process emphasizes the importance of scaffolding to allow students to continually build on existing knowledge. Initially, the student is presented with a task embedded within a familiar problem, to emphasize the validity of the student's implicit (pre-existing) knowledge, and is supported by the instuctor in understanding the new context. As the student gains confidence and takes on more challenging tasks, collaboration among students is encouraged to foster the students' immersion in the new culture and to allow students to gradually generalize their skills and knowledge. At this point in the process, the instructor's role is to foster this collaboration, and to guide students in correcting misconceptions. Finally, as the student becomes more autonomous, and gains a better understanding of the culture and principles at work, they are able to take on harder and more nebulously-defined problems, and transition from students to independent practitioners in the field.

As Collins describes, simulation-based learning is an excellent vehicle for teaching using a cognitive apprenticeship approach, for several reasons. First, the realistic, immersive approach of simulation-based learning is very much in keeping with the cognitive apprenticeship model, with its emphasis on modelling authentic activities and the culture associated with those activities. To this end, the simulation can be imbued with precisely the characteristics that are required for the student to gain the required skill mastery and cultural understanding. Second, students can repeatedly practice a skill through multiple iterations of a scenario, with the instructor increasing the difficulty of the scenario as the students gradually gain mastery over the skill being taught. That is, the instructor can customize the details of the scenario according to the degree to which students have a) learned the relevant skills, b) become immersed in the associated culture, and c) gained a generalized understanding of the principles at work behind the skills being taught. Third, the instructor can take on a variety of roles within the simulation, depending on the students' progress through the apprenticeship process. Initially, the instructor can play the role of a practitioner or expert in order to convey the necessary skills and culture. Next, the instructor can become a colleague, helping the students master the necessary skills and enter the culture. Finally, the instructor can take on the role of a customer, client or adversary, providing increasingly challenging problems for the students to solve.

> *The realistic, immersive approach of simulation-based learning is very much in keeping with the cognitive apprenticeship model.* [14]

The cognitive apprenticeship model was later evaluated by Järvelä, who found it to be effective within a technologically rich learning environment [44]. Specifically, a class of Grade 7 students was given a programming task to complete, and the teacher followed cognitive apprenticeship-based methods, such as scaffolding, modelling, and reflection, in guiding the students towards solutions to the task. Järvelä found, by conducting a qualitative analysis of the students' interactions with the teacher, that these strategies were generally effective, and students were able to benefit from the teacher's guidance. However, Järvelä also noted that the success of these methods was dependent on a reciprocal understanding of the situation between the teacher and the students.

Akin to constructivism is experiential learning theory developed by Kolb et al. [50], which views the learning process as a cycle involving four activities: concrete experience, observation and reflection, abstract-concept formation, and testing in new situations. This theory places much importance on reflection and abstract-concept formation, in addition to the learner's concrete experiences, as part of the learning process.

Building on Kolb's experiential-learning theory, de Freitas et al. proposed the exploratory-learning model (ELM) for learning within an immersive environment [17]. ELM introduces the notion of "exploration" through observations or collaborative activities. The ELM model is based on five phases: experience, exploration, reflection, forming abstract concepts, and testing in different situations. The first phase encapsulates the experiences that that the student might have as part of the learning process. This model expands upon Kolb's "concrete experience" activity by considering a range of experiences – that is, abstract and virtual experiences, as well as the lived experiences identified by Kolb. De Freitas et al. observe that virtual experiences "may be as definitive as those in the live context," and that they "may have a reinforcing impact upon learning objectives by allowing for mistakes to be made in a secure environment". The next phase, exploration, reflects the potential for students to become active and autonomous learners, especially through embedded 3D platforms that "bring alive the places within the classroom". However, the authors also recognize that employing an exploratory learning strategy requires new approaches to tasks such as planning learning objectives and assessment. Reflection, meanwhile, is highlighted as essential to enabling learning transfer between the virtual and real world. The next phase, the formation of abstract concepts, encapsulates a key output of an effective learning process. It is informed by the first three stages, and can then be validated or refined through the final testing phase. The stages of reflection and forming abstract concepts are acknowledged as potentially challenging for students, but the authors propose that these challenges can be mitigated through appropriate guidance from instructors. Finally, social interaction is identified as a key aspect of the learning experience that permeates the five phases, encompassing guidance from instructors, the creation of shared understanding of abstract concepts, and group social experiences.

The exploratory learning model considers abstract and virtual experiences, as well as the lived experiences identified by Kolb. [17]

In related work, Grabinger et al. proposed five key principles for a rich environment for active learning (REAL) [29]. These five principles are student responsibility and initiative, generative learning, authentic learning context, authentic assessment, and co-operative support. Taken together, these five principles support a constructivist learning environment where students are actively, co-operatively involved in learning realistic skills and principles, and are evaluated in a meaningful, appropriate way. Concepts from Grabinger's work that are relevant to my research are generative learning, an approach to learning which encourages students to generate new ideas through active involvement, and therefore relates to scenario-based training; creating an authentic learning context, which is connected to the realism and immersiveness of the virtual world platform; and authentic assessment, which is supported by my action recording and analysis tools.

These theories and studies support the value of simulation-based training in general and virtual world-based training more specifically, which presents students with knowledge and teaches them skills in a context and culture similar to that within which they will be using those knowledge and skills. Indeed, in analyzing the viability of virtual worlds for educational, Coffman et al. identified the potential for a learner to "take on roles to solve a problem that is relevant to the learner's real world" [13]. In terms of my research, this provides support for the use of avatars to allow students to either practice their own role, or take on other roles, in the process of going through a simulation exercise. In this analysis, Coffman highlighted the connections between virtual world-based education and social constructivism, communal constructivism, flow theory, and student-centred learning. Hobbs et al., meanwhile, asserted that "the social aspects of learning are also well supported by a virtual world and help maintain interest" [39]. In my research, support is provided both for verbal communication, through text and voice-based communication, and non-verbal communication, through inter-avatar distance and gestures. Winn et al., too, supported the use of desktop virtual environments for education, stating that "the sense of presence that students feel in the virtual environments is of major importance in the learning process", since it involves first-person experience, and thus supports the social construction of knowledge, according to the theory of social constructivism [88]. This validates my approach of providing students with an immersive, first-person experience.

The social aspects of learning are well supported by a virtual world. [39]

Finally, Messinger et al. proposed a means of categorizing the level of educational engagement according to the communication method used by students and teachers [59]. According to Messinger, educational engagement can be divided into four broad categories. The simplest of these, one-way communication, is analogous to the "traditional" model of a teacher presenting material to a class of students. Two-way communication broadens this to incorporate communication from the student back to the teacher, as well. The third category, education co-creation, enables the student and teacher to collaborate in creating educational content and experiences. Finally, community building views the teacher and students as members of an educational community, where everyone is able to participate in the educational process. Messinger advocates for the use of virtual worlds to deliver educational experiences (particularly within the context of corporate education), as they can "be used to promote engagement at all four levels of the proposed hierarchy" [59].

2.1.3 Learning in Virtual Environments

Supporting these theories, empirical studies have shown that the characteristics of virtual world-based education can indeed be quite effective. The experiments are summarized at a high level in Table 2.2, and described in the following paragraphs. It should be noted that most of these empirical studies have used virtual *reality*, rather than virtual *worlds*. That is, while virtual worlds are online environments that are accessed using a standard desktop/laptop computer interface (keyboard, mouse, and monitor), virtual reality systems are not usually online, and often involve specialized interface equipment, such as head-mounted displays (HMDs), pointing wands, and in some cases special large and/or curved screens. However, the immersive nature of virtual reality systems is a key characteristic that is shared by virtual worlds, and thus the results of the experiments. which explored this aspect of VR systems, are applicable to VW systems, as well. That is, these experiments provide empirical validation for the impact of immersive learning environments on specific aspects of the educational process.

Table 2.2: Summary of Empirical Validation of Virtual World-based Education

Author	**Skill Taught**	**Empirical Evaluation**
Cho et al.(2002) [10]	Attention	Used VR (HMDs), found that it improved results
Waller et al.(2001) [86]	Spatial skills	Compared VW to RL, found equivalent results
Goebbels et al.(2001) [27]	Anatomy (positioning bones)	Found that VR (CVE) provided good support for training
Huang et al.(2010) [40]	Student interest	Tested web-based tool, results showed increased interest, motivation

Cho et al. found that an immersive, VR-based cognitive training tool was more effective than a standard, non-immersive tool, and resulted in improved maintenance of attention [10]. Specifically, 26 teenagers were randomly assigned to one of three groups – VR training, non-VR training, or no training – for a period of two weeks. The students in the VR training group were placed in a simulated classroom, using HMDs and a 3D model created with 3D Studio Max. Within this environment, the students completed exercises that were designed to improve their ability to a) focus on a task and b) sustain attention to a task, involving responding to flags of certain colours and specific sequences of numbers, respectively. The students in the non-VR group performed the same exercises, but used a computer monitor and 2-D interface, rather than the HMD used by the VR group. The students in all groups were assessed through a Continuous Performance Task (CPT) test, which was given before and after the training sessions. While the students in all three groups showed some improvement from the pre- to post-training tests, the students in the VR group showed the most improvement, in terms of number of correct answers and decrease in perceptual sensitivity. These results indicate the positive effect of increased immersion in the training process.

Waller et al. have shown that spatial skills learned in a virtual environment transfer to real life [86]. In the experiment, participants were trained in navigating a maze, either in the real world or within a virtual environment. Within the virtual environment, participants were again divided into two groups: one was presented with a wire-frame representation of the maze, while the other was shown a maze with surface rendering. These three training environments (real-world, wireframe rendering and surface rendering) are shown in Figure 2.1. The participants were then tested in the real-world version of the maze, by estimating the location of a set of landmarks from several "sighting locations." The participants' performance was assessed by measuring the difference between the actual bearing and distance for each landmark-sighting location pair. The researchers found that while there was some difference between the real-world and surface-rendered virtual environment groups, this difference was much smaller than the difference that resulted from the characteristics of individual participants, such as prior computer experience, gender, and maze exploration time. The key conclusion that may be drawn from this result is that the participants' performance in the real-world maze was equally proficient after participating in training sessions in a variety of environments, with varying degrees of "realism." One can conclude that the participants were able to successfully transfer their navigational skills from the virtual world to the real world, even when the visualization provided by the virtual world was not entirely realistic.

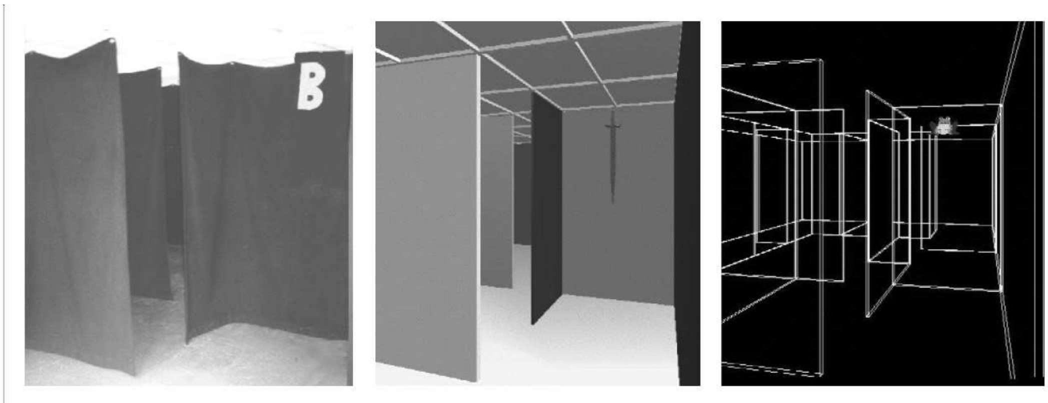


Figure 2.1: Training environments for spatial skills experiment (from [86])

In two related studies, researchers in collaborative virtual environments (CVEs) reached similar conclusions about the effect of immersiveness and virtual "presence" on education. A CVE is a VR system that is designed to be used by multiple participants. For instance, Goebbels et al. analyzed a distributed CVE (that is, designed for use in multiple locations) in terms of its suitability for medical education according to criteria such as integration of audio, video, and remote user representation. Specifically, The reseachers tested the student's ability to position three bones in a skeleton, and extrapolated from these results to draw conclusions about the use of the CVE for medical education, in general. The reseachers found that an immersive experience – including an audio link, video streaming, and appropriate representations of remote users and tools – supported the students and teachers participating in the bone-positioning task [27]. Moreover, forcing the user to compensate for missing functionality resulted in a decrease in usability and a sense of co-presence. In evaluating a web-based tool to teach anatomy through 3D renderings of the human body, Huang et al. found that students reacted quite positively to the experience [40]. Notably, the students reported increased motivation, interest, and attention in using the 3D system.

2.2 Virtual Worlds

One of the foundational propositions of my research is that virtual worlds can be taken beyond their immediately-obvious target applications in the social and entertainment spheres, and leveraged as a platform for compelling, meaningful educational experiences. Indeed, the possibility of using virtual worlds for education has been recognized by researchers for several years. Graham et al. identified Second Life as a place where educational institutions can "bring distance education students together, simulate experiences that would be too difficult... to undertake in real life" [30]. Similarly, Junglas noted that "virtual environments are capable of transforming processes and business environments by providing a safe haven for exploring, innovating and testing ideas", describing an example of training an emergency response team in new techniques [47]. Burkle et al., meanwhile, focused on the social aspect of the learning experience, highlighting the potential for “networking and collaboration, in an immersive experience and a participatory culture” [7].

> *Virtual worlds can be leveraged as a platform for compelling, meaningful educational experiences.* [30], [47], [7]

Educators have also begun to recognize the potential of virtual world-based training. In a recent survey of 162 post-secondary educators conducted by Neely et al., respondents emphasized the ability of virtual world-based training to simulate a real environment, and thus create an authentic learning experience [63]. The respondents also focused on the theme of student responsibility, highlighting the ability of virtual worlds to foster generative learning strategies, and the social aspects of virtual world-based training. However, the theme of technology as a hurdle was also a prevalent theme in the survey, with respondents describing the steep learning curve and technical instability associated with using Second Life. This indicates that there are still significant barriers to widespread adoption of virtual worlds for education.

There have also been a few commerical virtual world platforms designed to deliver virtual training experiences; the most well-known of these is the OLIVE (On-Line Interactive Virtual Environment) platform, which was developed by Forterra, and is now owned by SAIC[3]. OLIVE is an “enterprise-grade virtual world software platform” which supports interactive training simulations for a range of contexts, with a particular focus on government, military and healthcare. The OLIVE platform is priced according to the number of users, in increments ranging from 50 to 5,000. There are also additional training courses and content packs available, as well as a software development kit (SDK) which allows the development of custom prototypes and scenarios. The number of users the system is able to support and the range of support options available indicate that the system is intended for large companies and organizations, and requires high-level organizational support. Furthermore, the content packs that have recently been made available – organizational collaboration, army, navy, and first responder – indicates the extent to which the OLIVE platform is focused on military, business and disaster training contexts. To sum up, while the OLIVE platform provides precisely the kind of functionality that is required to deliver immersive, effective virtual world-based training simulations, it is clear that the platform requires a significant investment of both money and expertise on an organization-wide level.

Other commercial ventures include TruSim[4], a division of Blitz Games Studios that “creates effective, immersive and engaging training games using... 3D games techniques”, and ProtoSphere[5], an “immersive, social, learning and collaboration enterprise environment.” TruSim is currently focused on conducting a patient rescue case study to “ensure that instructional design for learning and game design are properly aligned [and] provide methods of validating that learning has taken place.” While this is certainly a worthwhile goal, and is closely related to my own research efforts, it is not clear what progress has been made toward this goal. Moreover, there does not appear to be any framework or model in place that would allow this case study to be generalized to other contexts. ProtoSphere, meanwhile, is oriented towards “business-focused collaboration and communication tools” that facilitate activities such as remote meetings, and does not appear to have much interest in developing educational applications.

[3] http://www.saic.com/products/simulation/olive

[4] http://www.trusim.com

[5] http://www.protonmedia.com/the-product/protosphere-overview/

Researchers have, for several years, been exploring various ways that virtual worlds can be used as a platform for training programs. It should be noted that the majority of these projects, unlike mine, are focused on a single application. Despite this limitation, an understanding of these projects, in terms of the technology used, educational contexts, and evaluation methods, is essential to my work. Furthermore, I will focus here, for the most part, on projects that use virtual worlds for interactive, task-based training, since that is the approach that I have taken in my work. However, I do present projects that use other approaches, to give a sense of the scope of the work that is occurring in this area.

Before describing work in this area, it is necessary to define what, exactly, I mean by virtual world-based training. The terms “virtual world” and “virtual environment” are used quite loosely by researchers and educators, and can refer to anything from a virtual reality (VR) training tool to a piece of software that uses a 3D interface [35], [73]. While there have been several efforts to define virtual worlds, the one that comes closest to my intentions is that of Schroeder, who defines a collaborative virtual environment as one in which “users experience other participants as being present in the same environment and interacting with them – or ‘being there together’ ” [72]. In my research, I use the following definition: a virtual world is one where multiple users can communicate synchronously in an environment with a 3-D perspective. This differentiates virtual worlds from stand-alone software on the one hand, which lacks collaborative communication functionality, and from VR-based tools on the other hand, which may contain communication capabilities, but are limited to interaction within a particular VR engine or platform.

2.2.1 Use of Virtual World Platforms in Education

Virtual worlds have been used for education in a variety of ways, and across numerous disciplines, and several researchers have conducted broad surveys of these efforts.

Cai et al. examined the potential of virtual worlds for conducting training programs [8]. In this work, he presented several useful analyses. First, he analyzed common virtual learning activities, including classroom emulation, co-creation, learning by doing, and simulation, with respect to the kinds of interaction involved (student, teacher, and community), and how each would be implemented in a virtual world. In Cai’s model, there are seven phases of development:

1. Identify business objectives of the new learning solution
2. Conduct needs analysis
3. Describe learning objectives
4. Develop game design
5. Conduct pilot test
6. Deploy the game
7. Conduct evaluation

Cai also compared several virtual environments – Second Life, Active Worlds, OpenSim, and the Torque game engine – in terms of their fitness for educational activities, using criteria such as ease of use, security, and development cost. He concluded that there was no single engine that would be ideal for all situations, but rather that each had advantages and disadvantages, which would make it well-suited for some applications, but not for others. In terms of my research, Cai describes Second Life as “suitable for multiplayer online collaboration”, and highlights both the widespread adoption of the platform and the broad range of user-generated content that is available.

> *Second Life is suitable for multiplayer online collaboration, based on its widespread adoption and the broad range of available content.* [8]

In a similar effort, de Freitas wrote a scoping study presenting the various ways that virtual worlds were being used for education, across a variety of virtual worlds and educational disciplines [16]. In this study, de Freitas presented projects that used Active Worlds, Second Life, Croquet, Project Wonderland, and OLIVE in contexts ranging from teaching science in K-8 schools, to conducting virtual conferences, to creating collaborative, mixed-reality learning spaces. A summary of the projects is shown in Table 2.3.

Table 2.3: Case Studies (summarized from [16])

Virtual World	**Context**	**Educational approach**
Active Worlds	Science (K-8)	Co-creation of artifact representing learning outcome
Second Life (SciLands)	Virtual conference	Meeting space for “hosting” an international conference
	Planitarium	Presentation space for astronomy students
Qwaq	Meta-Institute for Computational Astrophysics	Meeting space to foster collaboration among colleagues
Wonderland	MiRTLE	Distributed classroom environment, used simultaneously by students in UK, China
OLIVE	Surgical training, incident response	Providing scenario-based emergency response training for clinicians

Through these case studies, it is apparent that virtual worlds may be used for a wide variety of educational purposes, and that numerous options are available for the implementation of such projects. Furthermore, while each virtual world has its benefits and drawbacks, the range of applications and requirements means that each has at least one (and possibly many) context for which it is well-suited. De Freitas, in analyzing these case studies, identifies several opportunities for further development, including blending real and virtual experiences, involving learners in constructing spaces and content, and emphasizing exploratory learning as an integral part of the educational landscape.

A few researchers have addressed the challenges that result from viewing virtual worlds as a development platform. One project, by D. McDonald et al. [56], addressed the issue of integrating automated characters (“agents”) with varied interaction models in a virtual world. This research was motivated by the observation that, in creating a realistic scenario, restricting onself to a single interaction model would limit the kinds of characters that could be created, and the interactions that could occur. To overcome this limitation, it would be desirable to integrate agents with varied models, without worrying about the details of a particular model. To this end, the researchers defined five levels of abstraction that these models might use (perceptual, literal, semantic, interpreted, and narrative), and described a generic framework for the integration of agent models based on these abstraction levels. The researchers supported the utility of the framework by presenting a case study where it was used to integrate several agent models in a crowd-control scenario. For example, a "community leader" agent used a literal perspective to react to the user's actions, while an "agitator" agent used a semantic perspective to integrate domain knowledge into its responses. While this work is based on a context of creating realistic agents, it is relevant to my work in that it presents a model of inter-personal interaction that is robust and multi-layered.

Dini et al. investigated the challenges that would be faced by a designer seeking to create a planning and execution system for a virtual world [22]. These planning systems could be used in a range of contexts, including the enactment of virtual world-based scenarios. The authors identified a number of problem areas that had previously been analyzed when investigating the creation of a planning and execution system for non-virtual world contexts. These areas included partial observability (that is, the system's inability to see the entire environment), sensing and reporting environmental data, authorability (providing tools that enable users to create new content), and dealing with role-based constraints. In this last area, the authors highlighted very broad, believability-focused constraints, such as prohibiting someone with a "human" role from jumping over a building. For each of these areas, the authors discussed the ways that the issue could be addressed in a virtual world. While the authors did not present any formal models, the identification of a set of relevant problems is useful in that it can be used to ensure that a proposed model or framework is sufficiently robust, and deals with the identified concerns appropriately.

Both McDonald and Dini share my interest in investigating the viability of virtual worlds as a platform for game development – an interest which, in my case, is focused on educational or "serious" games – and have investigated issues from a platform-independent viewpoint. However, they have focused on specific challenges related to game implementation, while I am interested in developing a development framework for serious games.

In a related project, Marks et al. investigated the suitability of several game engines for use in medical simulation [53]. In this study, the authors evaluated three game engines (Unreal Engine 2, id Tech 4, and Source Engine) according to a range of criteria including editing capabilities, content creation, and gameplay. The authors found that all of the game engines were able to "fulfill a great set of the features necessary for building a simulated surgical training application" [53], including graphics and audio, multiuser interaction, and the simulation of elements based on basic physics (such as soft tissue interaction). They did, however, note that the engines were limited by their ability to simulate elements using more complex physics, and platform-specific file formats posed some challenges. While the game platforms analyzed in this research are not virtual worlds, per se, the authors are concerned with many similar issues, and the questions they ask should also be considered in evaluating virtual world platforms.

Finally, Mikropolous et al. recently conducted a ten-year survey of educational virtual environments [60], analyzing a total of 53 projects from 1999 to 2009. While this study included both VR-based systems (using wands, gloves, CAVE environments, etc.) and virtual worlds, it found that the majority of projects used desktop-based systems (including virtual worlds), likely because of their low cost as compared to VR-based systems. In this study, several interesting trends were noted. First, all but one of the projects used a constructivist approach (either implicitly or explicitly), which provides strong evidence for the appropriateness of this pedagogical approach in conducting virtual world-based training. Second, 40 of the 53 projects focused on science, technology and mathematics, although there are also projects that deal with history and culture. The authors explain this trend by observing that constructing mental models in science and mathematics can be difficult; since a virtual environment provides extensive visualization capabilities, this is a natural approach for these subject areas. Some other common characteristics of the projects include simulation of real-world, authentic tasks; multiple viewpoints and representations, which help students construct knowledge; collaboration among participants, which enables collaborative knowledge construction; and an opportunity for reflection, which is also an important part of the constructivist learning process. Most promisingly, most of the studies reported positive learning outcomes and continued student engagement, and some also reported positive student and teacher attitudes towards the virtual environments. Mikropolous concludes with the assessment that, currently, "virtual reality is a mature technology appropriate for pedagogical use." [60].

There is strong evidence for the appropriateness of the constructivist approach in conducting virtual world-based training. [60]

In conclusion, researchers have found a number of platforms that are suitable for simulation-based training, with some being more appropriate than others for certain kinds of training (high-fidelity simulations, for example, or collaborative exercises). In terms of my research, Second Life (along with, to a lesser extent, its open-source counterpart, OpenSim) appears to be the best platform for the kind of training I'm supporting, since it is mature, widely-adopted, and suitable for simulations involving multiple participants. It should be noted that OpenSim is appealing from a privacy standpoint, since virtual world servers can be deployed on private networks, rather than relying on the servers provided by Linden Labs, the company behind Second Life. However, Second Life provides superior communication capabilities (in terms of its voice over IP component), a much broader selection of user-generated content, and its client software and servers are more reliable, overall.

2.2.2 Virtual World Projects

Researchers have developed a number of virtual world-based training projects, spanning a variety of disciplines and using a wide range of technologies. These studies are summarized in Table 2.4, and described in the following paragraphs. The projects are sorted using two criteria, chosen according to the factors that are most important in my research. First, I have used the *training type* criterion, which indicates the educational strategy used by a given project. To simplify analysis, I have grouped these strategies into three categories: *meeting*, where the main goal of the project is to encourage students to meet and collaborate informally; *skill training*, where the goal is to enable students to learn a specific skill; and *simulation*, where the project enables students to experience one or more situations through a realistic re-creation. Within each strategy, I have sorted the projects according to the type of evaluation that was conducted. Some projects were not evaluated at all, and were conducted to prove a basic level of feasibility. Other projects were evaluated using student feedback which, while informative, does not necessarily relate to educational outcomes, nor is it easily analyzable. Others were evaluated through usability or feasibility studies, which provide a slightly more rigorous assessment of the viability of the project. Moving closer to the kind of validation that is relevant to my research, several projects reported qualitative educational outcomes, such as “improved motivation”, or other qualititave analysis of the project. These outcomes help validate the utility of virtual worlds for education, although they are not easily quantifiable. Finally, there are a handful of projects that have been rigorously evaluated, and provide some quantitative measure of their effectiveness, usually through test results. These projects provide the strongest evidence of the effectiveness of virtual world-based training.

Table 2.4: Summary of Virtual World-based Training Projects

Author	Learning Objective	VW	Evaluation Method	Findings
Meeting Spaces				
Buchanan (2009) [6]	Medical seminars	Second Life	None	None
Dickey (2005) [21]	Software for business	Active Worlds	Informal student feedback	Increased sense of immersion, lower attrition rate
Monahan et al. (2008) [61]	Social space for students	CLEV-R	Usability study	Intuitive interface, effective environment
De Lucia et al. (2009) [18]	Social space for students	Second Life	Educational outcomes	Improved student motivation, good support for social interaction
Petrakou (2010) [66]	Second language	Second Life	Observation analysis	Beneficial opportunities for conversation, some user interface challenges
Skill Training				
Walker (2009) [85]	Counselling	Second Life	Student feedback	Very positive experiences reported, some technical problems
Koenraad (2008) [49]	Second language	ActiveWorlds	Self-reported outcomes	Students felt confident in conversational skills, teachers reported increased student motivation
Vergara et al. (2008) [81]	Hematoma treatment	Custom VW	Educational outcomes	Equally effective as conventional education methods, offered chance for remote collaboration
Simulation				
McDonald (2009) [55]	Surgery	Custom VW	None	None
Nakanishi et al. (2004) [62]	Evacuation procedures	FreeWalk	None	None
Getchell et al. (2010) [25]	Archaeological dig procedures	Second Life	User feedback	Positive reception from students
Taylor et al. (2011) [77]	Paramedic	Second Life	Feasibility testing	Initial indications for potential of proposed system
Greci et al. (2010) [32]	Crisis response	OpenSim	Educational outcomes	Improved team communication and decision-making skills
Dev et al. (2007) [19]	Crisis response	Varied	Test performance	Comparison of pre- and post-simulation quiz scores showed significant improvement
Hudson et al. (2009) [41]	Border guard	Second Life	Test performance	28% increase in grades for relevant course component

2.2.3 Meeting Spaces

One theme in virtual world-based education projects is the use of the virtual world as a "meeting space" for students. In these projects, the researchers focused on the social and communication aspects of the experience, rather than the interactive, task-based approach that I am interested in with my work. A few such projects are described below, to give a sense of the work in this area.

The Ann Myers Medical Centre, supported by Sprott-Shaw Community College in British Columbia, focused on providing a well-established meeting place for medical educators and students, in order to facilitate educational sessions in a virtual environment [6]. Recent sessions have covered topics such as molecular oncology, post-traumatic stress disorder, H1N1, and osteosarcoma. The Ann Myers Medical Centre meeting space included some medical equipment and screens for showing presentation slides and medical images. While the goal of the project, providing training in a virtual world, was similar to that of MeRiTS, this project seeks to reach that goal through lectures and presentations, rather than process simulation. The project was closed in March, 2011, when the virtual space used for the Ann Myers Medical Centre came under new ownership, and the project's support was discontinued.

BCOR is a virtual space created in ActiveWorlds to support the delivery of an introductory Business Computing Skills course at the University of Colorado, Boulder [21]. In this space, students are able to view their assignments, access resources for their assignments, and collaborate with other students, as appropriate. The space is designed around a central plaza, with a road for each topic area (e.g., "Microsoft Word Lane") leading from the plaza to a building containing the relevant assignment information and online resources. At the back of each building is a patio area where students can collaborate on group assignments, and access resources to support these assignments. Finally, the students can view additional course information and submit assignments via an integrated web browser. While the validation of this project was quite informal, consisting of feedback from students who took part in the course between 1998 and 2000, students reported that the environment made them "feel like they were at school", and the course designers reported that, after the addition of BCOR and other online tools, the course became more popular, and the attrition rate dropped [21].

CLEV-R is a meeting space-based project by Monahan et al. at the University College Dublin [61]. In this project, researchers have created a custom, web-based virtual environment which includes a classroom, meeting room, and library. These meeting spaces, in addition to providing a common virtual place for students and teachers to interact, also provides users with the ability to upload lecture slides, take notes, access online resources, and use a web-accessible message board. One unique feature of this system is that, by leveraging the web-based implementation, the researchers were able to develop a version of the system for mobile devices (mCLEV-R), thus enabling "any time, anywhere" access to the virtual environment. The researchers conducted an initial usability study, which required students to complete a set of tasks individually, and then give a brief presentation to the other students as part of a collaborative discussion. The results of the study were very positive: 89% of the respondents found the navigation controls intuitive, 100% felt the lecture experience was effective, and 89% felt the environment provided an effective means of social interaction.

The SecondDMI project, undertaken by De Lucia et al. at the University of Salerno [18]. In this project, a virtual campus was created for undergraduate students, which included a common campus area, meeting spaces, lecture rooms, and recreational areas. These areas were also linked to Moodle, an open-source learning management system (LMS), to provide relevant educational content. To take full advantage of the resources available via Moodle, the researchers developed a plug-in, Multimedia Moodle, to deliver streaming videos to the students. To evaluate the SecondDMI project, the researchers asked the students to complete detailed questionnaires after using the virtual campus for three remotely-delivered lectures, and found that the environment supported synchronous communication and social interaction, as well as improving student motivation.

In a similar vein, work by Petrakou explored using Second Life for online education, and sought to identify interactivity issues through a virtual world-based class exercise [66]. By bringing a class of English language students in Sweden together with students from the United States, the project provided the Swedish students with practice conversing with native English speakers, while at the same time fulfilling course requirements for the American students. Petrakou conducted a detailed analysis of observational notes and feedback from instructors and students to determine the benefits and drawbacks of virtual world-based online education. She found that the students benefited from the ability to converse synchronously with others within a common virtual space, and were able to participate in “a diversity of activities, experiences, and interactions”. However, this diversity also presented technical and user interface challenges, which led the author to conclude that using the virtual world is, in itself, a collaborative learning activity, and instructors must ensure that students are given sufficient training in this environment before having the students focus on course content.

> *Students were, through the virtual environment, able to participate in a diversity of activities, experiences, and interactions.* [66]

In summary, virtual worlds have been used by researchers to provide students from a wide range of disciplines with educationally-focused meeting spaces. In evaluating these projects, researchers have found various positive outcomes, including increased motivation, support for meaningful social interaction, and an opportunity for a broader range of social experiences. However, the use of a virtual world also requires additional training for students and instructors alike. While the interface has been shown to be intuitive, facilitators must ensure that users are indeed comfortable with the virtual world before asking them to engage in any educational activities.

2.2.4 Skill Training

Some researchers have used virtual worlds for task-based training projects. These are described in detail below.

Walker created a Second Life simulation to help distance education students develop counselling skills [85]. Motivated by a desire to model counselling skills and provide immediate feedback to distance education students, she created a counselling facililty, which contained labs complete with couches, tissue boxes, and one-way mirrors for observation. The students conducted several counselling sessions in the virtual lab, and afterwards were asked to fill out perceived learning and attitudinal surveys. The results of the surveys indicated that using Second Life did not detract from the learning experience, although students who self-reported as very computer illiterate had difficulty getting comfortable with the virtual world, and students who reported learning the most also reported high degrees of affective (emotionally-related) learning. While some students encountered problems with graphics cards, internet connections, or other technology issues, other students had very positive experiences. One student reported that "actually seeing a counseling room and people when you were practicing your counseling techniques" was very beneficial, while another stated that "the ability to gesture, communicate, and to take on a character and also the stimulating clinical setting were very helpful and have great potential" [85].

In a similar effort, Koenraad created a virtual "Language Village" in ActiveWorlds to help teach French to high school students [49]. The idea of a Language Village, where students would visit simulated locations and practice their vocabulary and conversational skills, was already present in the curriculum, but was typically implemented as either an on-site event or a mobile resource where student teachers and props visited various schools. In this project, a virtual Language Village was developed which included a variety of simulated locations, such as a police station and bakery. While the focus of the project was on feasibility, rather than effectiveness, there were several positive outcomes reported. Half of the students reported learning more than they would have in a traditional classroom setting, and felt confident in their conversational skills. Teachers, meanwhile, reported that the students were enthusiastic, showed increased motivation, and spent more time on task.

> *Teachers reported that students were enthusiastic, showed increased motivation, and spent more time on task.* [49]

Vergara and colleagues at the University of New Mexico have developed a virtual environment-based tool to teach medical students about hematomas [81]. They have developed a 3-D, multi-user virtual environment (MUVE) within which students can interact with a virtual character, nicknamed "Mr. Toma," and other associated objects. A rigorous study of the system's effectiveness was conducted, comparing the effectiveness of MUVE-based training to a more conventional, paper-and-pencil method, in both in-person and distributed settings. Within each experiment condition (e.g, MUVE-based, in person), pairs of students were asked to complete problem solving and patient management tasks related to treating a epidural hematoma in a patient following an automobile accident. The students' level of knowledge was assessed via a test of medical concepts, which was administered to each student before and after the training session. This study demonstrated that the MUVE-based training session was equally effective as the conventional, paper-and-pencil method. Furthermore, the MUVE-based approach offers additional advantages, including the chance to collaborate with geographically dispersed students, and an increased sense of immersion while using the system. A considerable amount of effort was put into ensuring that the content was presented accurately and effectively, including consulting with an interdisciplinary team of subject matter experts. While the project used a custom virtual environment and was focused on a single context, thus severly limiting the project's extensibility, the educational validation presented offers strong motivation for this kind of project, and for the benefits of virtual world-based training.

2.2.5 Simulation

Within the last few years, researchers have begun using virtual worlds for simulation-based projects. These are the most relevant to my research, and are described in some detail below.

One such project is Pulse!!, a virtual medical education project led by McDonald at Texas A&M University, Corpus Christi. The centrepiece of this project is the Virtual

Clinical Learning Lab, which is “an interactive virtual environment simulating operational health-care facilities, procedures and systems” [55]. The developers of the system put a significant emphasis on the accuracy of the simulated environment, ensuring that details such as the posters on the office walls were authentic. See Figure 2.2 for a screenshot of the environment. While the system was initially focused on a naval hospital setting – the majority of the project’s $14.7 million in funding is from the US Office of Naval Research – it has since been tested in the Yale School of Medicine and Johns Hopkins School of Medicine, and has licensed the technology to BreakAway Ltd., for commercial development. Unfortunately, there has been little information released recently on this project, so the results of the aforementioned field tests are not known, neither is it clear whether the project is still ongoing. However, in terms of the relationship between environment accuracy and user immersion, the project represents a clear exemplar of an “increased accuracy will provide improved immersion” methodology.

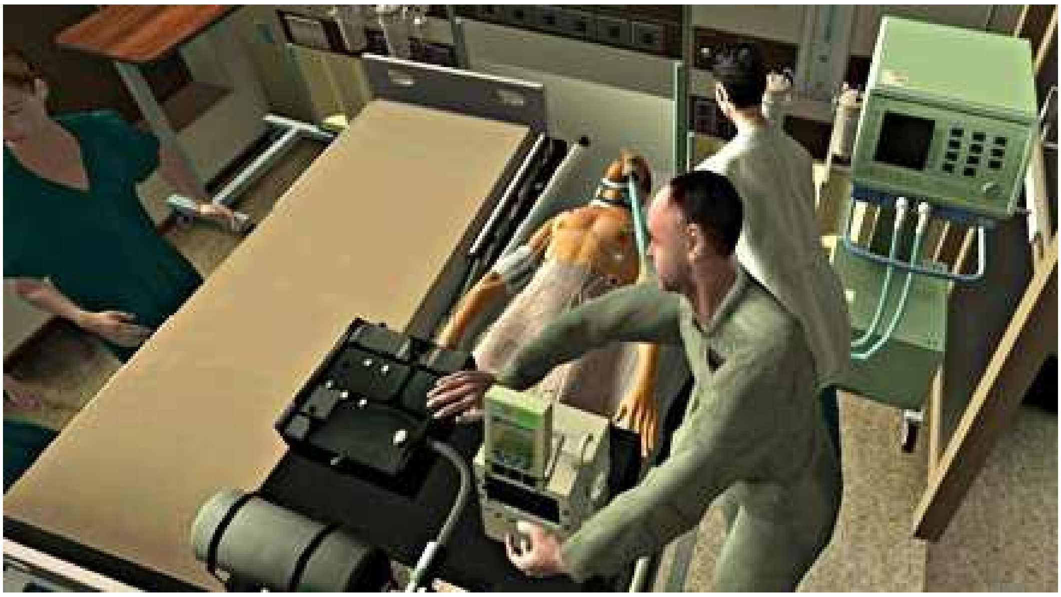


Figure 2.2: Pulse!! scenario (from [55])

FreeWalk/Q represents an early effort at creating a virtual meeting space by Nakanishi et al. at Kyoto University [62]. In this project, the researchers created FreeWalk, a custom virtual platform intended to “integrate diverse technologies related to virtual social interaction [such as] virtual environments, visual simulations and lifelike characters” [62]. They envisioned the platform as enabling a range of virtual social situations, including meetings, training, and shopping. Although much of their description is concerned with the implementation details of the system (e.g., implementing walking, turning and speaking actions), they also describe the benefits of Q, an interaction scenario description language. While this language is discussed in greater detail in Section 2.3.2, the language essentially allows the user to define a finite state machine consisting of perceptions and resulting actions within a particular scene. The researchers also highlight the potential for FreeWalk/Q as a training tool, describing a simulation of a crowd evacuation situation, which involves both human-controlled and automated characters.

Getchell et al. have developed a hybrid web-based learning/virtual world system for teaching students about archaeological fieldwork [25]. This system, called the Laconia

Acropolis Virtual Archaeology (LAVA) project, combines web-based project planning and management tools with a virtual world representation of a real-world dig site to provide students with a comprehensive simulation covering the process from writing a funding application and visiting the site, to performing the excavation and restoring the site for exhibition. The LAVA system addresses several challenges inherent in archaeological education: the inaccessibility of archaeological sites, both in terms of expense and geographical distance; the destructive nature of excavation, which restricts the number of students who can participate at one time; and the fact that any mistakes made are likely to be irrecoverable, which necessarily limits the degree to which students are allowed to participate in a real-world excavation. The first several phases of the process – writing a proposal, conducting an initial site visit, writing a funding application, and allocating a budget – are primarily conducted using web-based project management tools, although the initial site visit occurs in Second Life. The next major phase is conducting an excavation of the site, which is implemented using a 2-D web interface. The final phases of the process – visiting a reconstruction of the site and creating an exhibition in the visitor's centre – are conducted in Second Life. The system's archictecture links the content stored in the web-based system with the simulated site in Second Life, although it is not clear how (or if) this content actually has any effect on the simulated site. The excavation phase is supported by an excavation model, which is used to calculate the probability of a user (or team) finding an item at any point during the dig. However, since this phase is implemented using the 2-D web interface, it is not relevant to my research. Finally, the authors describe a user evaluation study that they conducted, in which they report a positive reception from students and a willingness on the part of educators to embed LAVA into the curriculum, and to develop additional scenarios.

Taylor et al., at the Imperial College in London, have developed a framework for creating both single-user and multi-user virtual patient exercises [77]. Using this framework, his team has begun to develop several scenarios: one that trains paramedics in responding to a situation involving hazardous materials, another that focuses on staff in a hospital emergency department, and a third that involves multiple agencies in a mass casualty situation. These scenarios are all supported by clinical decision trees, which model the behaviour of associated virtual patients. While the implementation of scenarios based on the framework is (as of February, 2011) at a very early stage – testing the validity of the virtual patients – the authors emphasize the framework's potential for use in creating a wide variety of virtual world-based scenarios. It should be noted, however, that the logic embedded in the framework is limited to specifying the behaviour of virtual patients. More complex scenario behaviour (interaction among entities, for example) must be specified outside of the framework, likely within the virtual world. Also, the early stage of scenario development limits the extent to which the platform can be empirically evaluated, and the educational assessment that can be done on the scenarios.

Greci et al. have also investigated using virtual worlds for emergency training [32]. In their research, they focused on preparing emergency department nurses to deal with patient surges in the event of a flu pandemic. Specifically, they trained the nurses in surge management practices through drills conducted in a virtual world, and also practiced a team-based approach to patient triage during a surge. The exercises were developed by an interdisciplinary team of researchers, educators and clinicians at UC San Diego and nearby VA hospitals [31]. To help evaluate the students' performance, the instructors used *machinimas* (digital videos of the virtual world) taken from various perspectives – a nurse manager in the Emergency Department, for instance, or a bird's eye view of the Command

Post. These machinimas were then shown to the students in instructor-guided reflective learning sessions. The reflective learning sessions, in turn, informed discussions that helped the students identify strengths, weaknesses, and recommended corrective actions. This process of creating machinimas for reflective learning was an important influence on the recording and analysis components of my framework, and in particular the trace-based video annotation component. The researchers found that these training exercises resulted in improved team communication and decision-making skills, and also improved the students' visual debriefing skills.

> *The virtual world-based training exercises resulted in improved team communication and decision-making skills, and also improved the students' visual debriefing skills.* [31]

Dev et al. have been active in developing simulations for a number of years, with a particular focus on virtual worlds and medical education [19], [38]. Dev has led projects that train students in anaesthesia crisis resource management, emergency medicine crisis resource management, and emergency department triage techniques in a situation with mass casualties. Some of these projects have used commercial VW systems, such as OLIVE and Second Life, while others used a web-based viewer developed by researchers at Stanford University. In the most relevant of these projects, Dev focused on assessing the educational effectiveness of an emergency medicine simulation. She had students go through a pre-training case to assess their baseline competency, then had them go through four training cases in the virtual world, and finally gave the students a post-training case to assess the students' improvement. For the pre- and post-training test cases, the students were observed by several evaluators, and evaluated (on a five-point scale) using a dozen measures of team performance, such as *knowledge of environment*, *communication with other team members*, and *delegation of responsibility*. Through comparing the results of evaluating the students' performance on the pre- and post-training cases, Dev found that the students improved by a significant margin after going through the VW-based training cases.

Working with an entirely different student population, Wrzesien et al. created a virtual environment for elementary school students to learn about aquatic life in the Mediterranean [89]. In this project, students interacted with an instructor, represented as a fish, while exploring a virtual representation of the marine environment. To test the virtual world's effectiveness, the researchers created virtual and traditional versions of an educational module, using the same set of learning objectives and educational content. A class of students was then assigned to either the virtual or traditional modules, and their learning and experience were assessed, via pre- and post-module tests and questionnaires, respectively. There were no significant differences in test performance between the two groups, indicating that the virtual world is equally effective, educationally. Moreover, the researchers found that the students in the virtual group enjoyed the experience more, and were more engaged in the experience.

The Canadian Border Simulation project, led by Hudson at Loyalist College, is a very good example of a practical and successful virtual world-based training program [41]. This project used a virtual border crossing to augment the in-class role-playing experience that students receive, and replace field placement opportunities which had previously been an integral part of the students' training, but were cancelled as a result of post-9/11 security concerns. Through this virtual border crossing, students were able to gain experience with a wide variety of traveler interview scenarios. The project is notable for the limited time and resources that were used in the creation of the simulation (as compared to a project like Pulse!!). Specifically, the project was developed over a period of three months, and was "only realizable with the affordable simulation tools that Second Life provides". Within these constraints, the developers (faculty, an active border guard, and 3-D designers) focused on ensuring sufficient accuracy in key aspects of the scenario, such as the border crossing, the vehicles that were to be searched, and the passengers' travel documents. See Figure 2.3 for a screenshot of the scenario. The instructors also gave careful consideration to the integration of the simulation within the broader context of the course curriculum, and ensuring that students were given sufficient training in Second Life before they participated in the simulation. The result of this planning was a scenario experience that helped improve student performance (a 28% increase in grades for the interview skills component, as compared to the previous year's class) and increased enthusiasm for the course. It should be noted, however, that the virtual objects and associated code was created specifically for the border crossing context, and is not extendable to other contexts or training situations.

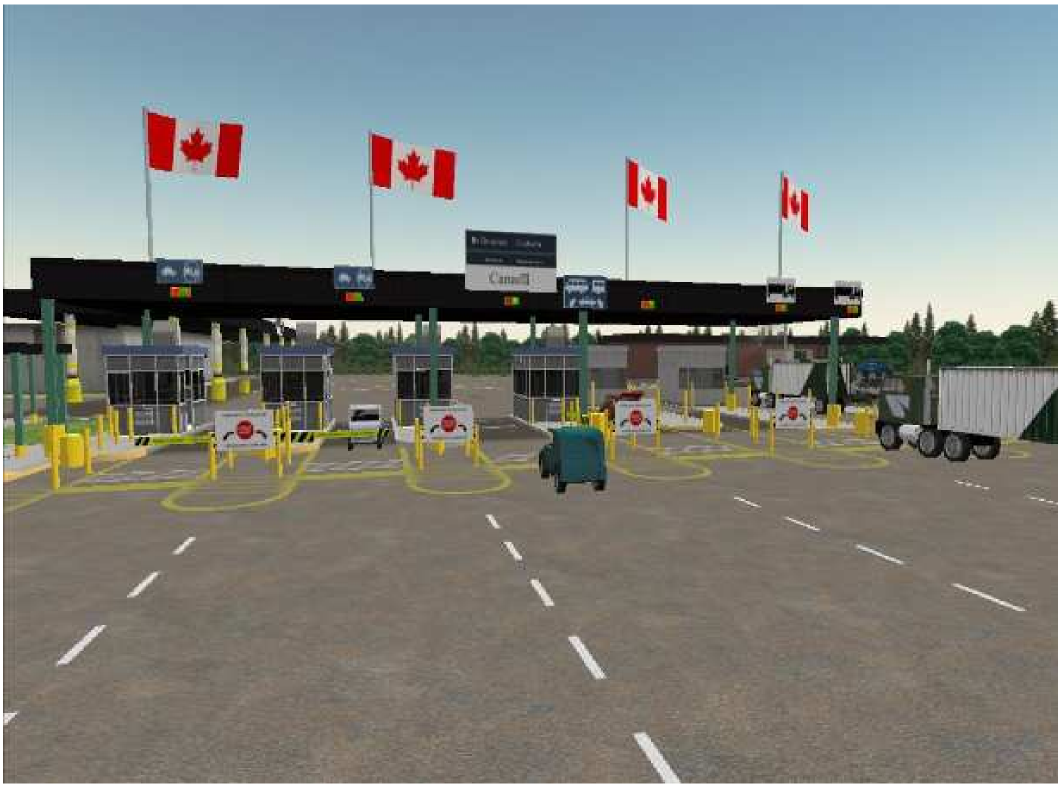


Figure 2.3: Canadian Border Simulation (from [41])

2.3 Behaviour Modelling and Analysis

A crucial requirement underlying much of my research is the need to represent, record, and analyze user behaviour in a virtual world. By meeting this requirement, my framework is able to not only present users with experiences, but ensure that these experiences are educationally meaningful and measurable. To support this goal, I have investigated a wide range of research on behaviour modelling and analysis. The first part of this requirement is the ability to understand and accurately represent a user's actions in a virtual world. Motivated by the scenario-based nature of my framework, I have investigated process modelling techniques, to ensure that the scenario being presented is modelled appropriately. Second, because of my interest in modelling user behaviour in a platform-independent manner, I have investigated work on avatar modelling. This research offers a variety of perspectives on the generic representation of avatar behaviour, ranging from avatar appearance to synchronizing avatar information across multiple virtual worlds.

Meanwhile, because of the need to validate students' educational experience when participating in virtual world training scenarios, I am interested in work related to action recording and analysis, particularly within a virtual world context. The question of action recording has been touched on, in a general way, by some researchers. In one case, researchers studying students' behaviour in a 3-D game environment noted a discrepancy between the students' self-reported behaviour and their actions as logged by the game [37]. Specifically, the students saw themselves as having consistently used a team-based problem solving approach, while the logs indicated that they initially attempted to solve the problems on their own, and only asked for help from other team members when they got stuck. This observation highlights the importance of action recording as compared to self-reported action evaluation methods, such as surveys and interviews. In more directly relevant work, Hurst identified the challenge of processing and understanding large quantities of log data as an obstacle in assessing whether virtual reality training transfers to the real world [42]. In order to better understand this issue, Hurst categorized the action recording data into three types: reflective data, machinima, and virtual environment data. The first category consists of standard participant feedback mechanisms such as questionnaires and interviews. The second category, machinima, deals with recorded in-world video, which is created using software such as Camtasia. Virtual environment data, finally, covers the action recording challenge that is discussed in this section. In this category, Hurst suggests capturing avatar actions and interactions with objects. While this categorization scheme, and particularly the third category, helped inform the student action recording and analysis component of my research, Hurst did not actually create any recording tools using the scheme. Her research, therefore, serves as a conceptual starting point for a piece of my work.

With regards to research focused on the task of actually recording actions, relevant work was found in two communities: the gaming community and educational psychology. From gaming researchers, I found literature on modelling processes and analyzing observed actions in order to derive more complex behaviours. From the educational psychology community, meanwhile, I found relevant research on identifying the student actions that are relevant to the learning process.

2.3.1 Identifying Relevant Actions

In addition to the broader pedagogical theories about learning, described in Section 2.1, I have also drawn on educational psychology research to determine which types of student actions are relevant, from an educational point of view, within the context of a virtual world simulation. I used this set of actions for guidance when determining which actions needed to be included in the ABM. The specific actions will, of course, vary from one educational program to another, but there are certain *kinds* of actions that students should engage in, regardless of the particular program being considered.

One interesting analysis of education in virtual worlds comes from Girvan et al. [26]. They conducted a case study from a communal constructivism basis (which was proposed by Scardamalia et al. [70]), with the goal of determining whether learning had, in fact, occurred. In this case study, they brought together small groups of educators, and had them learn about a new topic (the international banana industry) and create a virtual artifact (in this case, a book) summarizing what they learned. To assess this, they conducted semi-structured interviews, and performed an analysis of the chat logs from the educational sessions, as well as analyzing the virtual books. While this work does not touch on virtual world recording, specifically, its focus on using an established educational framework to evaluate learning in a virtual world is relevant to my research, as I am also interested in rigorously evaluating the learning that occurs within a virtual world-based educational context.

Land et al., in an effort to systematize the development of learning environments, proposed the use of grounded design, which blends elements of constructivism with an emphasis on real-world validation and iterative improvement [52]. The authors present some of the key ideas of grounded design, including a strong connection between system design and supporting educational theories and frameworks, and ensuring that the designs can be generalized, in order to ensure their applicability to multiple contexts. They also present several examples which embody these principles, and thus motivate the real-world applicability of grounded theory in developing learning environments. Again, while the authors don't address action recording, per se, they do discuss many issues that are relevant for recording educational experiences in a virtual world.

The most directly relevant research comes from Price, who wrote about creating a theoretical basis for the development of educational games [68]. In this work, Price established links between educational theory concepts, such as active experimentation, collaborative learning, and concept maps, and immersive education technologies, such as heads-up displays, voice chatting, and programmed non-player characters (NPCs). For example, one way in which the immersive environment can be used to convey a concept from educational theory is by converting a concept map into a network of rooms, which may be explored in a manner analogous to a student progressing from one topic to another. Similarly, Kolb's four-stage learning cycle, proposed as part of his Experiential Learning Theory [50], can be instantiated in an immersive environment as a series of four rooms, which take the learner from concrete experience to reflective experience, then abstract conceptualisation, and finally to active experimentation. Price also mentioned the creation of a *history* actor, which recorded various characteristics of a user's interaction with the immersive environment. Price also discusses collaborative learning – including theories on discourse and collaborative learning from Candlin and Johnson et al., respectively – and as being crucial for informing communication-based learning in immersive environments [9], [45].

> *Price established links between educational theory concepts, such as collaborative learning, and immersive education technologies.* [68]

In another practical exploration of the connection between educational theories and game elements, Marty et al. have, in the course of developing a game-based learning (GBL) environment, defined an explicit mapping from Activity Theory concepts to entities in the GBL environment [54]. For instance, Marty maps the the idea of a *link between activities* to a *corridor* within the GBL environment. In creating this mapping, Marty provides a means for educators to create appealing learning experiences which are strongly connected to pedagogically-based lessons and curricula. Furthermore, Marty's definition of this mapping provides further support for the applicability of educational theories to game-based educational methods, such as the virtual world-based training scenarios supported by my framework.

To sum up, there are several educational theories, such as communal constructivism, experiential learning theory, and collaborative learning, which are directly relevant to the issue of logging actions in the virtual world. Based on research grounded in these theories, I propose that the following core set of behaviours, shown in Table 2.5, should be captured by a logger for use within an educationally-focused virtual environment.

Table 2.5: Core Activities to Record

Behaviour	**Relevant Educational Theories**
Movement	Concept maps, learning cycles (Kolb [50])
Talking	Educational co-creation (Messinger [59]), collaborative learning (Johnson [45]), discourse (Candlin [9])
Gestures	Non-verbal communication (Mehrabian [58]), discourse (Candlin [9])
Entity creation	Communal Constructivism (Scardamalia [70])
Entity interaction	Cognitive apprenticeship (Collins [14])

2.3.2 Process Modelling Techniques

In choosing appropriate process modelling techniques for my research, I drew on both technical resources – workflow specification and modelling languages – as well as gaming and business-based methods of process analysis. Research from these perspectives are described in this section.

Game-based Process Modelling

A general approach to process modelling, which has been adopted by members of the gaming community, is generative design patterns (GDPs). In this approach, a structured set of design patterns is used to generate specific instances of the behaviour being modelled [57]. In work conducted by McNaughton et al. at the University of Alberta, the ScriptEase system was developed to help game developers apply GDP methods to computer role-playing games (CRPGs), through integration with the CRPG Neverwinter Nights (NWN) [57]. Within the context of NWN (and CRPGs in general), the authors of ScriptEase defined four basic GDP types: encounter, behaviour, dialog and plot patterns. Each of these patterns encapsulates a particular aspect of the CRPG gameplay experience and, in principle, the set of four patterns should cover the entire set of possible CRPG behaviours. ScriptEase is a GUI-based system that enables users to define CRPG behaviours in terms of the four basic patterns, with the ability to adapt each pattern as necessary in order to create the desired behaviour. Note that the user's adaptation capabilities range from simple pattern instantiation to creating new patterns and atomic actions for use when dealing with custom behaviours. The original version of ScriptEase then, in the last step of the process, translates this pattern-based description into an executable script in NWScript, the C-like language used by interactive objects in NWN. A more recent version of the system is able, through XML-based "translators", to convert the patterns for a given script to a target language [12].

While the ScriptEase project shares several common goals with my research, there are several important distinctions. First, the original ScriptEase project is platform-specific, since it converts GDP-based behaviour descriptions to NWScript, a language that is used only in NWN. This limitation, as previously mentioned, has recently been lifted through the use of ScriptEase translators, although the amount of effort required to create these translators is unclear. Second, and more importantly, the four basic GDP types are based on a CRPG-centric approach to behaviour modelling. That is, the four design pattern types are defined to encompass a CRPG-oriented set of behaviours, and may do a poor job of representing behaviours outside of this scope. Finally, as implied by the platform-specific nature of the ScriptEase system, the behaviours modelled using ScriptEase can only be executed through the NWN platform. Because of this limitation, the behaviour descriptions generated by ScriptEase are not suitable for use in a modular, component-based framework such as the one I have developed.

> *The core design pattern types used by ScriptEase are defined to encompass a CRPG-oriented set of behaviours, and may do a poor job of representing behaviours outside of this scope.*

Another type of game-based process modelling is case-based planning (CBP), which is particularly applicable to real-time strategy (RTS) games. In one CBP project, Darmok, Sugandh et al. developed a system for playing WARGUS, an open-source version of the popular game Warcraft II [76]. This system works by first extracting a set of behaviours from a collection of traces of games played by an expert. Using these behaviours as a starting point, the system develops a set of initial plans (e.g., “win the game”), which it uses to start playing the game. Then, as the game progresses, the system executes, adapts, and expands upon these plans. See Figure 2.4 for a flowchart of this process. The Darmok system introduces two key innovations to the standard CBP approach. One is that the system removes redundant (or otherwise useless) actions from a plan through dependency graph analysis. Second, the system integrates the addition of new actions to a plan into the CBP cycle, thus avoiding the expensive search process that would otherwise be required.

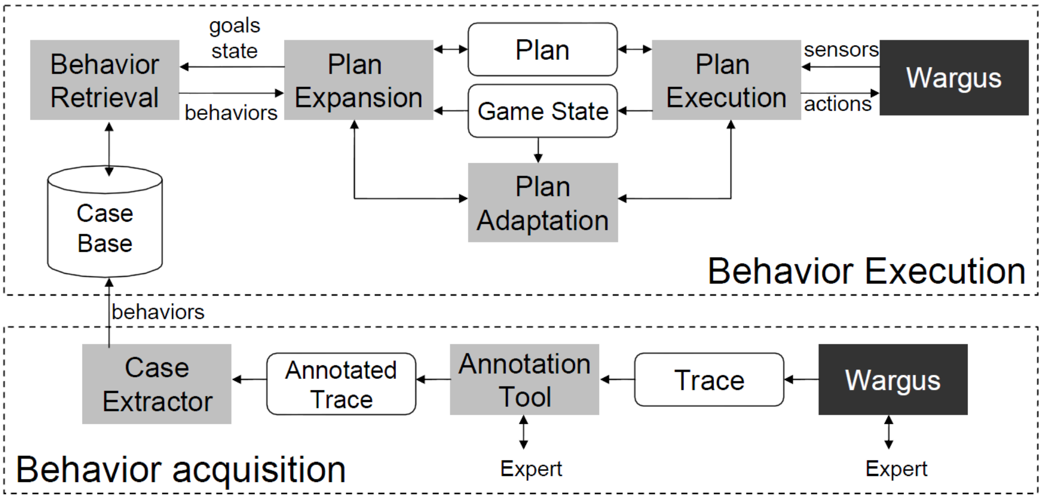


Figure 2.4: Flowchart of Darmok system (from [76])

This research, like ours, is concerned with plan specification and execution. In this case, the plans are made up of *behaviours*, which are specified using the following structure:

Behaviour = <*Declaration*, *Procedure* >
  Declaration = <*Goals*, *Preconditions*, *Alive conditions* >
    Goals = The intended goal for the behaviour
    Preconditions = Conditions that must be met before behaviour can occur
    Alive conditions = Conditions that must be met for behaviour to continue
  Procedure = <*Action type*, *Action*, *Subgoals* >
    Action type = <*Sequence*, *Parallel* >
      Sequence = The encapsulated actions must be executed sequentially
      Parallel = The encapsulated actions can be executed in parallel
    Action = A basic in-game action
    Subgoals = <*Parameters*, *Success conditions* >
      Parameters = Parameters that define this subgoal
      Success conditions = Logical conditions that may be checked to determine whether the subgoal has been met

However, the plans that are specified by the system are, by definition, goal-oriented, and require a constantly responsive and dynamic game environment in which to execute these plans. Moreover, the plans are intended to be executed by an automated agent, and are thus imperative, rather than descriptive, in nature. That is, the plan is defined in order to provide instructions to an otherwise inactive (automated) agent, rather than guiding or responding to the actions of an independent (human) actor.

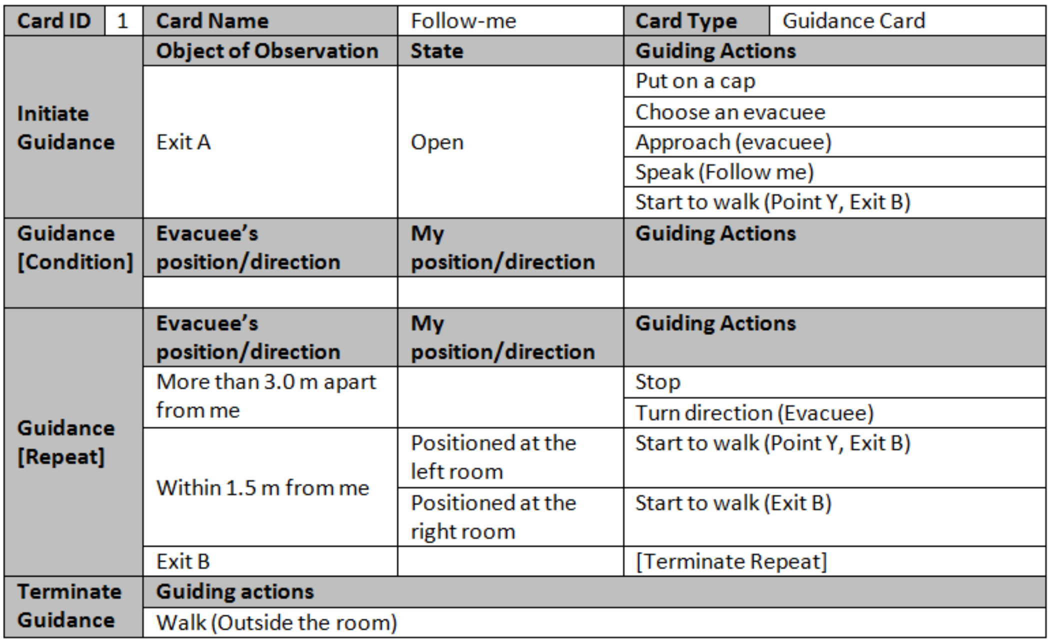
Q

Q is a scenario definition language that was developed by Nakanishi et al. to specify the behaviour of agents within the FreeWalk virtual world [62], which is described in Section 2.2.5. Q is an extension of Scheme (and, thus, of Lisp) which models scenarios, using an event-driven approach, with *perceptions*, *actions*, and *scenes*. A perception is an event that has been perceived by an agent, and then triggers one or more actions. These perception-action sets are then grouped into logical units called scenes. Transitions from one scene to another can, like other actions, be triggered by perceptions. A sample scenario (from [62]) is shown below.

```
(def scenario reception
   (scenel
      ((?hear “Hello” :from $x)
         (!speak “Hello” :to $x)
         (go scene2))
      ((?hear “Bye”)
         (go scene3)))
   (scene2
      ((?hear “Hello” :from $x)
        (!walk :to $x)
        (!speak “Yes, may I help you?” :to $x))
        (otherwise (go scene3)))
   (scene3 ...))
```

The Q language serves two key purposes: first, it specifies the behaviour of agents, in terms of the *perceptions* that an agent is capable of recognizing, the *actions* that the agent takes in response to the perceptions, and the *scenes* in which these perceptions and actions take place. Second, by defining the perceptions in a way that the agent can respond to them, the language is providing a means by which the user's actions in the virtual world can be parsed. These features of the Q language are crucial elements of my scenario enactment process and avatar behaviour model respectively, and thus Q plays an important role in informing my work.

In addition to the Q language, Nakanishi et al. have also introduced interaction pattern cards (IPCs), shown in Figure 2.5, which "capture the interaction patterns in each application, thereby providing a higher level of abstraction" [62]. These cards can be created with a spreadsheet, and then translated into Q statements.



Card ID 1	Card Name	Follow-me	Card Type Guidance Card
	Object of Observation	State	Guiding Actions
Initiate Guidance	Exit A	Open	Put on a cap
			Choose an evacuee
			Approach (evacuee)
			Speak (Follow me)
			Start to walk (Point Y, Exit B)
Guidance [Condition]	Evacuee's position/direction	My position/direction	Guiding Actions
Guidance [Repeat]	Evacuee's position/direction	My position/direction	Guiding Actions
	More than 3.0 m apart from me		Stop
			Turn direction (Evacuee)
	Within 1.5 m from me	Positioned at the left room	Start to walk (Point Y, Exit B)
		Positioned at the right room	Start to walk (Exit B)
	Exit B		[Terminate Repeat]
Terminate Guidance	Guiding actions		
	Walk (Outside the room)		

Figure 2.5: Example scenario defined using an IPC (from [62])

Once a scenario has been implemented in Q, it can be accessed by the FreeWalk system when a relevant cue is perceived, and the Q scenario then returns the resulting actions via the FreeWalk API. Specifically, Q calls the FreeWalk API and registers the action in a shared region of memory. FreeWalk then checks that memory region for any stored actions at the beginning of its next processing cycle, and executes any actions that are present.

Since Q is deployed within a research-oriented, custom virtual world, its wider applicability may initially appear quite limited. However, the structure of the Q language is quite general, and thus serves as an interesting basis for comparison to my own work, especially given its integration with a virtual world. While the Q language and my framework are based around several similar concepts – scenarios, scenes, actions, and perceptions – there are several elements of my framework that are not included in the Q/Freewalk system. One is the idea of *constraints*, or limitations on a particular action based on the current scenario state. Another is the ability to respond to system-generated events, which is included in my framework via the definition and execution of timed events.

2.3.3 Analyzing Observed Actions

There has been a significant contribution to research in this area from the games community, particularly in inferring complex behaviours from observed actions. This research is summarized in Table 2.6, and described in the following paragraphs.

Table 2.6: Summary of Action Observation Techniques

Technique	**Inferred Activities**	**Presentation Method**	**Audience**
Visualization based on action behaviour (Thawonmas [79])	chat, movement, fighting monsters, mission completion	cluster analysis	researchers
Temporal links (Greenhalgh [33])	movement, speech, interaction with objects	temporal links to previous activities	VW users
Activities in Unity engine (Raghavendra [69])	actions defined via *conditions*, *selection*, *events*	action history shown in-world or in 2-D map	game users
Actions in River City virtual world (Nelson [64])	relevant student activities (reading signs, talking to residents)	input for student guidance system	students
Pedagogical dungeon (Marty [54])	17 basic activities (e.g., chat, answer question), composite sequences	aura around user, additional armor	instructors

In one project, Thawonmas and Iizuka have proposed a system for visualizing players based on action behaviours [79]. In this system, they classify players as achievers, explorers, or socializers, and use a combination of mathematical analysis (classic multidimensional scaling) and text-based analysis (the KeyGraph method) to convert player behaviours into classifiable measurements.

Greenhalgh has proposed the use of explicit temporal links to augment the recording and replaying capabilities of a virtual world [33]. The main concept is that a virtual world, in addition to the standard content of avatars, terrain, entities, and the like, can contain a temporal link to a different time. This link can either refer to the current location, or to any other location in the virtual world. This feature increases the capabilities of the VW as a whole – the authors propose several creative uses of temporal links, including recorded tour guides, flashbacks of recent activity, and the storing and viewing of brief in-world messages. The authors recognize that the ability to implement temporal linking is dependent on being able to "capture all activities within a CVE [collaborative virtual environment], including ... movements, speech, and interactions with virtual objects"[34]. That is, this work represents an augmentation of an existing logging system, rather than a self-sufficient logging system in and of itself. However, it does present an interesting extension of the issue of logging in a virtual world.

Raghavendra has developed a tool to track user actions in games based on the Unity game engine [69]. This tool allows the user to define actions of interest using a generic language of *conditions*, *selection*, and *events*. Using the tool, the inherent complexity of monitoring "hundreds of objects, continually interacting with each other" is reduced to simply identifying a small set of relevant actions, and logging them accordingly. This action log can then be analyzed to create a map of the player's movement history, for instance, which can be shown either in-world or in a 2-D map.

Nelson and Ketelhut embedded a tracking system within a multi-user virtual environment, called River City, to record relevant student activities, such as reading signs, clicking on objects, asking questions of computer-controlled "town residents" [64]. These logs were then used to drive a student-guidance system, which provided hints to students in the form of reflective questions. This tracking system enabled the personalization of the student learning experience, based on his activity history; however, it tracked a fairly limited number of activities, and it could only be applied to the River City virtual world.

In what is arguably the most relevant project, Marty et al. created a "pedagogical dungeon" game-based learning environment, where action observation, processing, and visualization are seen as critical components, and are integrated tightly into the system [54]. In this project, observation agents are integrated into the system, providing "traces" of users' activities by logging 17 basic activities (e.g., answering a question, or chatting with another user). These traces are then processed in order to achieve an appropriate level of detail – that is, providing enough information to be pedagogically useful, without overloading the user with too much data. To achieve this, the system enables the user to define complex actions by linking basic activities using logical connectors *and*, *or* and *then*. Using these definitions, the user can encode a sequence such as "a student consults a help file after entering a room and answering a question incorrectly" as a single activity.

Once the data has been processed, it is presented to the user through a visualization engine. The researchers observed that, even with the definition of complex actions, users were faced with cognitive overload when viewing the data. This problem was not significantly improved by superficial improvements such as colour-coding the actions (e.g., chat actions shown in blue text), so the researchers added two higher-level visualizations to the system. One was an aura that surrounded each user which described their behaviour (e.g., talkative or empathetic). The other was the addition of armour to the student's avatar as he or she acquired new knowledge. In addition to these processing and visualization tools, the researchers also included "indicators", based on the recorded traces, to alert users to specific situations that might require the instructor to intervene. For example, if a student starts falling behind in a lesson, the instructor can provide additional help files. As with complex actions, these indicators can be combined to define arbitrarily complex situations. For instance, if most of the students in a class are falling behind, this situation (which could be defined as a composition of "one student is falling behind") could be addressed by significantly altering the composition of the lesson, or adding a number of supplementary exercises to help students understand the content.

This research is quite relevant to my work, as the authors tackle critical issues such as in-game action logging, parsing the resulting logs and presenting the information at an appropriate level of abstraction, and ensuring the logged and parsed actions are presented in an educationally meaningful manner. There are also aspects of game-based scenario design that are discussed further in Section 2.3.1. However, a key difference is that the action logging and parsing in the work by Marty et al. is occurring in a custom learning environment, rather than a virtual world such as Second Life or OpenWonderland. One implication of this implementation context is that the actions that are available in the custom learning environment are not necessarily representative of the actions that can be taken in any virtual world. Also, the logging tools can be integrated more closely into the design and implementation of the system than would be possible in a virtual world, which is typically developed by a third party.

> *The system provides traces of users' activities by logging 17 basic activities, which are then processed in order to achieve a pedagogically-useful level of detail.* [54]

2.4 Summary

To sum up this chapter, there has been a great deal of work in three key areas that relate to my research: educational psychology, virtual worlds-based training projects, and behaviour modelling and analysis. Taken together, these research areas offer valuable insight into some of the key challenges in my research: ensuring my framework is informed, and supported, by pedagogical research; analyzing existing projects to ensure that I am taking the lessons learned from these projects into account when designing my framework; and, finally, analyzing the students' actions from a pedagogical perspective, to determine whether the required educational objectives have been met.

Chapter 3

Avatar Behaviour Model

Platform independence is a key design principle of the MeRiTS framework. Specifically, in implementing the virtual world client, I do not want the selection of a particular virtual world to have much impact on the functionality of the other components of the framework. In principle, I want to support the implementation of the simulation environment using whichever virtual world is most appropriate for a particular context or project, without worrying about the impact on other parts of the system. To support these design principles, I have defined a model for describing the user's in-world actions, called the Avatar Behaviour Model (ABM). Using this model, the framework components have a common, implementation-independent language for referring to a user's behaviours. This ensures that, regardless of the virtual world used for implementing the simulation environment, the components have a consistent, unambiguous means of parsing the user's behaviours. Moreover, when working with a particular virtual world, one can implement a mechanism for informing the framework's components of each participant's in-world activities in terms of the behaviours defined in this model.

3.1 Background – Avatar Models

A wide variety of virtual worlds have emerged over the past decade, with target audiences ranging from socializing teenagers to businesspeople to school children. Several researchers have observed that many (if not all) of these worlds share some key characteristics. From this observation, these researchers have begun to analyze virtual worlds, in general, as a kind of interactive collaborative experience, and to create models from a variety of perspectives. These models, in turn, have helped inform the "general virtual world framework" aspect of my research. Of the modelling approaches taken, perhaps the most prevalent approach has focused on modelling the avatars used to represent users in a virtual world. As my research is based upon a generalized model of avatar actions, this approach is quite relevant, and thus I present several of these avatar-based models below.

Jarmon analyzed avatar behaviour and characteristics in a virtual world-independent manner [43]. In this paper, Jarmon took a social sciences-based approach to "the emergence of homo virtualis", and drew on research from a variety of fields to identify the key features of a person's virtual presence. These features include "3-D sensory-orthotics... robust camera controls, navigation capabilities, and the ability to create completely new virtual objects", methods of interaction (e.g., keyboard, headset, and mouse), and an "ecology of virtual actions", such as virtual touch, proximity, gaze direction, and movement. While these insights are certainly valuable, Jarmon did not specify any model or framework for viewing these capabilities as a cohesive whole, which is an important part of my work. Rather, the author used the identification of these capabilities to motivate the use of virtual worlds in education.

> *Drawing on social sciences research, Jarmon described an "ecology of virtual actions", such as virtual touch, proximity, gaze direction, and movement.* [43]

Jovanova proposed a general method of describing avatar appearance, behaviour and properties [46]. The goal of this work was to create a schema that would bridge the gap between graphics-based standards, such as Collada and MPEG-4, and agent behaviour-based standards, such as Virtual Human Markup Language, Behaviour Markup Language, and Emotion Markup Language. The proposed standard, MPEG-V (Information exchange with virtual worlds), defines a set of XML-based metadata tags, which separate an avatar's properties into four categories: appearance, animation, control, and communication. Using these tags, the idea is that an avatar can be described in an implementation-independent manner, and thus imported or exported between virtual worlds (such as Second Life and OpenSim).

Amaoka et al. developed a mathematical model to describe the personal space of agents in a virtual world [3]. This model takes into account the type of the relationship between two agents (business, friendly, or stranger), as well as other factors such as the agent's orientation (i.e., which way they are facing), their age, and gender. It also draws on psychology research, and particularly proxemics - the study of distance between people as they interact with each other. The model uses a Gaussian function, centred on the agent's face direction, and is parameterized according to the agents' age and gender. The authors included a demonstration of two agents interacting in a 2-D space, and showed how their personal spaces shifted during this interaction. The work did not extend beyond the personal space issue, and thus is not nearly as comprehensive as my research, but is related in its investigation of avatar characteristics in a virtual world-independent manner.

Verhulsdonck et al. investigated the issue of non-verbal communication in virtual worlds [82]. The authors identified a taxonomy of non-verbal communication proposed by Mehrabian, which described five types of non-verbal communication [58]: *oculesics* (eye contact and gaze), *diectics* (pointing), *gesticulation* (hand and arm gestures), *proxemics* (body distance), and *chronemics* (time between interaction). Research has shown that non-verbal communication is "an intrinsic part of the face-to-face communicative process", and helps participants reduce cognitive load by replacing words or other speech elements with gestures [82]. Verhulsdonck et al. argue that, given the importance of non-verbal communication in face-to-face communication, virtual world models should be capable of representing a range of non-verbal communication, and this should be integrated into models of virtual world communication and behaviour. Voynarovskaya provides additional evidence for the importance of non-verbal communication, albeit in a non-virtual context. In analyzing the behaviour of participants in long-term missions (e.g., space missions, polar expeditions), she identifies non-verbal communication as "useful... for monitoring the emotional states of subjects", since "humans naturally express their emotions through non-verbal behaviour" [84].

In addition to the preceding work, which identifies the behaviour capabilities and characteristics shared by all avatars, it is also important to identify the subset of this shared set

> *Given the importance of non-verbal communication, it should be integrated into models of virtual world communication and behaviour.* [82]

of behaviours that is relevant to my framework. With this goal in mind, I have identified three pedagogically-based themes that help focus my model on educationally-relevant activities. The first theme is *movement*, which is consistent with de Freitas and Neumann's identification of exploration as a crucial aspect of VW activity, and Price's linking between location and particular educational concepts [17], [68]. The second theme is *experiencing the world*, which includes both the sensing and object manipulation action categories. This theme is also influenced by de Freitas and Neumann's concept of VW exploration, as well as by Kolb's identification of experience, observation and reflection as key elements of the learning process [17], [50]. Third, the theme of *interacting socially* is supported by Girvan and Savage's use of chat logs as a means of assessing students' learning, Mehrabian's emphasis on the importance of non-verbal communication in effective social interaction, and Verhulsdonck's identification of the potential of virtual worlds to enable online non-verbal communication [26], [58], [82].

3.2 Syntax of ABM

The structure of the model, shown in Table 3.1, draws from work by Schank et al. [71] on codifying behaviours in terms of scripts and plans. Specifically, in order to define scripts and plans, Schank devised a basic grammar for describing any behaviour that a participant might take. This grammar classifies behaviours using generic definitions such as *abstract transfer*, *propel* and *consume*, and then using parameters within these generic actions to describe specific behaviours. While I use a different set of generic behaviours, my approach is similar to that taken by Schank. Augmenting Schank's basic definition of communication behaviours, I have drawn on work by Mehrabian [58] who, as mentioned in Section 3.1, identified five basic categories of non-verbal communication.

Table 3.1: BNF Grammar of ABM

1. *Behaviour* = <Movement | Sensing | Object Manipulation | Communication >

Theme: Movement

2. *Movement* = <Move | Sit | Stand >

3. *Move* = <Actor, Movement Type, Start Location, End Location >

4. **Movement Type** An element from the set of possible movement types within a particular virtual world

5. **Start Location** The actor's initial location

6. **End Location** The actor's final location

7. *Sit* = <Actor, Sit Location >

8. **Sit Location** The location where the actor sat

9. *Stand* = <Actor, Stand Location >

10. **Stand Location** The location where the actor stood

Theme: Experiencing the World

11. *Sensing* = <Actor, Modality, Source >

12. **Modality** Indicates the sense used by the actor (e.g., smell, sight, hearing)

13. **Source** The source of the stimulus sensed by the actor.

14. *Object Manipulation* = <Create | Hold | Transfer | Take | Interact >

15. *Create* = <Actor, Created Entity >

16. **Created Entity** The object created by the actor

17. *Hold* = <Actor, Held Entity >

18. **Held Entity** The object held by the actor

19. *Transfer* = <Source Actor, Recipient Actor, Transferred Entity >

20. **Source Actor** The actor, X, transferring the object

21. **Recipient Actor** The actor Y ($Y \neq X$), who received the object

22. **Transferred Entity** The object transferred from actor X to Y.

23. *Interact* = <Actor, Entity, Message, Options, Choice, Response >

24. **Entity** The object that the actor is interacting with

25. **Message** The text shown to the actor upon starting to interact with the object

26. **Options** A set of interaction choices C that the user is presented with

27. **Choice** The option $c \in C$ that the user has chosen

28. **Response** The text, action, and/or media shown after the user selects c.

Theme: Social Interaction

29. *Communication* = <Speak | Write | Gesture >

30. *Speak* = <Actor, Utterance>

31. **Utterance** The words spoken by the actor.

32. *Write* = <Actor, Message>

33. **Message** The message written by the actor.

34. *Gesture* = <Actor, Gesture Description >

35. **Gesture Description** A description of the gesture performed by the actor.

The behaviour categories (movement, sensing, communication, object manipulation) defined in the ABM are, based on a review my colleagues and I have conducted, equally applicable to any virtual world; all of the virtual worlds that we have analyzed support avatar movement, sensing of the environment, communication among avatars, and manipulation of in-world objects [11]. Furthermore, while the details of a behaviour may vary slightly from one virtual world to another, the behaviour definitions presented in this model are general enough to apply to most virtual worlds.

The ABM is, fundamentally, a user-centric model, in that each behaviour is defined from the perspective of the initiating user. Each action is characterized by a set of parameters that uniquely describe a particular instance of that behaviour. For instance, for the *move* behaviour, the combination of the actor, initial location, and destination are sufficient to uniquely describe a particular movement behaviour by a specific user. The parameters for each action are described in detail below. Note that, because of the user-centric nature of the model, the **Actor** parameter is present in every action definition, and has the same meaning throughout, and thus is omitted from the descriptions.

The movement behaviours (lines 2-10 in Table 3.1) are defined, broadly speaking, by the actor who has moved, the actor's destination location, and the way in which the actor moved to that destination. Locations, in the context of the ABM, are represented by a set of virtual world co-ordinates, x, y, z. However, these co-ordinate based locations can, through the MeRiTS framework, be linked to a richer model of scenario-based locations, as described in Sections 4.3.6 and 4.6.1. Also, the applicability of the elements in the set of movement types are dependent on the possibilities offered by a particular virtual world. In Second Life, for instance, the available movement types are walk, run, and fly; additional types (e.g., crawl or swim) would not be applicable to behaviours in that virtual world. However, the other parameters of the movement behaviour are general enough to be independent of a particular virtual world implementation. Finally, basing the movement behaviours on start and destination locations results in a certain lack of precision – that is, the actor's path from the start location to the destination is an implied linear route, rather than a continuously tracked set of co-ordinates. However, by recording an actor's movement with sufficient frequency, subsequent start and destination locations can be close enough to each other to make this lack of precision irrelevant.

The sensing behaviours (lines 11-13) are primarily defined by their **modality** parameter and a **source** parameter, which indicates the source of the stimulus sensed by the actor. The characteristics of the source are dependent on the modality of the sensing behaviour; for instance, it could indicate the direction in which the actor looked, or the sound that the actor heard. Also, the various sensing actions are recorded differently, depending on whether they are actively or passively sensed, whether they can be perceived by multiple people, and the degree to which the virtual world provides appropriate affordances for that sense. This issue is discussed in greater detail in Section 4.4.

Each of the object manipulation behaviours (lines 14-28) are defined in a similar manner. Each behaviour specifies an **entity**, which indicates the object being manipulated, and additional parameters, as needed. Specifically, the simplest behaviours – the *create* and *hold* behaviours – simply specify the relevant actor and object. The *transfer* behaviour is slightly more complex, in that it specifies both the **source actor** and **recipient actor**. The *interact* behaviour, finally, contains the parameters required to completely specify an interaction between an actor and the relevant object. That is, the entity first shows the actor a **message** and an accompanying set of **options**. The actor then makes a **choice** from among these options, and is presented with a **response** by the object.

The communication behaviours, finally, (lines 29-35) are defined in terms of the relevant actor and the message conveyed by that actor. Note that, in the case of the *gesture* behaviour, the gesture is dependent upon the communication possibilities afforded by a particular virtual world. For instance, one simple gesture is shaking one's head to indicate "no", which could be identified by a named gesture "head shake". In this case, the **description** would contain the phrase "head shake". If this gesture was not included in a virtual world by default, one might (depending on the degree to which the virtual world can be customized) be able to add it as a custom gesture.

3.3 Recording ABM-Based Behaviours

Once I have defined the set of behaviours that I am interested in through the ABM, the next step is to recognize and record instances of these behaviours in a virtual world. It should be noted that, because of the generality of the model, the behaviour definitions are (in theory) equally applicable to any virtual world, and thus these behaviours can be recognized and recorded in any virtual world. In a situation where a virtual world differed sufficiently from this model for a behaviour to be unrecognizable, then the recorder could either ignore that behaviour, or recognize a semantically equivalent behaviour and record an appropriately-translated instance of a standard ABM behaviour. It is also helpful, conceptually, to consider the model in terms of a particular virtual world. Since my implementation efforts have focused on Second Life, this virtual world is a natural choice for the following discussion.

The first step in this process is determining how each behaviour is executed in Second Life, so that it can be recognized by the recording device and relayed to the back-end in a meaningful way. Table 3.2 presents each behaviour in the model, and the corresponding means of executing that behaviour in Second Life. Note that, for the "Sense" behaviours, the means of executing the behaviour in Second Life varies according to the passivity of the behaviour. That is, for the hearing and smell behaviours, there is no direct means of executing the behaviour available in Second Life, since these are not *actions* that the user can take, but rather passive *reactions* to stimuli generated by in-world objects. The object that generated the stimulus sends a message to all users within a given radius, and those users have the stimulus conveyed to them through the recording device. For the more "active" senses (touch and taste), the results of the behaviour (touching or tasting something) are conveyed through text in a dialog box.

The next step is creating a recording device which observes each user's in-world activity, and recognizes all of the above behaviours. Each behaviour, once it is observed by the device, is converted to an appropriate set of parameters (according to the ABM definition of the behaviour), and these parameters are concatenated into a string representation of the behaviour. This string is then sent (using the native Second Life scripting command `llHttpRequest`) to a web service which is responsible for parsing the behaviour string, and then storing the behaviour for future analysis. However, in recognizing these observed behaviours, the device uses a variety of different mechanisms, depending on the characteristics of the behaviour. These differences in the recognition mechanisms are due to the details of the user interface and affordances provided by Second Life; in another virtual world, the recognition mechanisms might differ slightly from those used in Second Life. By designing the model in a virtual world-independent manner, my intent is that any virtual

Table 3.2: Execution of Modelled Behaviours in Second Life

Action	**Implementation in Second Life**
Move	Avatar movement, using arrow keys
Sense (sight)	Shift gaze using mouse, camera controls (zoom, rotate camera angle)
Sense (hearing)	An object which plays a sound clip for the user
Sense (smell)	An object which describes a smell to the user
Sense (taste)	An object which describes a taste to the user
Sense (touch)	An object which describes a texture to the user
Create	Create an item from one's inventory
Hold	Hold an item that one owns
Transfer	Give an item to another user
Interact	An object which uses dialog boxes to provide interactive content
Speak	Text and voice chat, instant messaging
Write	Scripted objects that display text messages on an object's surface
Non-Verbal	Built-in and custom animated gestures

world should have *some* mechanism for recognizing each behaviour, and thus one should be able to implement the recording device in any virtual world.

Behaviours are, in general, recorded by observing each behaviour as it is performed by the avatar wearing the recording tool. There are, however, slightly different strategies for handling *single-occurrence* behaviours, *movement* behaviours, and *sensing* behaviours. Single-occurrence behaviours (e.g., sending a text chat message or picking up an object) are relayed to the web service immediately (again, using `llHttpRequest`) upon detection. Movement behaviours, which can occur many times per second and may not be unique, are stored locally within the behaviour detection device for a brief period. These behaviours are parsed to remove duplicates, and lists of parsed behaviours are sent periodically to the web service, which is able to extract individual behaviour instances from this list, and store each instance appropriately.

The recording of sensing behaviours is dependent on the affordances provided by the virtual world. The *looking* behaviour recorded by directly tracking the avatar and recognizing when the gaze direction (retrieved via the SL scripting commands `llGetCameraRot` and `llGetCameraPos`) has changed. The *hearing* behaviour, on the other hand, is somewhat more complex, since the affordance provided by Second Life – playing a sound clip, using the SL scripting command `llPlaySound` – cannot be directly observed by the recording device. Rather, the object that caused the sound sends a notification message, indicating that a sound has been produced, to the avatar that initiated the behaviour. Each avatar's recording device, meanwhile, listens for these messages, and processes them appropriately. Note that some behaviours, such as hearing and smell, could also be considered from a "passive" point of view – that is, a user might *hear* a sound without intentionally *listening* to the object that generated the sound.

The remaining sensory behaviours – smell, taste, and touch – are not well-supported by the affordances provided by Second Life. In this case, they are simply described to the user via content shown in dialog boxes and, with regards to the recording system, considered as a special case of object manipulation, described below. The recording of sensory behaviours is summarized in Table 3.3, below.

Object-manipulation behaviours, such as object creation, transfer and interaction, are

Table 3.3: Recording sensing behaviours in Second Life

Sense	**Passivity**	**Multiple people?**	**Means of recognition**
Sight	Active	Single	Tracking user's gaze direction
Hearing	Passive	Multiple	Notification message received from object that played sound clip
Smell	Active	Single	Notification message received from object that showed user a description of smell
Taste	Active	Single	Notification message received from object that showed user a description of taste
Touch	Active	Single	Notification message received from object that showed user a description of texture

recorded by parsing notification messages sent by the object to the recording device, in parallel with the messages that the object sends to the user that is manipulating it. Specifically, when a user interacts with an object, that object sends a notification message to the recording device (in addition to an interaction message that is sent to the user). The device then converts this message into an ABM-based string representation of the interaction, and sends the resulting string along the web service.

3.4 Summary

In this chapter, I have presented the Avatar Behaviour Model, which defines the behaviours that an avatar can exhibit in a virtual world. I have also provided motivation for using this model as a basis for recognizing and recording behaviours in a virtual world. Finally, I have discussed such a device, implemented in Second Life, that recognizes and records these behaviours.

Chapter 4

MeRiTS

Motivated by the challenges described in the introduction (Chapter 1), and informed by the research described in the related work (Chapter (2), I have defined four core requirements for my platform:

1. The platform must support instructors in creating realistic, educationally effective scenarios in a usable manner.

2. The platform must enable participants (e.g., students) to experience these scenarios in a realistic, engaging way, and to provide a meaningful opportunity for learning.

3. The platform must record each participant's actions while going through the scenario.

4. The platform must enable instructors to analyze the resulting scenario trace, so that the instructors can evaluate the students' performance, and determine whether educational goals have been met.

To meet these requirements, I have created a framework for virtual world-based training called MeRiTS (Mixed Reality Training System). MeRiTS enables instructors to create and deploy virtual world-based training scenarios, and to analyze students' behaviours in those scenarios, using a combination of a virtual world client, a set of scenario definitions, and a scenario execution engine. This record, in addition to supporting instructors' analysis of student behaviour, could also be used to support students' reflection on their own behaviour. I have designed MeRiTS as a general-purpose platform, upon which a wide variety of scenarios may be constructed. Supporting this design is a model of participant behaviours (described in Chapter 3), which provides logical consistency to the framework as a whole, while maintaining platform independence. In this chapter, I describe the architecture of the MeRiTS framework – that is, the components that make up the framework, and their relationship to one another. I also present several key implementation challenges and corresponding design decisions that were a part of creating the virtual world client.

4.1 Training Scenarios

The architecture of MeRiTS is summarized at a high level in the following paragraphs, and described in detail in the corresponding sections. Each scenario is defined and enacted by the scenario definition and execution components of the framework, respectively. Through these components, an instructor is able to create a scenario that comprehensively describes all relevant actions and allows the participant, in a non-restrictive way, to pursue a range of actions at any point in the scenario. In describing these scenarios, and the MeRiTS framework as a whole, I will use the following terms:

Scenario A training exercise, encapsulating interactive objects, associated rules and processes, and one or more participating students.

Participant A participant in the scenario. Note that this participant may be a student, an instructor, or a facilitator.

Instructor An educator with some involvement in the scenario. This person may help develop the scenario, facilitate its enactment, or evaluate the results.

Object An interactive entity within the scenario, which has some representation (e.g., a 3D model) in the virtual world.

Action An interaction with an object that affects the state of the scenario.

Workflow A set of procedures that determine how an object will respond to each action.

Constraint A limitation on a particular action initiated by a participant.

Behaviour Represents an instance of a generic behaviour, as defined by the ABM.

Scenario Trace A record of the sequence of behaviours exhibited by each participant while enacting the scenario.

There is a certain amount of unavoidable overlap between some of these terms. Specifically, a *participant* is the term that is used for *anyone* who is present in the scenario, whether they are a student, facilitator, or an instructor. I also refer specifically to an *instructor* as someone who is involved in designing, facilitating, or evaluating the results of a scenario. An instructor may also be a participant in a scenario, if their presence is required for the scenario to proceed successfully – for instance, if the instructor needs to take on a role for the scenario to proceed smoothly.

Similarly, the term *behaviour* covers all of the behaviours defined by the ABM (which is presented in Chapter 3), including interactions with objects. The term *action*, meanwhile, refers specifically to interactions that have some impact on the scenario. Therefore, every action is also a behaviour, but not every behaviour is an action. For the purposes of describing the MeRiTS architecture, it is helpful to be able to refer to these scenario-influencing behaviours using a distinct term.

The range of scenarios that can be created is quite broad, and may be considered according to the following criteria:

Number of Participants The framework can support single-participant scenarios or multiple-participant, collaborative scenarios.

Communication Depending on the participation characteristics, communication may be non-existant (in the case of a single-participant scenario), asynchronous (where participants leave messages for each other), or synchronous, where participants communicate with each other while they participate in the scenario. In the asynchronous communication situation, messages may be created, sent and received via special-purpose objects, which would in turn store and retrieve content from an information storage component of the MeRiTS framework.

Process The framework can support scenarios with varying levels of structure. The degree to which a scenario is structured may be seen as a continuum, with exploratory scenarios at one end, and process-oriented scenarios at the other. In an exploratory scenario, there are no processes or restrictions imposed on the participant. In a process-oriented scenario, on the other hand, the participant must follow a rigidly-defined process. One can also imagine a range of "loosely-organized" scenarios falling between these extremes. Indeed, within a single scenario, there may be elements that can be categorized as either exploratory or process-oriented. For example, within the EMT hand-off scenario (described in Section 5.1), the treatment of the victim at the roadside is exploratory in nature. There are a wide range of treatment options available, and the participant is free to choose or dismiss any of the options, and to pursue the chosen options in any order. However, the transfer of the victim from the roadside to the hospital follows a rigidly-defined sequence, and is thus decidedly process-oriented.

Objects Objects may be implemented in a variety of ways, which are described below:

- inactive: An inactive object (or prop) establishes a sense of realism, but does not possess any behaviour.
- isolated: An isolated object possesses behaviour which does not relate to the scenario process.
- integrated: An integrated object possesses behaviour which is connected to the scenario process, and thus enables the participant to fully experience the scenario.

Roles A participant may take on a particular role within the scenario, which carries with it certain responsibilities. For example, in the EMT hand-off scenario, a nurse has a very different set of responsibilities from an EMT. Within the EMT role, one can consider a *lead* role as having a different set of responsibilities from a *supporting* role. These roles are integrated into the MeRiTS system in the form of role-based rules that can be enforced, and they may also be important in the evaluation of each participant's actions by the instructor.

The architecture and implementation of the MeRiTS system is described in detail in the following sections. For each component, the functional architecture is presented, describing the means by which a scenario is defined, the process of running a scenario, and the capabilities provided for analyzing participants' in-world actions.

4.2 Architecture

The component architecture of the MeRiTS platform is shown, at a very high level, in Figure 4.1. In this diagram, arrows between components indicate information passed from one component to another.

The *scenario definition* component is shown at the top of the diagram. Each scenario is defined through a set of workflows, augmented by constraints and timed events. Each workflow specifies the behaviour of one of the interactive objects in the scenario – that is, it defines the way in which the object responds to each available action, and the ways in which that action impacts the scenario as a whole. Complementing the workflows are

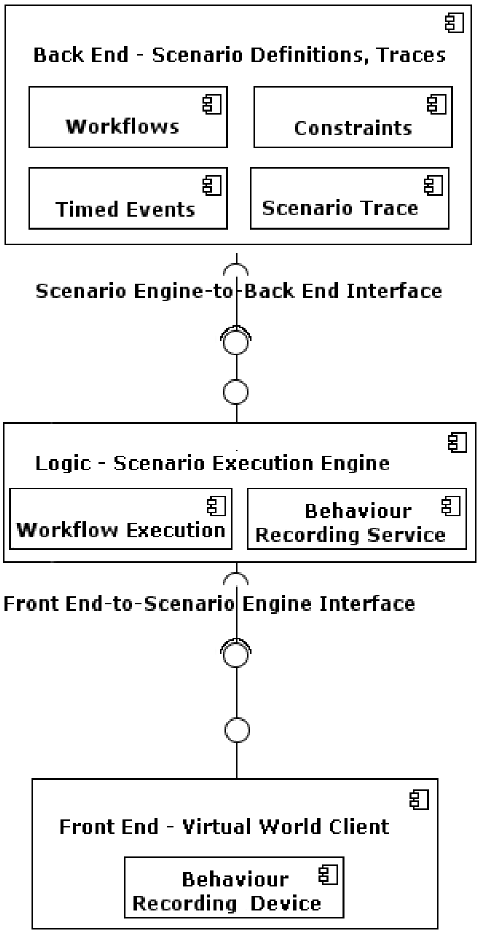

Back End - Scenario Definitions, Traces
Workflows
Constraints
Timed Events
Scenario Trace
Scenario Engine-to-Back End Interface
Logic - Scenario Execution Engine
Workflow Execution
Behaviour Recording Service
Front End-to-Scenario Engine Interface
Front End - Virtual World Client
Behaviour Recording Device

Figure 4.1: MeRiTS Architecture - Component Diagram

constraints that define the feasibility and advisibility of the actions defined by the workflows. Taken together, the workflows and constraints provide an action/response-oriented definition of the scenario, which is used to guide the participants through the scenario experience. Finally, the scenario definition may also include timed events, which add an element of non-participant-initiated interaction to the scenario.

Each participant interacts with the scenario through a *virtual world client* component, shown at the bottom of the diagram. Through this component, each participant experiences the scenario in an immersive environment. During the enactment of the scenario, the participant is able to explore the scenario environment, communicate with other participants, and interact with objects. As with the scenario definition component, the virtual world client provides a highly flexible environment which the participant is free to explore as he or she wishes. This component is described in detail in Section 4.4.

As the participant moves, communicates, and interacts with the virtual world, a *behaviour-recording component* observes these activities and records them in an online repository. Once they are recorded, they can be subsequently processed and analyzed by the instructor, using a comprehensive set of behaviour analysis tools. The behaviour recording component is described in Section 3.3, and the analysis tools are covered in Section 4.6.

The components described above are connected through the *scenario-execution engine*, shown in the middle of the diagram. This component connects the virtual world client with the scenario definition, and thus enables the client to respond appropriately to participants' actions. Specifically, the component includes a workflow execution sub-component, which executes the workflow corresponding to a given object. The result of executing the workflow – which is determined by the chosen action and the overall state of the scenario – causes the relevant object, and the scenario as a whole, to respond accordingly. The component also includes functionality for recording each participant's behaviours, thus creating a scenario trace which can be subsequently analyzed by the instructor. This component is described in Section 4.5.

Finally, underlying the core components of the MeRiTS system, I have defined a model for describing avatar actions in a virtual world. This model is described in detail in Chapter 3.

In presenting the various components of the architecture, I will be making use of a running example, to help explain the components and their relationship to each other. The example is a simple patient rescue scenario, where the participant (or participants) needs to assess and treat a person who has collapsed, facedown, on a city street. In this scenario, the participant needs to do a quick patient assessment, perform CPR, move the patient to a stretcher, and push the stretcher to an ambulance for transport and further treatmnet. The objects, actions, and rules of the scenario are explained as the relevant components of the framework are presented.

4.3 Scenario Definition

The scenarios, which are at the heart of the MeRiTS framework, are defined as instances of the scenario definition model shown in Figure 4.2. In this diagram, and the other model diagrams in this section, the arrows betweeen elements of the model represent relationships between those elements, with indications of cardinality included at the endpoints of the arrows where necessary. For instance, the connection between the *scenario* and *workflow* entities may be read as "a *scenario* is enacted according to one or more *workflows*". Each component is described briefly in the following paragraphs, and defined in detail in the following sections.

Two of the most fundamental parts of the model are the *variables* and *objects*, which maintain the scenario's state and represent the scenario's interactive entities, respectively.

The behaviour of each interactive object is defined through a *workflow* (which is executed by the workflow-execution component). Augmenting these workflows are a set of *constraints* which define the conditions under which the participants can or should use the interactive objects. The instructor may also define *timed events* for a scenario, which cause the scenario to generate actions when a certain amount of time has elapsed in the enactment of a scenario. Finally, the scenario definition may include a set of *locations*, which associate virtual world co-ordinates with meaningful places or landmarks in the scenario.

The relationships between the workflows, constraints, and timed events are maintained by the aforementioned variables and objects. As a whole, these components enable the instructor to define scenarios that are reliable, realistic, and dynamic. That is, the scenarios are reliable in that the full set of actions and scenario responses are specified through the workflows and constraints, and therefore the results of any participant action may be appropriately defined. The framework's object-behaviour specification capabilities enable the instructor to create objects that behave in an appropriate, realistic manner. Finally, the incorporation of both participant-driven and timer-triggered events enable the instructor to create scenarios that provide participants with an engaging, evolving experience.

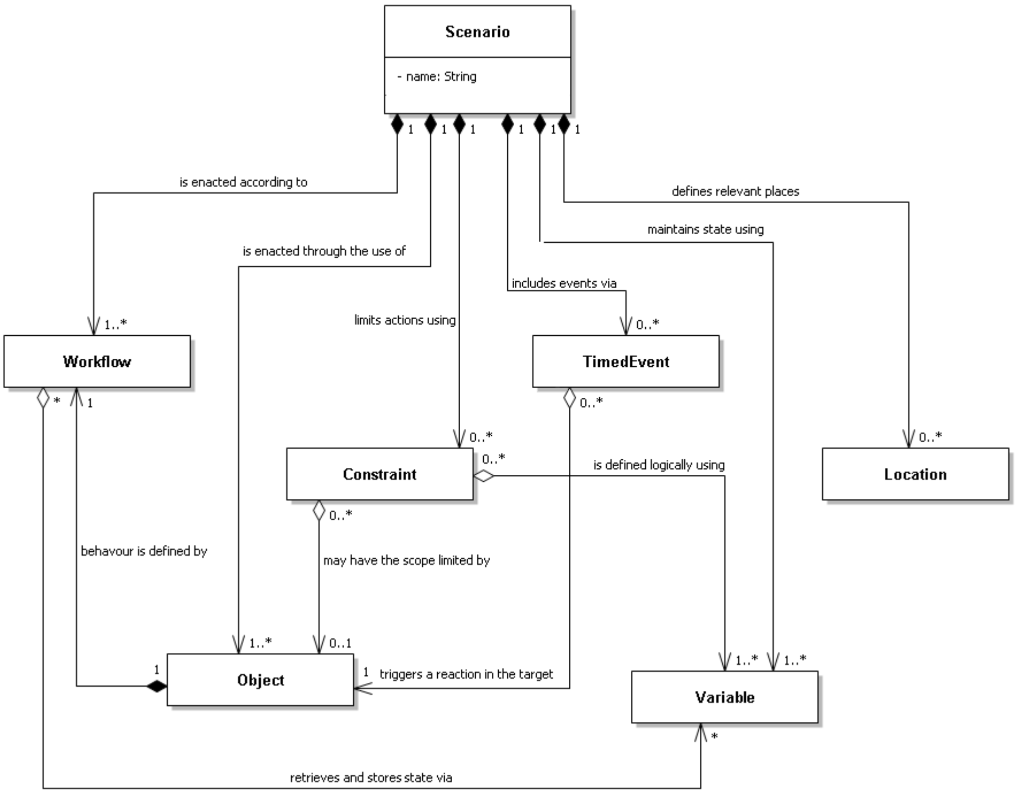


Figure 4.2: MeRiTS Architecture – Class Diagram

In the following sections, I describe each of the elements that make up the scenario definition, and finally explore some of the tools that are used to create these definitions.

4.3.1 Objects

The objects within a scenario, each of which is represented in-world by a 3D model of the entity, embody the ways in which a participant can interact with the scenario. The definition of each object includes a set of *actions*, each of which has a *type* and *parameter*, to facilitate conceptual organization. Furthermore, the objects act as a point of connection for the scenario's workflows, constraints, and timed events. Each object, finally, has a set of *co-ordinates*, which represent its location in the virtual world. These co-ordinates may, in turn, help specify object-based locations, which are described in further detail in Section 4.3.6. These definitions and relationships are shown in Figure 4.3.

The *parentName* field is required for *complex objects* that have their actions distributed among a set of hierarchically-organized *child objects*. In this case, the actions for the object would defined for the complex object (and would be implemented in the corresponding workflow for the complex object), but they would be available to the user by interacting with the appropriate child objects.

In the running "patient rescue" example, I define two objects for the scenario: the *pa-*

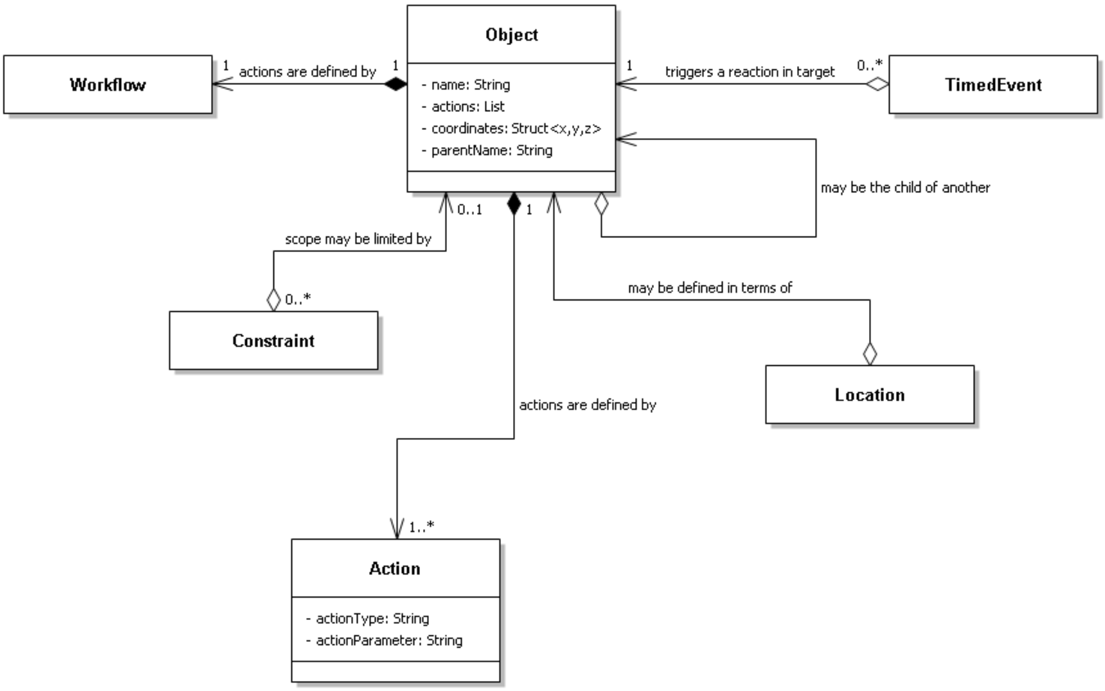


Figure 4.3: Object Definition - Class Diagram

tient and a nearby *stretcher*. The patient is defined as a complex object, with head, neck, and arm child objects as shown in Figure 4.4.

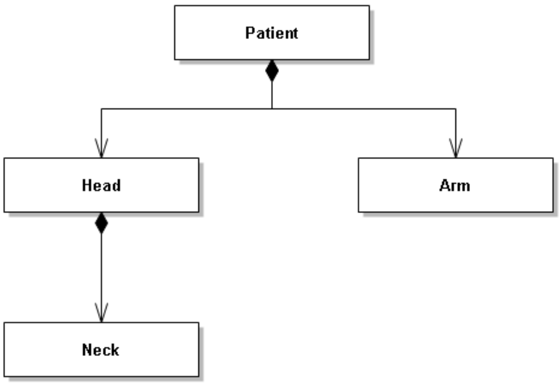


Figure 4.4: Example Patient Composition

The participant may interact with the patient via the actions shown in Table 4.1.

The stretcher, meanwhile, is a simple object, with no child objects. The participant may interact with the stretcher via the actions shown in Table 4.2.

Table 4.1: Patient actions

Action	Type	Param	Available Via	Description
Roll over	Move	Roll	Patient	Roll the patient over from his back to his front (or vice versa)
Check pulse	Assess	Pulse	Neck, Arm	Check the patient's pulse, either in his wrist or his jugular vein
Perform CPR	Treat	CPR	Head	Perform CPR on the patient, as a basic form of emergency treatment
Move to stretcher	Move	Stretcher	Patient	Move the patient on to a nearby stretcher
Drop pulse	Change vitals	Drop pulse	Patient	Reduce the patient's pulse by 10 BPM

Table 4.2: Stretcher actions

Action	Type	Param	Description
Push to patient	Move	Patient	Push the stretcher to the patient
Push to ambulance	Move	Ambulance	Push the stretcher to the ambulance

4.3.2 Variables

The variables within a scenario are responsible for maintaining the overall state of the scenario. That is, each variable maintains a particular piece of this state (e.g., the patient's pulse), and the set of variables as a whole maintains the state of the entire scenario at any point in time. The variables, therefore, form a crucial part of the definition and execution of a scenario, especially considering that the definition of the scenario is distributed across multiple workflows, and may include timed events and constraints, as well. Each variable is defined by a *name*, which is unique for the scenario to which the variable belongs, a *type* drawn from a set of basic variable types – integer, string and boolean – and a default value. Each variable also stores its current *value*, which represents one part of the current state of the scenario. As with the object entities, variables act as a means of connecting the scenario's workflows and constraints. Note that since a given variable *might*, but does not *have to*, correspond to a particular object, the variables exist independently of the objects contained within a scenario. The variable definition is shown in Figure 4.5.

In my running example, I define variables to store the patient's pulse, the current position and orientation of the patient, and the location of the stretcher. These variables are shown in Table 4.3.

4.3.3 Workflows

Executable workflows are used to define the behaviour of each object (including complex entities such as simulated patients and other non-player characters) in a scenario.

The workflows are defined according to the workflow-definition model shown in Figure

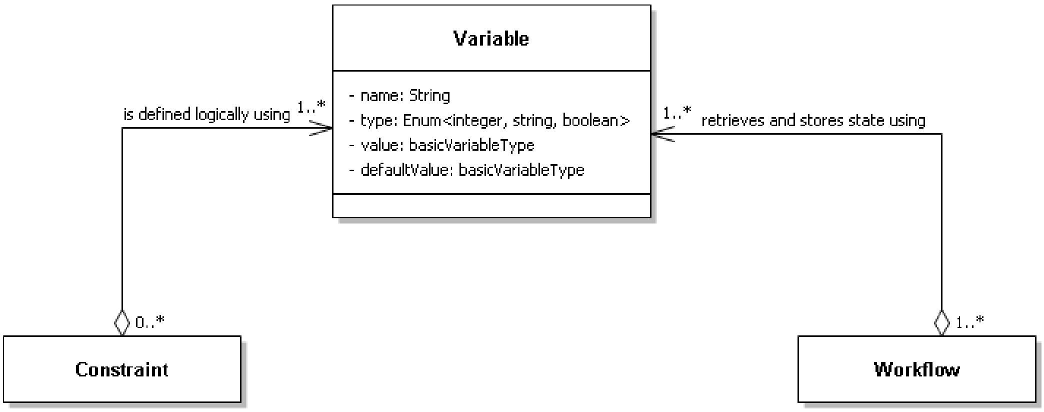

Variable
- name: String
- type: Enum<integer, string, boolean>
- value: basicVariableType
- defaultValue: basicVariableType
is defined logically using
1..*
0..*
Constraint
1..*
retrieves and stores state using
1..*
Workflow

Figure 4.5: Variable Definition - Class Diagram

Table 4.3: Patient Scenario Variables

Name	**Type**	**Default Value**	**Description**
Orientation	string	faceDown	Whether the patient is lying face up or face down
Position	string	onFloor	Where the patient is currently located
Pulse	integer	50	The patient's current pulse
Stretcher Location	string	ambulance	The current location of the stretcher

4.6. This model encompasses all of the workflow elements that are required to express the behaviour of objects within a scenario. Through the workflow model, the behaviour of any workflow may be specified using a collection of components (each of which is either an *assignment*, *sequence*, or *conditional* element).

More specifically, a workflow is defined for a given *object*, and its state is defined by a set of *variables*. The execution of the workflow begins with an *initialize* element, which sets the workflow variable values according to the values passed in through the *input variables*. The workflow is terminated by a *finish* element, which returns the results of the workflow – both the message and/or actions that are to be conveyed to the participant and any updated variable values – via the *output variables*. The steps in between are made up of a collection of *component* elements, which are separated into the following types:

- **Assignment**: assigns new values to one or more *variables*
- **Sequence**: defines a sequential ordering for a given set of *components*
- **Conditional**: defines a set of if/else conditions, and associates a *component* with each condition. Note that each condition, in turn, tests one or more *variables* to see if the associated *component* will be executed.

Note that both the sequence and condition elements are based on the composite pattern. That is, they are both defined as types of components, and also both *contain* components. Therefore, one can create arbitrarily complex control flow structures using various nested combinations of sequence and condition elements. Note that each of these control flow elements, in order to have some impact on the state of the workflow, will ultimately contain

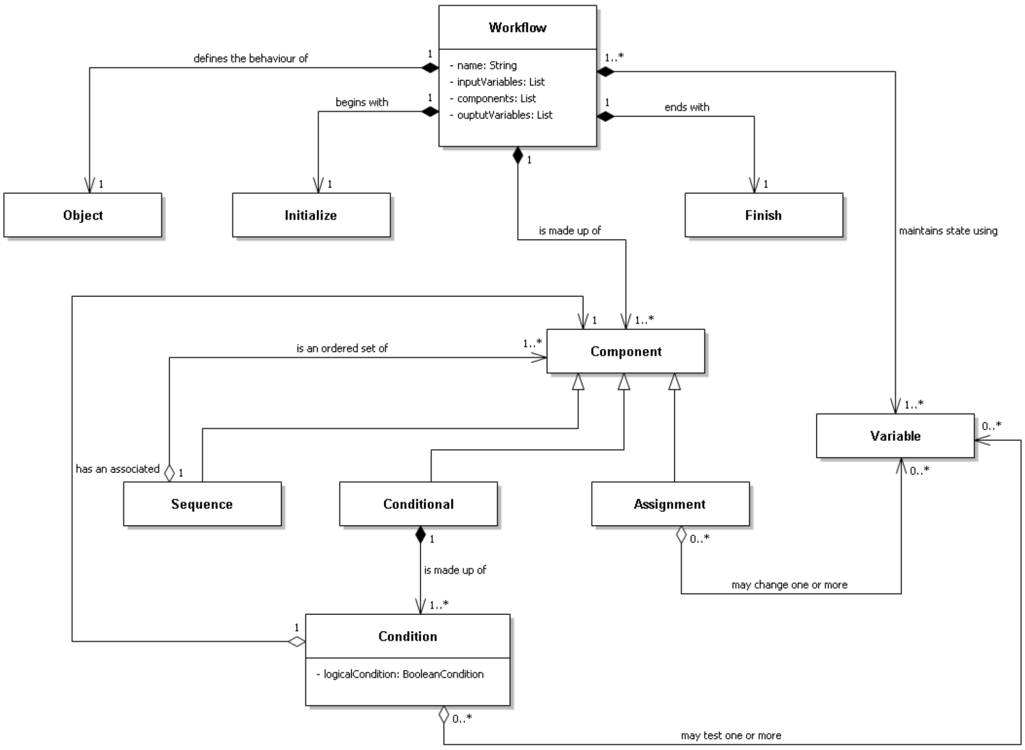


Figure 4.6: Workflow Definition - Class Diagram

one or more assignment statements. This composite-based approach to the definition of the *component* element means that the *initialize* and *finish* elements cannot be included as subtypes. That is, since these elements can only be included at specific places in the workflow, they cannot be defined as a subtype of an element that can be included anywhere in the workflow.

For the purposes of actually defining the way in which an object responds to a given set of actions, each workflow will have the structure shown in Figure 4.7.

In this structure, there is a *conditional* element corresponding to each of the *actions* that have been defined for an object. Within each *conditional* element, there is a set of elements that define the object's response for that action. These elements will likely include a *sequence*, to organize the elements that are responsible for responding to a particular action, *assignment* elements, to modify variable values and return messages to the client, and possibly additional *conditional*, to perform further decision-making logic related to the given action.

Returning once again to my running example, I have created a workflow for the Patient object, shown in Figure 4.8. Note that grey boxes indicate conditional components, while white boxes indicate assignment components. For example, if the participant chooses to roll the patient over, this action is first processed in terms of its place within the "move" category (which corresponds to the first conditional component). Within this component, the action could be either to roll the patient over (left branch), or to move the patient to the stretcher (right branch). Since the participant has chosen to roll the patient over, the left branch is followed, and the patient's orientation is then checked to see if he is currently face-up or

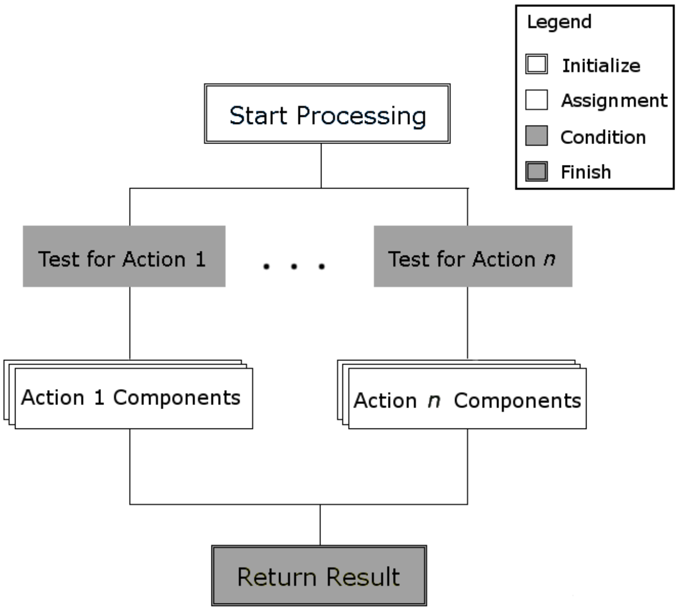


Figure 4.7: Structure for Object Workflows

face-down. The orientation is then changed appropriately, and this result is returned to the service that requested the execution of the workflow.

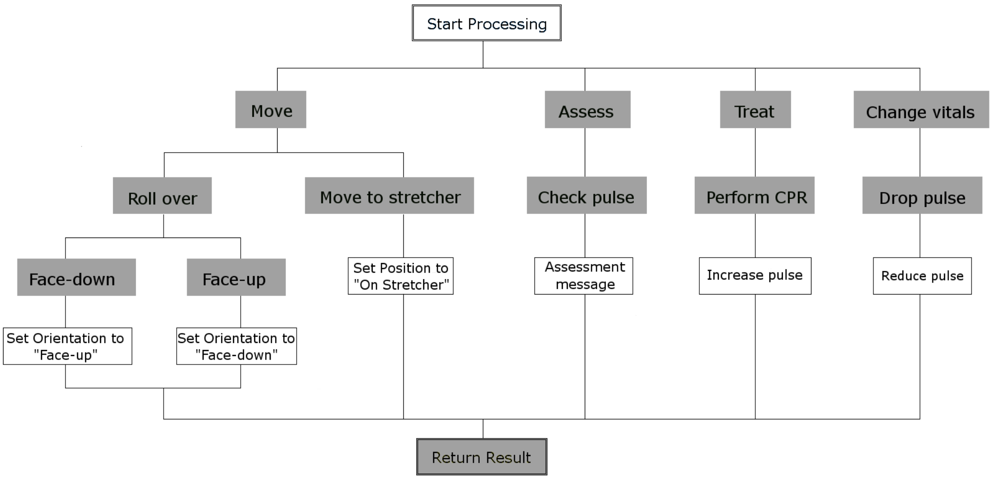


Figure 4.8: Patient Workflow

I have also created a workflow for the stretcher which, given the limited set of actions for the object, is much simpler than the patient workflow. It is shown in Figure 4.9.

For more on the process of defining a workflow, see Section 4.3.7.

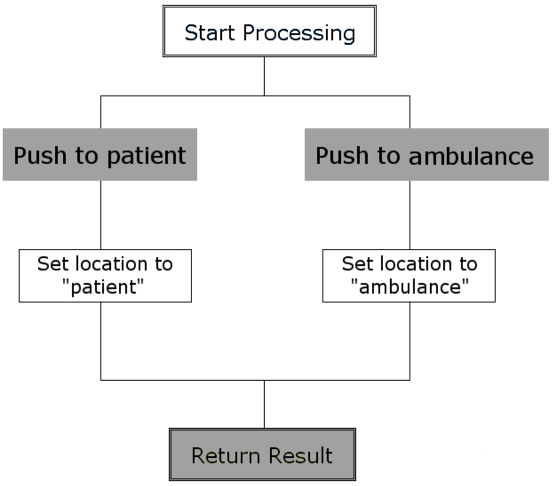


Figure 4.9: Stretcher Workflow

4.3.4 Constraints

In addition to the workflow definition, MeRiTS also supports the definiton of *constraints*, which constrain the behaviour of interactive objects in the scenario. While the functionality provided by constraints can also be implemented using workflow-based conditional statements, there are two crucial advantages to using constraints. First, from a conceptual point of view, it allows the error-handling logic to be separated from the "normal action processing" logic. Second, from a design standpoint, the primary advantage of this approach is that constraints can be applied across multiple objects, whereas workflow-based conditions are limited to the object with which the the workflow is associated. This limitation of workflow-based conditions means that a constraint that needs to be applied across multiple objects must be duplicated within each associated workflow. The constraints are defined using the model shown in Figure 4.10.

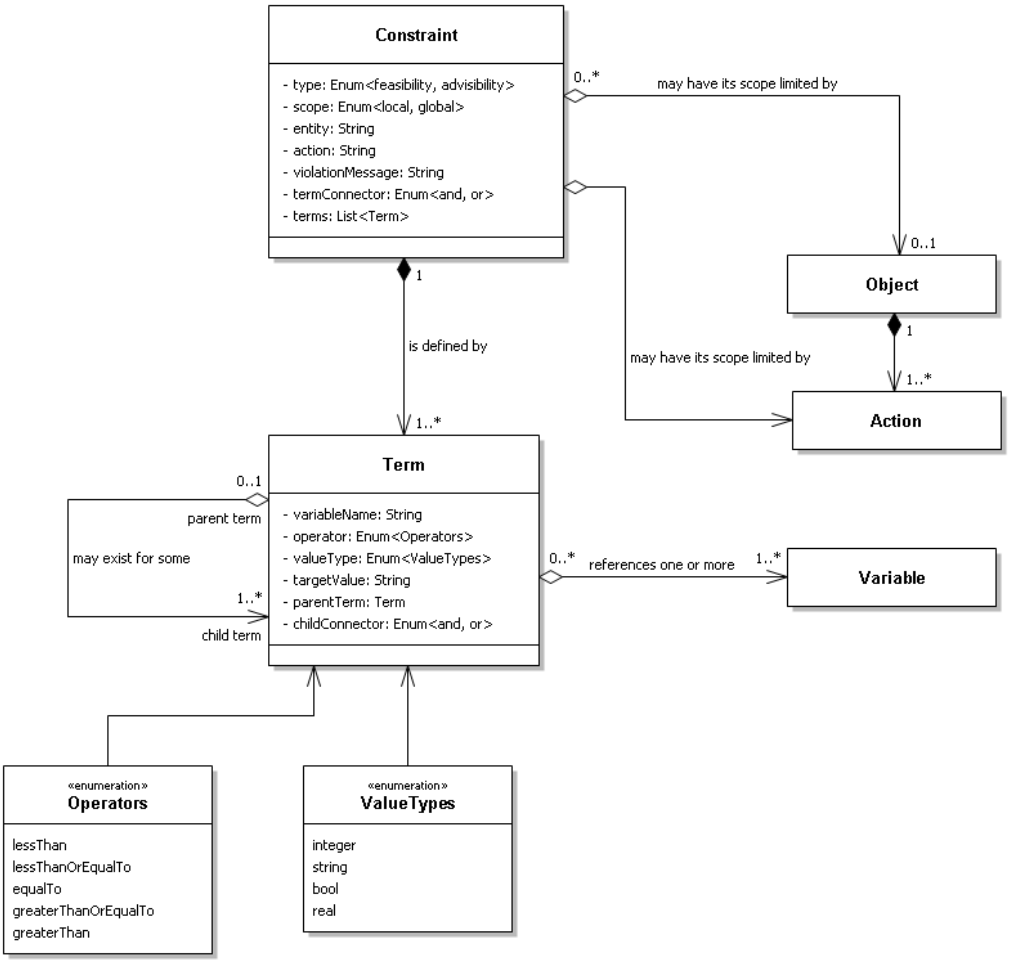


Figure 4.10: Constraint Definition - Class Diagram

The constraints are divided into two types: *feasibility* and *advisibility* constraints. Feasibility constraints describe conditions that, if violated, result in infeasible or impossible actions, for instance, moving a patient onto a stretcher when the stretcher has not yet been retrieved, or performing this action more than once. Advisibility constraints, meanwhile, are appropriate for conditions that *may* be violated by a participant, but it is understood, in the context of the particular scenario, that this is not a good idea. For instance, in a healthcare context, a patient *may* be moved if his vital signs are poor, but this is inadvisible in terms of the continued stability and health of the patient.

Constraints may also be defined at either a *local* (that is, object-specific) or *global* scope. Constraints defined at a local scope are applied only when a participant interacts with a particular object. For instance, the example previously given of moving a patient onto a stretcher could be defined as a local constraint, applicable only to the patient object. Meanwhile, the patient health constraint could be a global constraint, applicable to any action taken by the participant.

The constraint *entity* and *action* are used for defining local constraints, and specify the object and action to which the constraint applies. Using these parameters, a constraint may be associated with a specific *object*, and limit the conditions under which the participant can take certain actions using that object. The *violation message* stores the violation message, which is returned to the participant who initiated the action that caused the violation. If the constraint that has been violated is a *advisibility* constraint, then the violation message is appended to whatever message is generated by the execution of the relevant action. If, on the other hand, the constraint is a *feasibility* constraint, then the violation message is all that is returned to the participant. The *term connector*, finally, determines how the constraint's terms will be evaluated, as a composite logical expression. If the connector is *or*, then the constraint will be met if *any* of its term are true. If the value is *and*, then the constraint will only be met when *all* of its terms are true.

Each of the terms of a constraint define a condition on a given variable. The *variable name*, *operator*, *value type*, and *target value* are used to define the logical condition against which the current variable value will be tested, to see if the constraint is met. For instance, a term might have the following parameters: $<$*variable* = Pulse, *operator* = greater than, *value type* = integer, *target value* = 50 $>$. These parameters correspond to the logical condition "Pulse>50", and as long as this condition is satisfied, then the constraint as a whole is considered satisfied.

Finally, the *parent term* and *child connector* fields are used to define composite terms, by creating a hierarchical relationship among the terms of a constraint. This hierarchical relationship is required for constraints that use a blend of *and* and *or* comparisons in their logical expressions. The composite terms, which can be nested arbitrarily deeply, may then be organized using the *parent term* and linked using the *child connector*. The constraint term class uses the composite design pattern, in that both simple and composite terms are associated with constraints in the same way, and the logical statements they represent is parsed using a consistent approach.

From an implementation perspective, the constraints are stored as part of the MeRiTS database, and integrated into the process of executing each action. Specifically, each constraint is stored as a one-to-many relationship between a parent *constraint* entity, and one or more child *constraint term* entities, as defined in the constraint definition model. When a participant executes an action, all relevant constraints (that is, all global constraints, as well as all constraints that are defined for the action that was taken) are retrieved from the database, and each constraint is checked to see if the logical statement created by its terms are satisfied. If not, either the action is cancelled and the violation message is returned immediately (for a feasibility constraint), or the violation message is appended to the result of the action (for an advisibility constraint).

Within the context of the running example, I have defined the following constraints:

Table 4.4: Patient Constraints

Name	**Type**	**Scope**		**Description**
		Object	**Action**	
Can check pulse	feasibility	Patient	Check pulse	$Orientation =$ face-up
Can perform CPR	feasibility	Patient	Perform CPR	$Orientation =$ face-up
Can push to patient	feasibility	Stretcher	Push to patient	$Location \neq$ patient
Can push to ambulance	feasibility	Stretcher	Push to ambulance	$Location \neq$ ambulance
Should push	advisibility	Stretcher	All	$Role \neq$ bystander
Should move patient	advisibility	Global		$Pulse$ <100 and $Pulse$ >60

Taken together, the workflows and constraints offer the participant non-restrictive guidance as they go through the scenario. That is, the participant receives feedback but, as long as he or she does not violate feasibility constraints, then the participant is not restricted in terms of the actions (either correct or incorrect) that may be executed.

4.3.5 Timed Events

An additional component of the scenario definition is the implementation of timed events. These events, in contrast to the action-response model used by the interactive objects and associated workflows, are intiated when a specified amount of time has elapsed in the scenario. A timed event is defined according to the model shown in Figure 4.11. Events are typically defined when the scenario is developed, although they can also be modified subsequently, if necessary (e.g., by an instructor facilitating the scenario). However, because of the way that timed events are executed by the virtual world client, one cannot change these events while a scenario is in progress. Any such changes to the timed events will be reflected in the scenario the next time the scenario is enacted.

The *first occurrence time* determines the time that the event timed first occurs, in seconds elapsed since the start of the scenario. The *frequency* determines whether the event is a single or repeatedly-occurring event, and the *time interval* determines the frequency of a repeated occurrence event. The *target* and *action* identify an object – and an action for that object –that is triggered when the timed event occurs. The *probability*, finally, is an optional parameter that may be used to assign a probability to the timed event occurring. If this parameter p is defined, then the event will only happen with probability p.

The timed event execution mechanism is implemented, primarily through the virtual world client, using the following process.

1. When the scenario is restarted, the MeRiTS web service sends a list of the timed events for the given scenario (let us assume there are n such events) to the virtual world client.

2. This list of events is processed by a timer component in the virtual world, and each occurrence time ($t_1 \cdots t_n$) is stored by the timer.

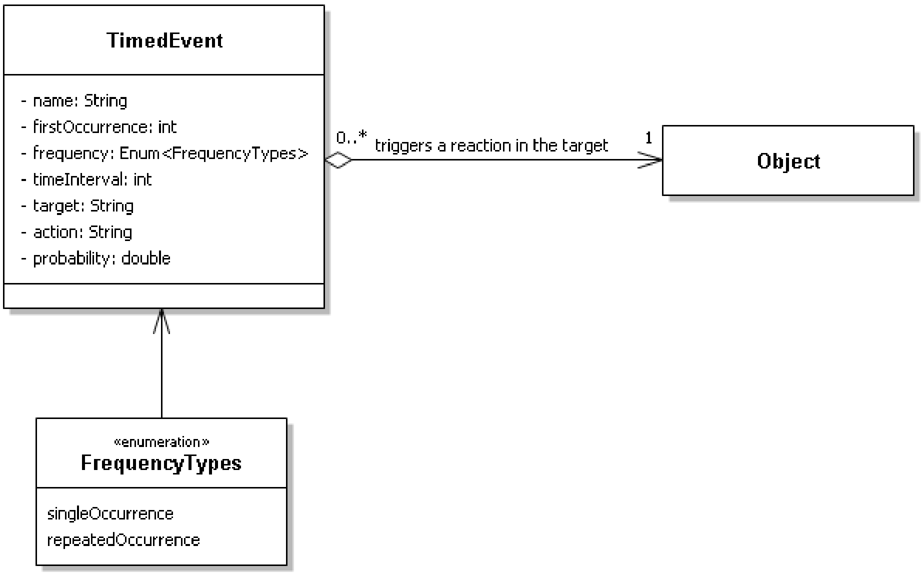


Figure 4.11: Timed Event - Class Diagram

3. The timer then iterates once per second and increments the elapsed scenario time t.
4. When the elapsed scenario time t is equal to some stored occurrence time $t_i (1 \leq i \leq n)$, it sends the associated *action* message to the appropriate *target* object (optionally with probability p). If the event is a repeated occurrence event (with time interval int_i), then the timer creates a new occurrence event at time $t_i + int_i$.

The triggered action may, in turn, generate a call to the target object's workflow. Through this mechanism, the scenario can include timer-initiated workflow events with minimal interaction between the virtual world client and the workflow engine.

In my running example, I have defined an event to represent the patient having a heart attack two minutes into the scenario. This event would be defined as shown in Table 4.5. The *dropPulse* action identified in this event is defined in the last row of Table 4.1.

Table 4.5: Patient Timed Event

Parameter	Value
Name	Heart attack
First Occurrence	120 seconds
Frequency	Single occurrence
Time Interval	N/A
Target	Patient
Action	changeVitals, dropPulse
Probability	N/A

4.3.6 Locations

In order to facilitate the parsing and analysis of participant actions within a scenario, one can define locations of interest for each scenario. These locations may be defined in three ways: using fixed co-ordinates, in terms of a target object, or in terms of a target object under certain conditions. This location definition model is shown in Table 4.12.

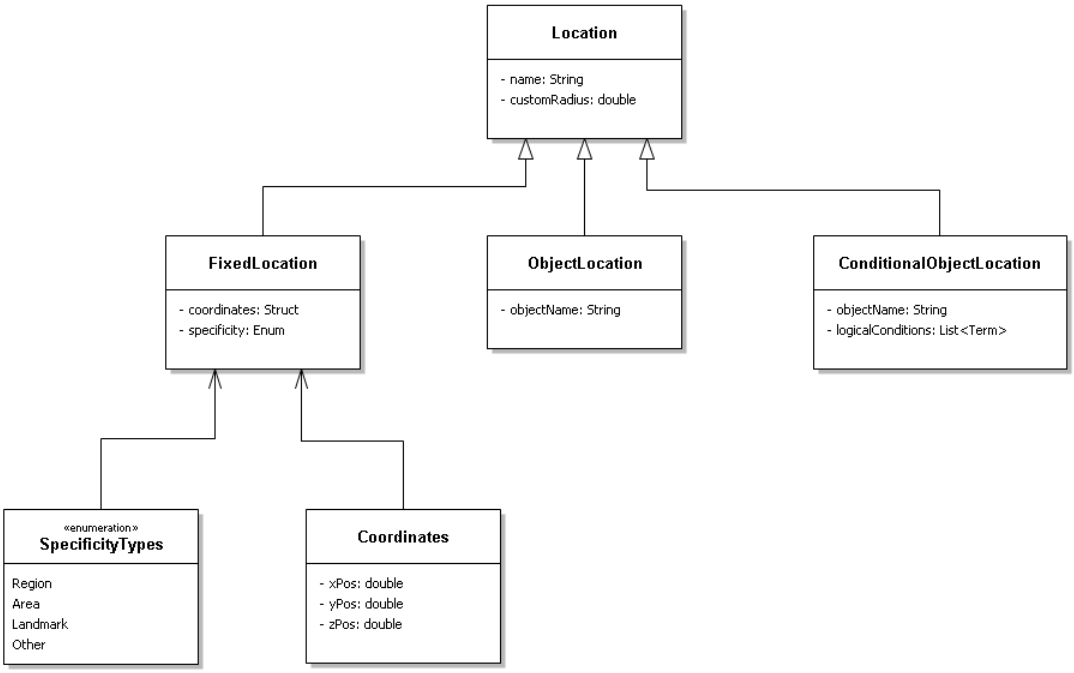


Figure 4.12: Location - Class Diagram

The first type of location is defined in terms of a *fixed* point. This type of location consists of a name, a set of virtual world co-ordinates, a level of specificity, and an optional custom radius for locations that do not fit the defined levels of specificity. The levels of specificity – region, area and landmark – have been chosen based on Hall's work on *proxemics* [36]. In this anthropologically-focused work, Hall defined four spatial delineations: intimate, personal, social and public. Each distance has an associated close and far phase, ranging from six inches (near intimate) to 25 feet (far public). Using this work as a reference point, I have defined three levels of specificity as shown in Table 4.6.

Table 4.6: Levels of Specificity

Level	**Radius (m)**	**Proxemics Equivalent**
Region	10	Public (far)
Area	2	Social (close)
Landmark	1	Personal (far)

At the most general, one can consider locations within a scenario at the *region* level. In a healthcare context, for example, such locations might include "at the accident scene", "in the hospital", or "at the patient's house". These regions may then be used to give the instructor a broad sense of each participant's location as they proceed through the scenario.

At a more precise level, one can consider relevant *areas* for each scenario. Returning once again to the healthcare context, one might be interested in locations such as “in the parking lot behind the hospital”, “inside Trauma Room 1”, or “near the victim at the accident scene”. The areas, thus, give the instructor a more precise sense of each participant’s location.

At the most fine-grained level of specification, an instructor might be interested in a participant’s proximity to certain key objects or features. In this case, one can define *landmarks*, that indicate precisely-defined places of interest. These landmarks, in a healthcare context, might include “at the heart monitor”, “at the victim’s car”, or “at the ER department reception desk”. The association of a participant’s co-ordinates with a landmark location is quite rigid, as implied by the defined radius of 1 metre. Note that, while a landmark may seem similar to an object-based location, there are two key differences. First, a landmark may refer to a non-interactive entity within the scenario – that is, a “prop” that does not have any impact on the functionality of the scenario – while an object-based location must refer to an interactive object included in the scenario definition. Second, a landmark’s location is fixed – that is, a specific set of co-ordinates – while an object-based location will vary as the object’s position changes.

These locations of interest can be defined hierarchically. That is, *regions* can contain one or more *areas*, which in turn might contain several *landmarks*. Therefore, depending on the level of analytical granularity required, one can “zoom in” on a particular location within a scenario to determine each participant’s movement within that area.

The second type of location is defined in terms of an *object*. The definition for this type of location is dependent on the named object, and the location itself will vary depending on where the object is located within the scenario. For object-based locations, the default specificity is equivalent to the *landmark* definition given previously. However, as with fixed locations, one can also define a custom radius for an object-based location, to provide a different level of specificity, as required. By tracking a participant’s proximity to an object (via object-based or conditional object locations), one can obtain a preliminary idea of the participant’s *intent* to use that object. That is, one can assume that a participant who moves to within a certain distance of an object intends to use that object, and one can analyze the correlation between movement to an object-based location and interaction with that object to determine the degree to which a participant followed through with his or her (implied) intent to use an object.

The third type of location, a *conditional object* location, is defined in terms of an *object* and a set of *conditions* which must be met for the location to be considered relevant. This additional set of criteria provides the capability to track a participant’s proximity to a given object at certain key points within a scenario. For instance, one might only be interested in a participant’s proximity to a stretcher when a patient has been placed on the stretcher. These conditions are defined using the same *term* entities that were used to define constraints (discussed in Section 4.3.4). As with object-based locations, the conditional object location is, by default, defined with a level of specificity equivalent to a landmark, and this can be altered as necessary using the custom radius parameter.

Finally, it is worth noting that the concept of locations in a scenario is related to a broader issue of representing a scenario’s setting(s) in a virtual world. That is, the locations are identified using virtual world co-ordinates; these co-ordinates, by implication, are part of a larger setting within the virtual world. This setting might include elements such as scenery (trees, hills, etc.), buildings, and inactive “prop” objects. Because these setting elements must be implemented entirely within the chosen virtual world, and do not have any direct relationship to the definition or enactment of the scenario, they are not included in the MeRiTS-based scenario definition.

4.3.7 Workflow Implementation Process

The workflows are implemented using Business Process Execution Language (BPEL), an executable workflow definition language [90]. BPEL, generally speaking, consists of three types of constructs. First, there are constructs that connect other web services, either by invoking them, waiting for invocation, or creating an invocation-response message exchange. Second, there are program control constructs - such as if statements, variable assignment, and loops - that form the core of most imperative programming languages (e.g., Java, C++). Third, exception and error handling constructs allow the workflow to recover from, and in some cases even undo the results of, invalid states and erroneous input. Other important parts of a BPEL workflow include the definition of the workflow namespace, the address and port type of the workflow (to allow external web services to access the workflow), and the definition of internal variables used by the workflow for data storage and exchange.

A BPEL workflow is represented as an XML document and can be executed by a BPEL engine, of which there are several[1]. However, there is no official graphical representation for a BPEL workflow, although Business Process Modeling Notation (BPMN) is a graphical representation standard that can be used for this purpose.

BPEL was chosen because its constructs are sufficient for expressing the behaviour that needs to be defined, it is a well-established workflow definition and execution language, and there are several mature tools for defining and executing BPEL workflows within a web services context. The BPEL language includes elements that correspond to each component of the workflow model (described in Section 4.3.3), as shown in Table 4.7.

Table 4.7: Relationship between model and BPEL

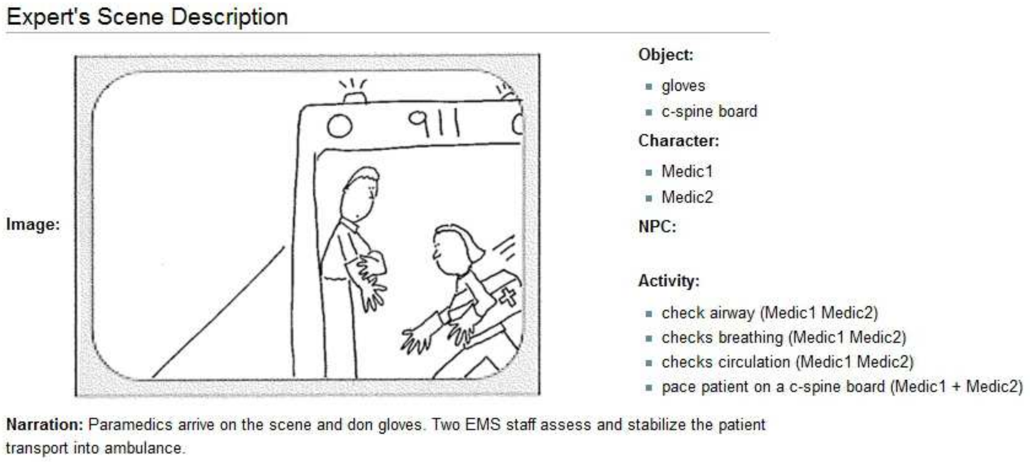
Workflow Component	**BPEL element**
Input variable	Part of the input message
Sequence	Sequence
Condition	If
Intialize	Receive
Assignment	Assign
Finish	Reply
Output variable	Comma-separated list of name-value pairs
Result	Comma-separated list of result-value pairs

While BPEL is simpler than a programming language such as C++ or Java, it is likely to be too technically challenging for most content experts (e.g., instructors) to use directly in defining scenarios. Although graphical tools for designing BPEL workflows – of which there are several – make some of the details easier to manage, these tools still rely on the user understanding the underlying programming concepts, an assumption that is unrealistic for non-technical users. Indeed, West et al. have reported that, despite the availabilty of

[1] Some examples include the Oracle BPEL Process Manager (http://www.oracle.com/technetwork/middleware/bpel/), IBM Websphere Process Server (http://www-306.ibm.com/software/integration/wps/), Microsoft BizTalk Server (http://www.microsoft.com/biztalk/en/us/default.aspx), and Apache ODE (http://ode.apache.org/)

process modelling tools, actually eliciting a process from stakeholders and formalizing that process remains a challenge [87]. To address this challenge, I have experimented with some tools and processes to enable collaborative workflow definition by technical users, while allowing instructors to retain control over the meaning of the workflow. The details of these approaches are described below.

One approach I have tried is an iterative design process, involving collaboration between technical experts and instructors. In the first part of this process, the instructors describe the workflow, using whatever techniques are most natural for them (e.g., diagrams, sketches, descriptive prose). To facilitate the sharing of this information, my colleagues and I created a wiki (that is, a collaboratively-edited webspace) where content experts can post their workflow description. In the next step of the process, technical experts use this initial workflow description to elicit a set of entities and actions. Using the wiki, these entities and actions can then be made available to the content experts for validation. A part of one such workflow is shown in Figure 4.13. Then, based on these validated entities and actions, the technical experts create an abstract representation of the workflow, which can be understood and validated by the content experts. Finally, an executable workflow can be developed by the technical expert, based on the validated abstract workflow. This approach is quite informal, technologically, as it uses a storytelling-based technique to elicit the workflow, which is gradually converted into a useable artifact.



Expert's Scene Description

Image:

Object:
- gloves
- c-spine board

Character:
- Medic1
- Medic2

NPC:

Activity:
- check airway (Medic1 Medic2)
- checks breathing (Medic1 Medic2)
- checks circulation (Medic1 Medic2)
- pace patient on a c-spine board (Medic1 + Medic2)

Narration: Paramedics arrive on the scene and don gloves. Two EMS staff assess and stabilize the patient transport into ambulance.

Figure 4.13: Workflow Description Wiki

Another approach I have explored is the use of a graphical workflow modelling tool. One such tool is Visual Understanding Environment (VUE)[2], developed by researchers at Tufts. VUE allows the user to create a workflow by defining nodes (represented using rectangles) and then creating connections between them (represented using lines between the rectangles). One advantage of VUE is that, in addition to allowing users to export workflows to several image formats (which can then be easily uploaded to a wiki), the tool also creates an XML-based description of the workflow. This description, in turn, can then be automatically converted into a BPEL workflow, as BPEL is also an XML-based format. That is, the nodes and connections of the VUE representation are converted to a set of *sequence*, *if*, and *assign* elements in BPEL that represent an equivalent workflow. The resulting generated workflow may require some fine-tuning by technical experts, but

[2] See http://vue.tufts.edu/

it provides a very good starting point for the development of an executable workflow. The VUE tool is shown in Figure 4.14. This approach is much more technologically formal than the iterative design process, in that a usable workflow artifact is produced directly through the VUE tool.

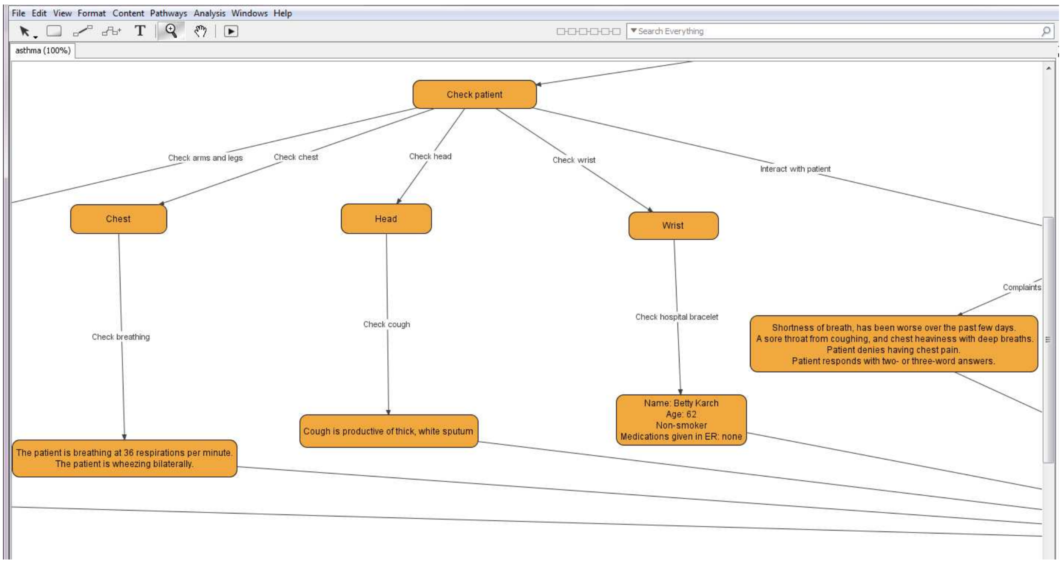

File Edit View Format Content Pathways Analysis Windows Help
Search Everything
asthma (100%)
Check patient
Check arms and legs
Check chest
Check head
Check wrist
Interact with patient
Chest
Head
Wrist
Complaints
Check breathing
Check cough
Check hospital bracelet
Shortness of breath, has been worse over the past few days.
A sore throat from coughing, and chest heaviness with deep breaths.
Patient denies having chest pain.
Patient responds with two- or three-word answers.
Name: Betty Karch
Age: 62
Non-smoker
Medications given in ER: none
Cough is productive of thick, white sputum
The patient is breathing at 36 respirations per minute.
The patient is wheezing bilaterally.

Figure 4.14: A simple workflow modelled in VUE

4.4 The Virtual World Client

The virtual world client allows the participant to experience the workflow in an interactive, immersive way. A screenshot of a virtual world scenario is shown in Figure 4.15. The virtual world includes a variety of elements, which I present in three parts: elements related to the scenario setting, avatars used by the scenario participants, and interactive objects. An additional component of the client is the behaviour recording device, which is described in Section 3.3. Each of these categories are discussed in the following sections, with particular attention paid to the implementation of the behaviour of interactive objects, which is one of the most complex parts of the virtual world client.

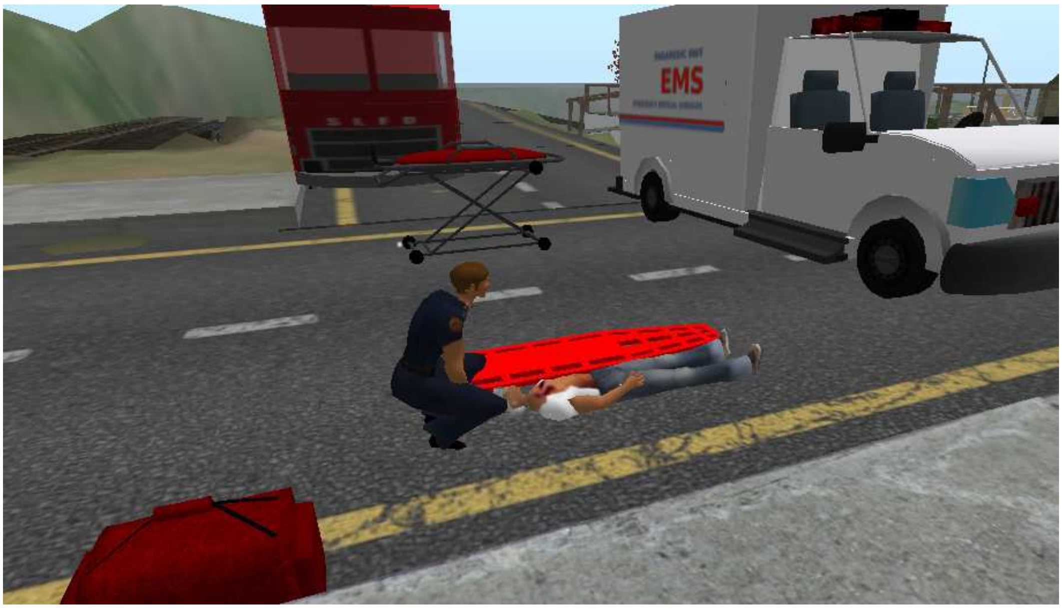


Figure 4.15: Virtual World Client

4.4.1 Creating Settings Within the Virtual World

When creating a scenario in a virtual world, one must consider the setting in which the scenario will take place. That is, one must create a setting that is detailed enough to give the participant a feeling of *presence* as they experience the scenario. This setting will typically consist of a landscape (e.g., hills, rivers, and road), background elements (e.g., buildings and trees), inactive “props”, and other entities as needed (e.g., posters on walls).

Although this setting does not have a direct impact on the execution of the scenario, in terms of interactive objects, workflows, and the like, the setting is related to the locations that are defined for a scenario (as mentioned in Section 4.3.6). Specifically, one must ensure that the meaningful locations for a scenario are appropriately represented in the setting, and interactive objects are not obscured by background elements or props.

I make a distinction here between *interactive objects* and *props*. The former are virtual world entities that possess interactive capabilities (as described in Section 4.4.3), and may trigger the execution of associated workflows. The latter are inert virtual world representations of real-world entities, with no interactive capabilities. To sum up, while both types of entities must be appropriately represented in the virtual world (an issue which was discussed in Section 4.4.3), the participant cannot interact with props in any way, and thus their impact on the scenario is quite minimal.

In my running patient rescue example, the setting might include the elements listed in Table 4.8.

Table 4.8: Setting for Patient Rescue Example

Setting Entity Type	**Entities for Setting**
Landscape	Sidewalk, road, hilly terrain
Background elements	Nearby buildings and stores, bus shelter
Props	Parking meters, garbage cans, parked cars
Other entities	Poster in bus shelter

While none of the above items are directly related to the successful completion of the patient rescue scenario, they all contribute to providing the participant with a sense of being in a “real” environment while he or she is going through the scenario.

4.4.2 Avatars

Each participant interacts with the virtual world via an *avatar*, which was described at a high level in Chapter 1. To briefly summarize, an avatar is a customizable alter ego through which the participant can explore the virtual world, communicate with other participants (via their avatars) and interact with objects. To take advantage of the capabilities offered by the MeRiTS framework, the avatars must be equipped with additional, customized items. One of these is the behaviour recording device, which was described in Section 3.3. Another is the heads-up display (HUD) that has been developed to enable participants to hold items, which is described in Section 4.4.3.

For each scenario, meanwhile, the avatars must be customized to appropriately convey the role(s) being taken on by the scenario participants and the overall context of the scenario. For instance, in a medical context, all of the nurses’ avatars would wear scrubs, while the doctors’ avatars would wear white coats. Furthermore, all medical staff would be equipped with a stethoscope, as a standard piece of worn medical equipment.

Finally, depending on the setup of the virtual world, and the requirements for a particular educational context or scenario, the avatars might need to be granted specific group membership or access rights in order to participate in the scenario. In Second Life, for instance, many educational areas are developed on private islands, which can only be accessed by members of a certain group. This restriction means that each participant’s avatar must be invited to the appropriate group in order to access the island and participate in the scenario. A related consideration is that these groups can (and usually should) be defined with very limited modification rights, so that members of the group can *participate* in the scenario, but cannot *change* the entities in the scenario, either intentionally or accidentally.

4.4.3 Interactive Objects

Interactive objects are an essential part of a virtual world scenario, since they encapsulate a broad range of interactive capabilities. These objects may have a *simple* or *composite* structure. For simple objects, my main concern is the representation of the object in the virtual world. For composite objects, I had to consider the ways in which the object’s constituent parts could be combined and structured.

The other challenge that must be considered is implementing an object's behaviour, as specified by the associated actions in the corresponding workflow. This, typically, involves the addition of native code to the virtual world representation of the object. This code calls web services responsible for interpreting the result of the action, both in terms of any immediate changes to the object's appearance, as well as the impact of the action on the process workflow.

These challenges are described in the following subsections.

Object Structure

An interactive object may have either a *simple* or *composite* structure. A *simple* object is treated as a single, indivisible entity in the virtual world. For a simple object, then, the composition of the object is clearly defined, and the structural challenge lies in representing the object appropriately. Depending on the virtual world that is chosen, one may be able to create the physical representation using external 3D modelling software, using in-world 3D modelling tools, or by using 3D models created by other users. I have found, through development experience in a virtual world, that each approach has benefits and drawbacks. In my experience, using models created by other users tends to offer the best tradeoff between development effort and object realism. However, there are also certainly situations where the other two approaches are required, and result in realistic, customized object representations.

A *composite* object, on the other hand, is made up of multiple objects, one of which is defined as a "parent". This composition of objects may be modified dynamically, and is particularly important when multiple objects move from one location to another. For example, consider an object A that has been placed on top of another object B. When B is moved to a new location, A should travel along with B in a synchronized manner.

I have developed a mechanism that solves this problem by merging the relevant objects into a single composite object, and then moving the new composite object to the given destination. This mechanism uses the "composite" design pattern, where all objects – whether they are simple or composite – are handled in a similar manner by an algorithm or process. For details of the mechanism's implementation, see Appendix D.

The mechanism allows much of the disassembly and re-assembly process to occur in parallel among the parent and child entities within an object, while maintaining the necessary organization and synchronization through a relay entity. This parallelization is crucial to the implementation of the mechanism, since the total number of entities being detached and re-attached can reach into the hundreds. Given the number of entities involved, the mechanism would be unacceptably slow if each of these operations occured sequentially.

Note that, while the disassembly and re-assembly process is in progress, the affected objects are quite "brittle", in that their composition may not be correct in terms of either their original or new parent entities. To address this issue, each object can, through the use of "in transition" and "finished" statuses, indicate whether it is in this brittle state or ready for use in the scenario.

Object Behaviour

Interacting with an object in a virtual world can pose a variety of challenges, depending on the affordances and capabilities provided by the virtual world. The issue of interacting with an object, broadly speaking, can be divided into two phases: enabling the participant to interact with an object in a variety of ways, and implementing the required response to each of these interaction possibilities. Within these two phases, I have separated the participant-object interaction process into five steps:

1. Participant indicates intent to interact with object
2. Object responds with interaction possibilities
3. Participant chooses an action
4. Object processes action, potentially relaying processing request to scenario execution engine
5. Object conveys result of action to participant through dialog box, sound clip, or other response mechanism

The first three steps of this process occur within the first phase (enabling the participant to interact as desired with the object), while the last two steps are part of the second phase (responding to the participant's chosen action). In the following paragraphs, I will describe this process in detail, beginning with enabling the participant to indicate his or her intent to interact with the object.

In Second Life, the default affordances provided for interacting with an object are as follows: an object can be touched, worn, sat on, driven, or taken by a player. These affordances are not necessarily appropriate for all scenarios, and more complex kinds of interaction, such as carrying or using an object, can be quite difficult to represent. In summary, the default affordances provided were not, in this case, sufficiently varied or flexible to support the activities that make up the first phase of the process.

Due to of the inadequacy of the default affordances, I needed to develop an interaction mechanism that could present multiple interaction options to the participant in an intuitive, flexible way. Second Life offers two types of interaction, by default: left-clicking (or single-clicking on a one-button mouse) on an object will "touch" the object and trigger the code associated with this interaction. Right-clicking, on the other hand, will bring up a series of pie-chart menus with additional options, such as "sit", "pay" or "take". While making use of the additional menus was an appealing option, initially, I discovered that this was not a good choice, from a usability point of view. When interacting with an object, participants were not sure whether they were supposed to left- or right-click, and in the latter case, they were not sure which option, of the dozens that were presented, they were supposed to pick. To deal with this lack of clarity, I employed a convention where participants would *always* left-click (that is, touch) an object in order to begin interacting with it, and any additional interaction options would be presented through dialog boxes. This approach, which addressed all three steps in the first phase of the interaction process, ensured that participants would always know how to begin interacting with an object, and subsequent choices would be clear, relevant, and manageable.

Another challenge that I faced was dealing with objects that, in real life, must be picked up and used. After several attempts, the solution I settled on involved creating an auxiliary user interface component, referred to in Second Life as a heads-up display (HUD), that the participants could use to pick up objects. This HUD communicates with the objects that might be affected by an item that the participant has picked up (e.g., a patient object, when the participant picks up medical equipment), and informs the object(s) which item (if any) the participant is currently holding. Each of the affected objects changes its interface dialog process accordingly, so that it first asks the participant if he or she wants to use the held item. If the participant decides not to use the held item, then the affected object shows the participant the usual interaction dialog box. Although this issue does not directly map to the phases and steps outlined above, it has a significant impact on the steps in the first phase.

Once the method of interacting with an object was determined, I then had to develop with an approach for actually imbuing that object with interactive capabilities. One of the chief challenges of implementing interactive objects in a virtual world is balancing the need for flexibility – that is, the ways in which a participant can interact with an object – with consistency in the code and architecture.

In the case of my Second Life-based implementation, I have implemented a procedure for interactive objects which balances these concerns. The action choices and results are defined in a block of code at the start of the code, while the rest of the code is consistent across all objects. The action choices, results, and other relevant variables are shown in Algorithm 1.

Algorithm 1 Pseudocode for Interactive Objects

```
1: $objectName \leftarrow$ the name of the object
2: $parentName \leftarrow$ the name of the object's parent, if it exists
3: $dialogChannel \leftarrow$ channel for messages sent to or from object's dialog box
4: $publicChannel \leftarrow$ channel for interaction between objects, held item HUD
5: $interactionChoices \leftarrow$ array of interaction choices (including via held items)
6: $interactionResponseTypes \leftarrow$ array of response types for each interaction
7: $interactionResponses \leftarrow$ array of responses for each interaction
8: $validItems \leftarrow$ array of items that are relevant to this object
9: $heldItems \leftarrow$ associative array of items held by each avatar
```

In this first segment, I have defined the variables that are used in the code. These variables can be separated into a few categories. First, there are variables that store general information about the object (lines 1-2): its name and the name of its parent, if applicable. The *parentName* variable is required for objects that are part of a hierarchy of objects, as described in Section 4.3.1.

Second, there are *channel* variables (lines 3-4) that route messages sent between the dialog box and the LISTEN function, and between each object and the held items HUD of the participant interacting with the object.

Third, there are the *interaction* variables (lines 5-7), which define the ways in which the participant can interact with this object, the type for each interaction response, and the content of each interaction response. Depending on the interaction response type, the corresponding response content entry will store slightly different information, as shown in Table 4.9.

Table 4.9: Interaction Response Information

Interaction type	**Response Information**
show message	The content of the message
play sound	The sound clip to be played
change appearance	The new texture for the object
rotate object	The angle at which the object should be rotated
call scenario execution engine	The parameters of the action to send to the server

For interactions whose response is a request to the scenario execution engine, the result corresponds to an *action* that has been specified in the scenario definition. The type and parameter for this action are retrieved from the $interactionResponses$ array, embedded in a HTTP request (via the `llHttpRequest` function) and sent to the scenario execution engine.

Fourth, there are the *item* variables, which manage the relationship between the item held by the participant interacting the object and the options presented by the object.

Algorithm 2 Pseudocode for Interactive Objects (TOUCH function)

```
function TOUCH
    if distance($avatarPos, objectPos$) $\geq distanceThreshold$ then
        show dialog (“You are out of range”, [“OK”], $dialogChannel$)
    else
        if avatar is holding a relevant item then
            $dialogMessage \leftarrow$ “Do you want to use the $item$ or interact with the $object$?”
            show dialog ($dialogMessage$, [$item$, $object$], $dialogChannel$)
        else
            $dialogMessage \leftarrow$ “What action do you want to take?”
            show dialog ($dialogMessage, interactionChoices, dialogChannel$)
        end if
    end if
end function
```

The TOUCH function, which is called when a participant touches the object, acts as a first point of contact for the participant. It ensures that the participant is close enough to the object to be able to realistically interact with it, and then presents an initial set of options to the participant. The participant's response is handled by the LISTEN function. If the participant is holding a relevant item, then the set of options is restricted to either using the item or interacting directly with the object. If the participant chooses to use the held item, then the chosen item passed into the LISTEN function. If the participant chooses to interact directly with the object, then the parameter is set to the name of the object.

Otherwise, the options are drawn from the *actionChoices* array, and the LISTEN function parameter is set to the chosen action. Note that the TOUCH function corresponds to step 1 (indicating intent), and potentially steps 2 and 3 (presenting options and making a choice) in the object interaction process.

The LISTEN function processes the participant's dialog box responses, and also deals with messages from other objects. Note that the function is a callback that is defined by the Second Life scripting language, and is unrelated to the **listen** behaviour that was defined in the ABM (in Table 3.1). While this function may seem to address the behaviour encapsulated by the **listen** behaviour, this is simply an unfortunate syntactic coincidence. The LISTEN function presented here “listens” for a variety of messages, both from the participant and from other objects in the world, and thus forms an essential part of the interaction script. The participant's responses fall into a few broad categories:

First, the participant might have chosen to interact with the object's parent, if the object is part of a hierarchy. In this case, the interaction request is passed to the parent object by triggering a call to the parent object's LINKEDMESSAGE function, which is described below.

Algorithm 3 Pseudocode for Interactive Objects (LISTEN function)

```
function LISTEN($channel, message$)
    if $channel == dialogChannel$ then
        if $message$ indicates that participant wants to interact with parent item then
            send linked message to $parent$
        else if $message$ indicates that participant wants to interact with object then
            $dialogMessage \leftarrow$ “What action do you want to take?”
            show dialog ($dialogMessage, interactionChoices, dialogChannel$)
        else
            $responseContent \leftarrow interactionResponses[message]$
            $responseType \leftarrow interactionResponseTypes[message]$
            if $responseType ==$ show a message then
                show dialog ($resultContent, actionChoices, dialogChannel$)
            else if $responseType ==$ play a sound then
                play sound ($responseContent$)
            else if $responseType ==$ change the object’s appearance then
                change texture ($responseContent$)
            else if $responseType ==$ change the object’s rotation then
                change the object’s rotation by $responseContent$ degrees
            else if $responseType ==$ send message to scenario execution engine then
                $URL \leftarrow$ address of scenario execution engine
                append $responseContent$ to $URL$ as GET parameters
                HTTPrequest($URL$)
            end if
        end if
    end if
    if $channel == publicChannel$ then
        if $message$ indicates item has been picked up or dropped then
            update $heldItems$ array accordingly
        end if
    end if
end function
```

Next, if the participant is holding a relevant item (e.g., a stethoscope, when interacting with the patient’s chest), then the participant may have been offered the option of using the item, and instead chosen to interact with the object directly. In this case, the participant is shown a follow-up dialog menu that asks them to select an action from the *actionChoices* array. In either case, these parts of the function address steps 2 and 3 of the overall process.

If the LISTEN function has been called in response to the participant’s action selection, the object responds accordingly, using the *actionResults* associative array (which includes responses to the participant choosing to use a held item). These object responses are written generically, so that the required information for a specific response (a message in a dialog box, for example, or a sound clip) is supplied via the associative array. Furthermore, when the response required by the participant’s choice involves processing the action using a workflow, the feedback is not shown through the LISTEN function. In this case, the object makes a request of the scenario execution engine, using the content in the *actionResults* array to determine the request parameters. The HTTPResponse function (described in Algorithm 5) is responsible for handling the response to this request, and thus providing the participant with the necessary feedback. Whether it is the LISTEN or the HTTPRESPONSE function that ultimately handles the action response, this portion of the LISTEN function is responsible for dealing with step 4 (action processing) of the participant-object interaction process. If the response is contained within this section of code, then it is addressing step 5 (conveying the action result), as well. Otherwise, this step is handled by the HTTPRESPONSE function.

Finally, if the message being processed by the LISTEN function is from another object (and thus being relayed on *publicChannel*), then the message must be parsed and processed accordingly. For the purposes of this description, the only relevant messages that may be received are from the held/worn item HUDs, informing the object that a participant has decided to hold, drop, wear, or discard a given item.

Algorithm 4 Pseudocode for Interactive Objects (LINKEDMESSAGE function)

```
54: function LINKEDMESSAGE($message$)
55:     if $message == objectName$ then
56:         $dialogMessage \leftarrow$ "What action do you want to take?"
57:         show dialog ($dialogMessage, interactionChoices, dialogChannel$)
58:     end if
59: end function
```

The LINKEDMESSAGE is used for hierarchical message-passing. Through the LINKEDMESSAGE function, an action request that has been delegated to a parent object can be received by that object and processed accordingly. This function, therefore, deals with steps 2 and 3 of the overall process.

Algorithm 5 Pseudocode for Interactive Objects (HTTPRESPONSE function)

```
60: function HTTPRESPONSE($body$)
61:     $responseMessage \leftarrow$ portion of $body$ containing returned message (if any)
62:     $responseAction \leftarrow$ portion of $body$ containing returned action (if any)
63:     if $responseMessage \neq null$ then
64:         show dialog ($responseMessage, emptylist, dialogChannel$)
65:     end if
66:     if $responseAction \neq null$ then
67:         take action based on content of $responseAction$
68:     end if
69: end function
```

Finally, the HTTPRESPONSE function handles the result of a HTTP request to the MeRiTS server. This request is initiated from the LISTEN function (on line 51) in order to process an action using the workflow. The returned HTTP result represents the feedback from the workflow on the result of the action. In most cases, the result is simply a message, which is shown via a dialog box. In some situations, however, the result may trigger more complex responses, such as coordinated movement among several objects. In this case, the response includes a *responseAction* parameter, which contains the information necessary to execute the required response. In any case, the HTTPRESPONSE function deals with step 5 of the object interaction process.

In terms of the running example, I have defined the variables for the patient object – the complex parent entity and a child entity (an arm) – and the stretcher as shown in Algorithm 6.

Algorithm 6 Pseudocode for Scenario Objects

```
// Patient Object (parent entity)
$objectName \leftarrow$ “Patient”
$parentName \leftarrow$ “”
$dialogChannel \leftarrow 1$
$publicChannel \leftarrow 100$
$interactionChoices \leftarrow$ [“Roll over”, “Move to stretcher”]
$interactionResponseTypes \leftarrow$ [“call scenario engine”, “call scenario engine”]
$interactionResponses \leftarrow$ [“move, roll”, “move, stretcher”]
// Patient Object (arm entity)
$objectName \leftarrow$ “Patient - Right arm”
$parentName \leftarrow$ “Patient”
$dialogChannel \leftarrow 2$
$publicChannel \leftarrow 100$
$interactionChoices \leftarrow$ [“Check pulse”, “Move arm”, “Work with patient”]
$interactionResponseTypes \leftarrow$ [“call scenario engine”, “rotate object”, “call parent”]
$interactionResponses \leftarrow$ [“assess, pulse”, “30”, “Patient”]
// Stretcher Object
$objectName \leftarrow$ “Stretcher”
$parentName \leftarrow$ “”
$dialogChannel \leftarrow 3$
$publicChannel \leftarrow 100$
$interactionChoices \leftarrow$ [“Push to patient”, “Push to ambulance”]
$interactionResponseTypes \leftarrow$ [“call scenario engine”, “call scenario engine”]
$interactionResponses \leftarrow$ [“push, patient”, “push, ambulance”]
```

Given these implementation details, I present a few sample execution paths through the pseudocode. Note that this pseudocode focuses on the way that the action is processed by the relevant object, and does not include the mechanism by which the action is recorded. For details on the recording mechanism, see Section 3.3. Also, the overall interaction process step, when relevant, is indicated in italicized parentheses.

First, consider the case of a participant deciding to move the patient’s arm. This interaction would proceed as follows:

1. The user clicks on the patient’s arm *(step 1)*.

2. The TOUCH function is called, and the user is shown a dialog box with the options contained in the $interactionChoices$ array *(step 2)*.

3. The user selects the “Move arm” option, and this choice is passed to the LISTEN function *(step 3)*.

4. The LISTEN function processes the “Move arm” message as follows *(step 4)*:

    (a) the $responseContent$ is given the value in $interactionResponses$[“Move arm”] (“rotate object”)

    (b) the $responseType$ is given the value in $interactionResponseTypes$[“Move arm”] (“30”)
    (c) since $responseType ==$ “rotate object”, the object is rotated by 30 degrees *(step 5)*

Second, consider a slightly more complex case of the participant deciding to roll the patient over. In this case, I have assumed that the patient is face-down, and the participant has decided to roll the patient face-up. This interaction would proceed as follows:

1. The user clicks on the patient *(step 1)*.
2. The TOUCH function is called, and the user is shown a dialog box with the options contained in the $interactionChoices$ array *(step 2)*.
3. The user selects the “Roll over” option, and this choice is passed to the LISTEN function *(step 3)*.
4. The LISTEN function processes the “Roll over” message as follows *(step 4)*:
    (a) the $responseContent$ is given the value in $interactionResponses$[“Roll over”] (“call scenario engine”)
    (b) the $responseType$ is given the value in $interactionResponseTypes$[“Roll over”] (“move, roll”)
    (c) since $responseType ==$ “call scenario engine”, a HTTP request is sent to the scenario execution engine with the parameters “move” and “roll”.
5. The scenario execution engine processes the request, and eventually returns a result.
6. The HTTPRESPONSE function is called, and the content of the result is parsed as follows *(step 5)*:
    (a) The result contains a response message: “Patient rolled over successfully.” This message is shown to the user via a dialog box.
    (b) The result contains an action: “Roll patient over”. The patient is then rotated by 180 degrees around its centre.

Finally, consider a case where the participant initially clicks on the arm, and then decides to move the patient to the stretcher. This interaction would proceed as follows:

1. The user clicks on the patient’s arm *(step 1)*.
2. The TOUCH function is called, and the user is shown a dialog box with the options contained in the $interactionChoices$ array *(step 2)*.
3. The user selects the “Work with patient” option, and this choice is passed to the LISTEN function *(step 3)*.
4. The LISTEN function determines that the user wants to work with the parent, and calls the LINKMESSAGE function.
5. The parent entity receives the message via the LINKEDMESSAGE function, and shows the user a dialog box with the options for interacting with the parent *(step 2)*.

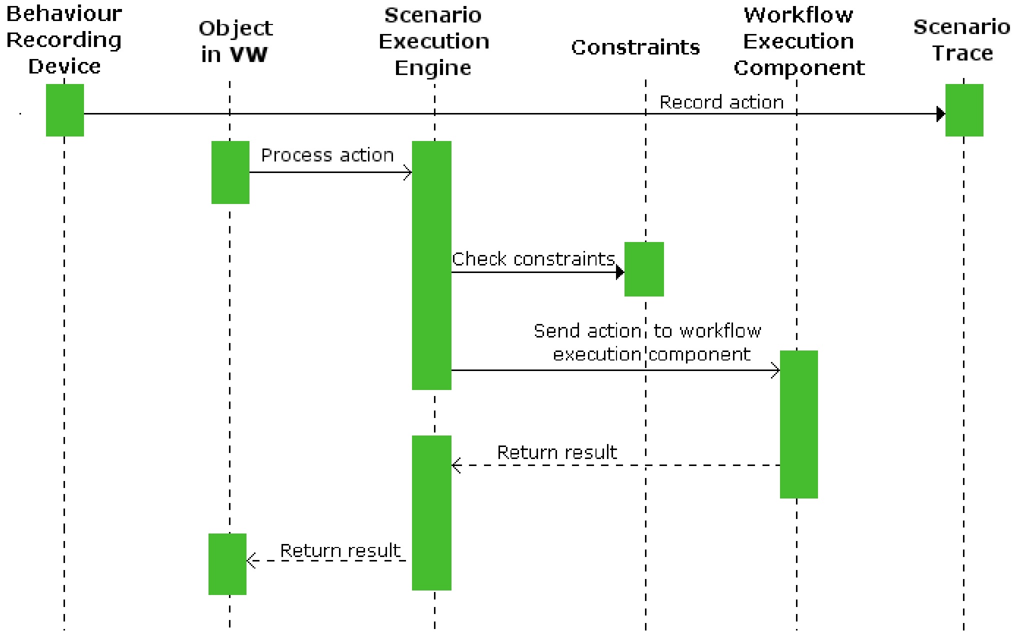
6. The user selects the "Move to stretcher" option, and this choice is passed to the LISTEN function within the parent entity *(step 3)*.

7. The LISTEN function then processes the message, as in the previous case *(steps 4 and 5)*.

4.5 Scenario Execution Engine

This section describes the role that is played by the scenario execution engine in linking the virtual world client to the scenario definition, and in recording the detected action.

When a participant takes an action in the virtual world, the interaction between the various components proceeds as shown in Figure 4.16, and described in the following sequence. There is some overlap between this description and the discussion of object interaction in Section 4.4.3, although the two sections deal with the issue of processing a participant's action from different perspectives. Specifically, step 1 in the sequence below corresponds to steps 1-3 of the object interaction process described previously, and step 2a below corresponds to the "relay request to the scenario execution engine" possibility within step 4 of the interaction process. Steps 3-5 below, meanwhile, describe in detail the mechanism (which is omitted from the interaction process) by which the relayed request is handled by the engine. Step 6 below, finally, corresponds to the last step in the interaction process.



Behaviour Recording Device
Object in VW
Scenario Execution Engine
Constraints
Workflow Execution Component
Scenario Trace
Record action
Process action
Check constraints
Send action to workflow execution component
Return result
Return result

Figure 4.16: Sequence Diagram

1. Participant takes an action in the virtual world (VW)

2. Action is processed simultaneously by behaviour recording device and VW object:

   (a) VW object sends message to scenario execution engine describing action

   (b) Behaviour recording device sends message to behaviour recording service describing the action, in terms of ABM

3. The relevant components process their respective messages:

   (a) The scenario execution engine processes the action in three steps:

      i. Values for the relevant *variables* are retrieved from the DB

      ii. Engine ensures that action does not violate any *constraints*

      iii. If not, action is passed to workflow execution component

   (b) Behaviour recording service sends behaviour description to scenario trace DB

4. Workflow execution component executes object workflow with respect to the action:

   (a) Workflow execution component follows required conditional branch for action

   (b) Required workflow components are executed, with results or updated variable values added to *result* string

   (c) The *result* string is returned to the scenario execution engine

5. Scenario execution engine interprets result from workflow execution component

   (a) Any variable values are extracted from the result string, DB updated accordingly

   (b) Resulting message and action (if any) extracted and returned to object in VW

6. VW object receives resulting message and action (if any), and responds accordingly

The scenario execution engine receives a message from the interactive object indicating the action that was taken and the identity of the actor. The engine then retrieves the current variable values from the MeRiTS database and, using these values, checks to see if the given action violates any constraints. If a feasibility constraint is violated, then the engine returns an appropriate message to the virtual world object. If an advisibility constraint is violated, then the engine stores the message as a warning to be appended to the message portion of the action result. If no feasibility constraints are violated, the engine sends the action to the workflow execution component (which executes the workflow for the relevant object) for processing. The workflow execution component, following the approach described at the end of Section 4.3.3, first executes the workflow's initial conditional element(s) to determine which components of the workflow to execute. The relevant components are then executed, with any variable values or results (either messages or actions) appended to a continually-updated *result* string. Finally, once all the components have been executed, the *result* string is returned to the scenario execution engine. This engine, upon receiving the *result* string, extracts the result message/action, updates variable values as necessary, appends any warning messages to the result message, and returns the result message/action to the virtual world object.

The action, meanwhile, is also sent to the behaviour recording web service, described in terms of the ABM (which was discussed in Chapter 3). This web service, in turn, stores each behaviour instance as an entry in a database table, whose schema is shown in Table 4.10. This table collects the recorded behaviours of all simulation participants, which enables the analysis of both an individual's behaviour and of group behaviour through the Scenario Trace Analysis Toolkit (STAT) component. A sample set of database table entries is shown in Appendix E.

Table 4.10: Structure for Storing Scenario Traces

Field Name	**Description**
Actor	The ID of the avatar that executed the behaviour
Behaviour Type	The type of behaviour (e.g., move, chat, interaction)
Parameters	The behaviour parameters movement: type, start location, end location textChat: message gesture: gesture description hold/drop: object interaction: message, options interaction response: choice, response
Timestamp	The time at which the behaviour occurred
Scenario	An identifier for the scenario in which the behaviour occurred

4.6 Scenario Trace Analysis Toolkit

The Scenario Trace Analysis Toolkit (STAT) builds upon the functionality of the behaviour recording component, which is described in Section 3.3. While the scenario trace generated by the behaviour recording component is a crucial starting point for educational evaluation, it is not, on its own, sufficient. The scenario trace is composed of “raw” recorded behaviour entries, as defined by the ABM model. These entries require further parsing, aggregation and analysis before they can be meaningfully interpreted by an instructor. Once the scenario trace has been appropriately parsed and aggregated, it can then be analyzed in a number of ways, thus enabling the instructor to extract information at a higher level of abstraction – and educational significance – than would be possible with the scenario trace alone.

The process of translating the “raw” recorded action entries into a set of parsed actions that can be easily understood by the instructor is as follows.

1. The behaviour entries are retrieved by a web service, which filters the results according to optional criteria (specified by the instructor via a web-based interface), such as a date range, behaviour type, or location range.

2. The web service performs a variety of optional statistical calculations on the retrieved behaviours, which are described in detail in the following sections.

3. The web service saves the retrieved actions to a text file, and shows them to the instructor in their “raw” format.

4. This text file, in turn, is processed further by a server-side script, and the resulting parsed behaviours are shown to the instructor.

The parsed behaviours, furthermore, are then used by the trace-based video annotation component, which is described in Section 4.6.4.

The various parsing and analysis capabilities, which are encapsulated in the Scenario Trace Analysis Toolkit, are described in the following subsections. This section concludes with a description of the trace-based video annotation component.

4.6.1 Movement Analysis

There are three different types of movement analysis supported by my analysis component: proximity to locations of interest, proximity to other participants, and paths traversed by individual participants.

To analyze a participant's proximity to locations of interest, the analysis component retrieves the relevant locations for the given scenario, and compares the participant's co-ordinates to the co-ordinates for each location. Through this mechanism, the participant's "raw" movement record (expressed in terms of virtual world co-ordinates) can be converted to a sequence of locations that the participant has visited. These locations are defined at varying levels of specificity, according to the level of granularity required in the analysis process. For more details on the definition of locations for a scenario, see Section 4.3.6.

The analysis of regions, areas, and landmarks is a relatively simple task, in that it involves comparing a static, pre-defined location with a participant's location. Another way that the participant's location information can be analyzed is by tracking a participant's location relative to other participants. Given a set of participants P, one can determine the distance between all pairs of participants P_x and P_y. This analysis is performed by analyzing the set of location records for all participant, sorted by the record timestamp, and tracking the "current" location for each participant. Each time a new record is encountered, the "current" location for that participant is updated, and the distance between that participant and all others is updated. Based on this continually-updated distance measure, the toolkit calculates the average, minimum and maximum distance between each participant, and can be easily extended to include other similar summary statistics.

Finally, by tracking the distance between subsequent co-ordinates for a participant, the toolkit is able to find the distance traveled by that participant. This measure, therefore, can be used as one measure of the level of activity of a participant, particularly if the scenario requires a significant amount of movement by the participants.

4.6.2 Object Manipulation Analysis

The scenario trace contains a record of every action initiated by each participant. This subset of the scenario trace can be analyzed in order to understand the actions each participant has taken while participating in the scenario. This analysis has several immediate applications to assessing educational outcomes for the participants.

First, the set of actions taken by each participant (or by the participants as a group) can be compared against a normative set of "correct" actions to see if a) the participant(s) executed all of the required actions, and b) if the participant(s) executed any unnecessary or counterproductive actions.

At a deeper level, the actions can be compared to a normative sequence of "correct" actions, to see if the participant(s) executed the actions in the expected order. This is especially important in cases where the order of actions is a key part of the correctness of the participant's enactment of the scenario. In a healthcare context, for example, the sequencing of actions is of critical importance when assessing a patient. This is, therefore, an essential part of evaluating the participant's competence in performing a patient assessment.

Finally a participant's (or group's) actions in a given scenario can be analyzed over multiple simulations of the scenario, to assess the extent to which the participant (or group) has improved with practice.

4.6.3 Aggregate Metrics

In addition to the above action-specific analysis of the scenario trace, several broader analyses of the trace are possible. These are described below.

Time length and Pace

Based on the first and last timestamp associated with a scenario, the STAT component recognizes the length of the entire scenario session, and also the lengths of active and inactive segments. This information can be used as a measure of the time required to complete certain key tasks, and possibly also as an indication of a participant's level of confusion, by detecting periods of inactivity. This may be especially relevant when the simulated activity is time sensitive. By examining a) the total number of behaviours that have taken place during a scenario and b) the scenario time length, an instructor can determine the number of behaviours per time unit during a scenario session, and thus determine the overall pace of that session, as well as identify unusually busy (or slow) segments of activity.

Conversation and Social Interactions

The messages sent between participants can be analyzed to understand the social interaction that occurred during the course of a scenario.

At the simplest level, the number of statements made by each participant can be counted, offering a rough estimate of the degree to which each participant participated in a conversation. In the context of a negotiation or exchange of information, an imbalance in these totals can indicate control or dominance over the conversation by a particular participant. This information can then be correlated or validated against other scenario data to confirm this preliminary indication.

Analyzing the available information further, the content of the conversation can be analyzed to determine the tone of individual participants, or of the conversation as a whole. For instance, the presence of slang terms could indicate a lack of seriousness on the part of one or more participants.

Finally, by analyzing the set of participants in a given conversation (or over multiple conversations), one can draw conclusions about the ways in which the participants tend to form conversational groups. That is, one can identify sets of participants that tend to talk to each other, or other participants that are excluded from conversations. Taking this a step further, one can use these analyses to draw conclusions about the overall nature of the interaction between participants. These conclusions could, for instance, help categorize the interaction among participants and instructors according to the taxonomy proposed by Messinger et al. [59].

Attention

The assessment of a participant's level of attention requires a more subjective kind of analysis by the instructor, and has not yet implementated in the STAT component. However, this analysis *could* be performed by synthesizing data related to a participant's gaze, movement, and conversation. A focused participant would likely move between relevant locations, focus his or her attention on relevant objects, and not engage with idle conversation with other participants. A distracted participant, on the other hand, might wander around the scenario environment, focus on any object that seemed intriguing, and have lots of extraneous conversations. This is not intended to be a rigorous description of how to analyze this aspect of a participant's behaviour. Rather, it is an indication that there is the *potential* for this kind of analysis to be performed using the scenario trace data, and that this analysis can be used by the instructor to assess a participant's level of attention to the scenario.

Patterns and Sequences

Finally, my analysis component includes two different knowledge-extraction methods: pattern matching and sequence mining. Frequently occurring sequences can be used (a) on a per-participant level, to identify a participant's pattern of behaviour across multiple attempts, or (b) on a scenario level, to identify common behaviour patterns among participants in a class. These patterns may indicate, for example, an area that needs more attention from instructors.

The sequence mining functionality is implemented as follows: Each behaviour in the scenario trace is assigned an ID (non-negative integer) corresponding to the ABM-based type of the behaviour and the associated parameters. Through this mapping, the scenario trace is compressed into a sequence of integers, which can be scanned to recognize repeatedly occurring instances of subsequences, where these subsequences represent behaviour sequences. For example, a "wandering" behaviour pattern could be defined as a pattern going back and forth between two locations of interest. That is, the behaviour that corresponds to moving from A to B might be given an ID of 1, and moving from B to A might be given an ID of 2. In this situation, a behaviour sequence of 1,2,1,2,1,2 would indicate that the participant was moving back and forth repeatedly between A and B.

This integer sequence is further processed by the Apriori pattern recognition algorithm [1], which enables the automatic identification of frequently occurring sets of integers across multiple integer sequences, which in this case refer to different scenario sessions. These sets of integers may then be converted back into sets of behaviours, using the behaviour-to-integer mapping, to recognize frequently occurring sequences of behaviours across different scenario sessions.

4.6.4 Trace-Based Video Annotation

The last tool within the STAT component is a trace-based video annotation tool. This tool automatically annotates a video recording of a scenario session with the parsed behaviours for that session (elicited from the scenario trace), using the Actionable Video Annotation (AVA) service [11] to create the necessary video annotations. When the instructor and/or participants view the video through the AVA-based video player, they will see an annotation describing each parsed behaviour, superimposed on the video at the appropriate moment.

This capability to automatically annotate a video with each participant's behaviours overcomes a significant drawback of standard video-based virtual world recording: its inability to capture every participant's behaviour. Video recording software records the screen and audio from an individual participant's perspective, while the dialog boxes that drive user interaction are only shown to the participant who has initiated the interaction. To put it more precisely, the video recording software can only capture the behaviours $b(V)$ initiated by the participant V who is running the recording program. Because of this limitation, the recording omits information about the behaviours $b(P)$ initiated by participants P s.t. $P \neq V$.

This automatic annotation capability provides the instructor with a solid starting point for further annotating and understanding the video recording of a scenario session. Given the broad range of capabilities offered by the AVA system, one can easily imagine an instructor customizing the annotated content via an integrated, AVA-based, web interface. Some of these annotation customization possibilities are described below.

Using the video annotation interface, the instructor can augment an automatically annotated video recording of the participant's behaviours, adding annotations in several ways. These automatically added annotations may be customized by the instructor to focus on specific behaviours or participants, as appropriate. Then, the instructor can manually add annotations to indicate where missing behaviours *should* have been, for instance, or to ask questions of the participant at key points. The set of parsed behaviours can also help here, to indicate to the instructor the location of important behaviours in the timeline. Through the AVA-based video player, then, the instructor can show each participant a video annotated with that participant's behaviours. The instructor can highlight the successful completion of tasks via the automatically added annotations, and draw the participant's attention to incorrect, incomplete or missed tasks through additional, manually-added annotations.

For each parsed behaviour, a message is sent to the AVA API requesting that a message be added to a set of annotations, at the time that the behaviour occurred relative to the start of the scenario session.

More specifically, the process of transitioning from a set of parsed, ABM-based, behaviours to video annotations occurs as follows.

1. The scenario trace is retrieved by a web service, and the retrieved "raw" behaviours are saved to a text file.

2. The behaviours in this text file are parsed by a server-side script, and an annotation is generated for each behaviour using the following mechanism.

3. For each parsed behaviour, the script calls the *addAnnotation* method of the AVA API, populating the annotation to be added with the content for the current parsed behaviour. Timing is an important element of the annotation – that is, the annotation should appear in the video precisely when the participant is executing the behaviour that is being annotated. However, the behaviour is recorded according to its timestamp (that is, the date and time that the behaviour occurred), while the annotation must be added according to its frame number (that is, the number of seconds since the start of the video). To bridge this gap, the script stores the timestamp of the first parsed behaviour, and uses this as its reference point for calculating the frame number for subsequent behaviours. For instance, if the first behaviour occurs at 12:34:00 (hh:mm:ss), and the next behaviour occurs at 12:35:30, then the frame number for this behaviour is calculated as 12:35:30 - 12:34:00 = 90 seconds, or frame 90.

4. Through the *addAnnotation* method, the AVA API constructs an XML file that stores the content, start time, and end time of each annotation.

5. Finally, the annotated video client uses an Adobe Flex-based interface[3] to show the user (i.e., participant or instructor) the annotated video, drawing on the XML file to display the required annotations at the appropriate times.

[3] See http://www.adobe.com/products/flex.html

While I did not have the opportunity to evaluate the trace-based video annotation component, through conversations with educators from a variety of discipines I received informal confirmation that this kind of functionality would, in principle, be quite beneficial to students and instructors alike.

4.7 Summary

To summarize this chapter, I will return to the four requirements for my platform; namely, the platform must:

1. support the creation of realistic, educationally effective scenarios;
2. enable participants to experience these scenarios in an immersive way, allowing meaningful learning to occur;
3. record participants' actions as they go through the scenarios; and
4. enable instructors to analyze these scenario traces, and thus determine whether the scenario's educational goals have been met.

In this chapter, I described the components of the framework, and the interaction of these components. In the process, I implicitly addressed these concerns. However, in order to more clearly explain how the features of the framework support these requirements, I will briefly look at each requirement in turn, and investagate the ways in which this framework addresses each requirement.

The first requirement, supporting the creation of scenarios, is met through the definition of objects, actions, variables, workflows, constraints, and timed events that make up a scenario. These components, when considered together, make up a comprehensive scenario definition, which is flexible enough to allow a variety of participant experiences. The definition of these scenario components is described in Section 4.3. In order to enable non-technical users to actually define a scenario using these components, I have created a collection of web-based authoring tools. These tools present the user with an intuitive, menu-oriented graphical interface for defining the various components, and are connected to the appropriate database tables in the MeRiTS framework. I need to do further development and testing of these tools to ensure that they are indeed user-friendly, and provide users with the appropriate degree of control over the definition of their scenarios.

The second requirement, enabling immersive enactment of a scenario by participants, occurs through the virtual world client component of the framework (described in Section 4.4. The virtual world client, in its most basic form, allows synchronous communication, some interaction with objects, and the exploration of realistic environments. However, through the MeRiTS framework, I have augmented the client with features such as robust object interaction, dynamic object composition and, most importantly, integration with the aforementioned scenario definition components. Robust object interaction capabilities allow participants to use in-world objects in ways that are appropriate and relevant to the scenario. Dynamic object composition ensures that objects behave in a realistic way, even when they are grouped with (e.g., placed on top of or inside) other objects in the scenario. This preserves the participant's sense of immersion in the scenario, and helps ensure that they have a focused, enjoyable experience. Integration with the scenario definition, finally, allows the detailed behaviours and relationships established in these definitions to be experienced by the participant through an engaging, immersive platform. With the integration of these additional features, the virtual world client becomes a much more fully-featured, appealing platform for educational simulation.

The third requirement, recording participants' behaviours, is addressed in Section 4.5, which describes the integration of this ABM-based recording capability with the rest of the MeRiTS framework. This section presents the mechanism by which the recognized behaviours are stored by the MeRiTS framework, and connects this to the analysis of the recorded scenario trace. The recording device itself, and the mechanism through which behaviours are recognized and relayed by the device, is discussed in further detail in Section 3.3.

The fourth requirement, finally, is met through the STAT component, described in detail in Section 4.6. The tools in this component enable the instructor to extract meaningful information from otherwise unmanageable low-level scenario traces, and to *understand* the extent to which participants are absorbing the intended lessons from their scenario experience.

Chapter 5

Empirical Evaluation

In order to validate the components of the MeRiTS framework, as well as the educational effectiveness of the system as a whole, I have conducted a number of case studies and experiments over the past several years.

In my first case study (described in Section 5.1) I worked with educators at a local technical college to develop a scenario to train paramedics and EMTs in communication and procedural skills. Through this experience, I made significant improvements to the MeRiTS framework, obtained student feedback on the virtual world client, and gained valuable insights about the process of developing a virtual world scenario.

In my second case study (described in Section 5.2), I worked with colleagues in the School of Business on a virtual world-based marketing experiment. This experiment, which explored issues of negotiation and personal characteristics, provided me with an opportunity to evaluate the action recording and analysis components. By creating a virtual setting within which my colleagues could conduct their simulated economic exchange, I was able to conduct a case study focusing on the action recording and analysis aspect of the framework, and provide validation for the effectiveness of the analysis tools.

In my third experiment (described in Section 5.3), I worked with instructors in the Faculty of Nursing at the University of Alberta to develop a scenario for ER nursing students. In this experiment, I developed clear educational assessment methods alongside the virtual world scenario, and thus were able to obtain some empirical evidence of the educational effectiveness of the scenario. Specifically, by comparing the performance of students who used the virtual world scenario before and after participating in the scenario, I was able to measure the learning that occurred through their experience participating in the scenario.

Scenes from the three experiments are shown on the following pages, to provide a visual sense of the range of contexts that were explored.

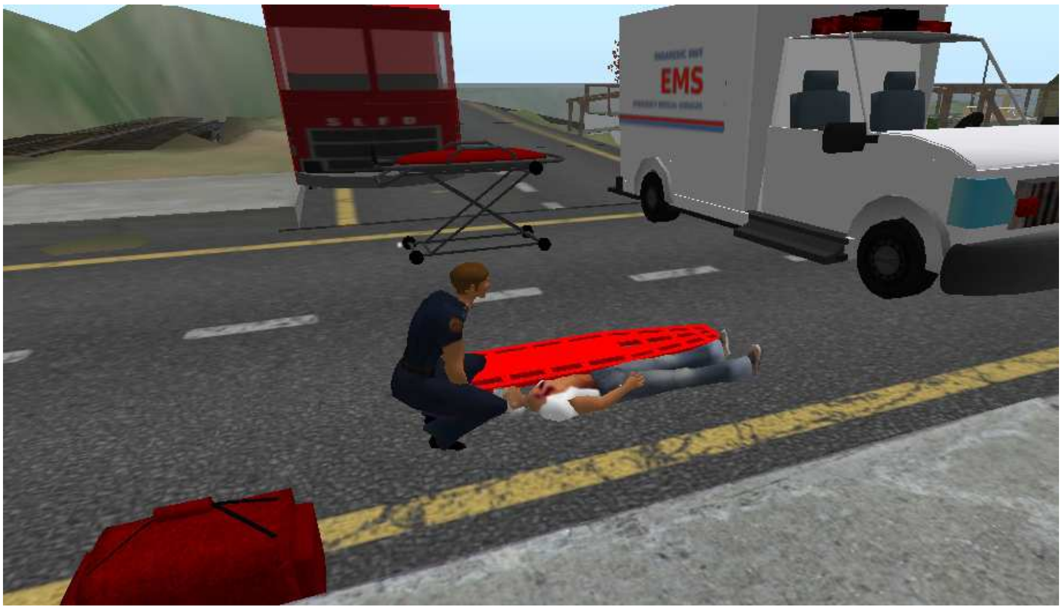


Figure 5.1: EMT Experiment – Accident Scene

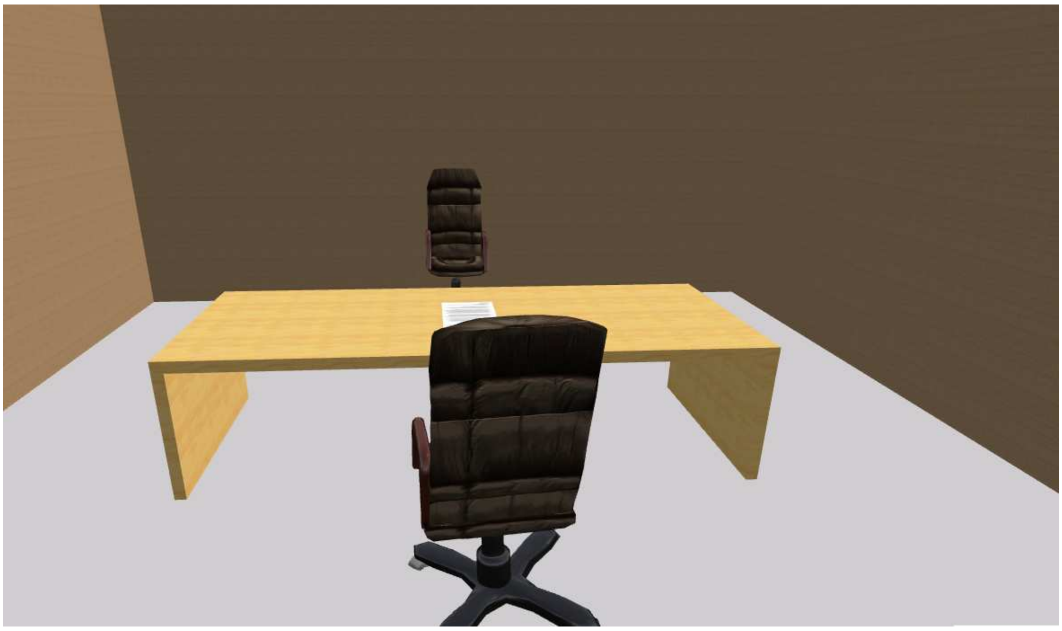


Figure 5.2: Marketing Experiment – Cubicle

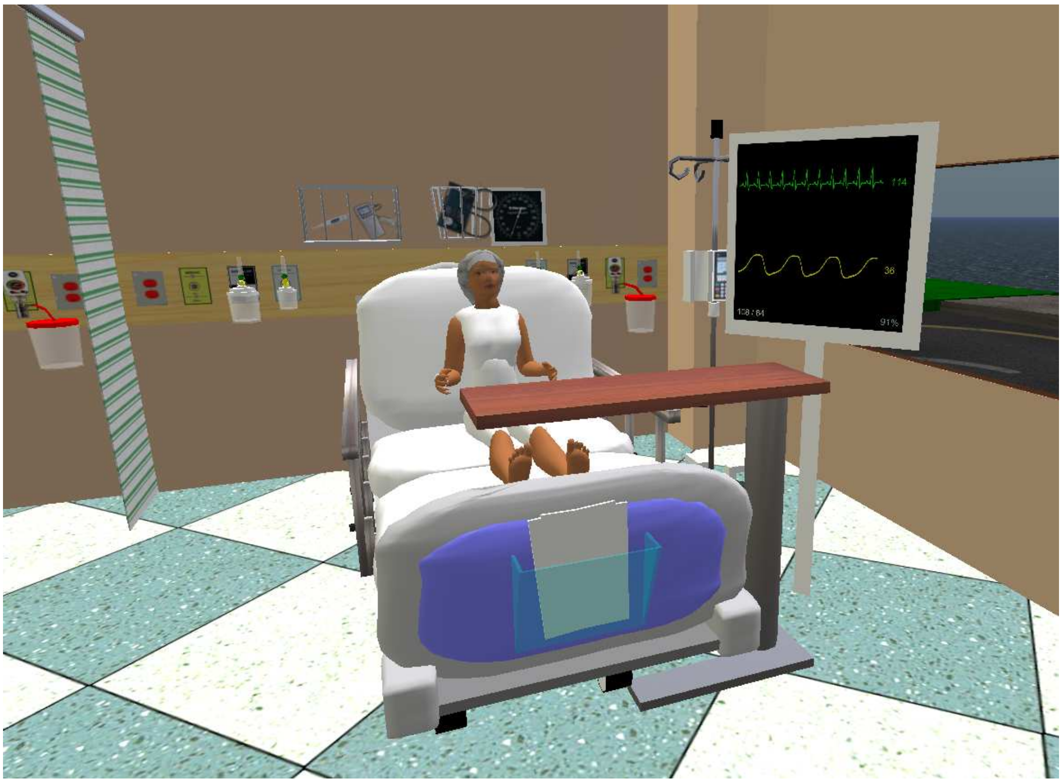


Figure 5.3: ER Experiment – ER Ward

Based on my experience with these case studies and experiments, I have developed a set of guidelines to avoid potential hurdles and pitfalls.

5.1 Case Study: EMT Patient Handoff Training

Our first full-fledged MeRiTS-based case study was a VW-based training scenario for paramedic and Emergency Medical Technician (EMT) students at the Northern Alberta Institute of Technology (NAIT).

EMTs need to acquire and apply a diverse set of knowledge and skills to provide the best patient care. Furthermore, effective communication within, and across, disciplines is critical in order to coordinate activities with co-workers and hospital staff. Typically, these skills are taught through real-life scenarios, which are expensive and impose significant restrictions on the participants.

The scenario that I created focuses on providing an opportunity to practice the victim rescue process, and to gain experience in interdisciplinary communication. Specifically, EMT students (typically operating in a team of two, with one student taking a “lead” role, and the other acting as an assistant) must first rescue a victim from a car accident and bring him to the ambulance, maintaining appropriate communication with other members of the EMT team. Next, the students must bring the victim to the hospital and, while in transition, conduct an initial conversation with a radio dispatcher and, potentially, members of the emergency room (ER) staff at the hospital (if the dispatcher decides to patch the EMT through to the hospital). Finally, the students must bring the victim into a hospital ER department, where they have a hand-off conversation with members of the ER staff. This conversation elicits any additional details about the victim rescue and treatment process, and further tests the EMT students' interdisciplinary communication skills. It should be noted that the roles that are adopted by the scenario participants – lead and assistant EMT, nurse, and radio dispatcher – are not specified or enforced by the MeRiTS system, and thus any participant can take any action, regardless of his or her role in the scenario. However, the assignment of roles to participants can be assigned and enforced informally by instructors and facilitators, and used by instructors as part of the student assessment process.

In creating this scenario, I am seeking to address several educational shortcomings of real-world simulations. By providing students with an opportunity to practice their skills in a relevant context (that is, the virtual representation of the accident scene and hospital environment), I hope to improve the applicability of their medical skills training. In delivering the scenario through a shared online space, I am removing several significant logistical constraints – requiring all participants to be co-located, for instance – and enabling participation by distance learning students. Finally, by integrating interdisciplinary communication into the design of the scenario, I am ensuring that students get relevant, meaningful practice and training in this crucial skill.

5.1.1 Settings and Models

In designing and implementing the scenario, I began by defining the required settings, objects, workflows, and constraints. The creation of these basic scenario building blocks (defined in Section 4.3), provided us with an initial implementation, which I then used as a vehicle for discussion, requirements validation and refinement with the NAIT instructors. Each of these elements is described in the following paragraphs and, where appropriate, shown in Appendix A. Within the appendix, Section A.1 shows screenshots of the settings for this scenario, Section A.3 shows the interactive objects that are used in the scenario, and Section A.4 describes the workflows and constraints for those objects. Note that, while the incorporation of timed events was a possibility that was discussed with the NAIT instructors – specifically, including an injury to the patient that caused blood to spurt from the patient's chest, starting mid-way through the scenario – in the end these events were not included in the scenario.

To maintain the necessary level of immersion and realism, I needed to develop appropriate settings in which the victim treatment and transfer could take place. The two key settings, therefore, were the accident scene and the hospital. In the first setting, the main challenge was to provide each participant with the same level of background information that would be available in a real-world rescue situation. Through the virtual world, the participant needed to be presented with any sights, sounds and smells that could contribute to his or her assessment of the accident scene. Some of these, such as the sound of a barking dog or the tree that the accident victim struck, could be conveyed through sound clips or in-world objects, respectively. Others, such as the smell of gasoline, needed to be conveyed via a written description. The accident scene is shown in Figure A.1.

In the second setting (the hospital), the focus was on providing a space where students, instructors, and facilitators could interact, conduct a hand-off conversation, and have a post-scenario debriefing session. To achieve this goal, I created an ER space that was large enough for many participants to occupy simultaneously, and also created a separate debriefing area for use after the scenario has been completed. The ER is shown in Figure A.2, and the debriefing area in Figure A.3.

Once the settings were developed, a set of locations for the scenario were defined, based on the key landmarks in the scenario. These locations are shown in Figure 5.4, below. These landmarks were particularly important in this scenario, since the process of transferring the victim from the accident scene to the hospital required a great deal of movement by the participants – from the accident scene to the ambulance, for instance, and then to the hospital – and thus it was essential to have a clear picture of where, exactly, the participants were located as they worked through the scenario.

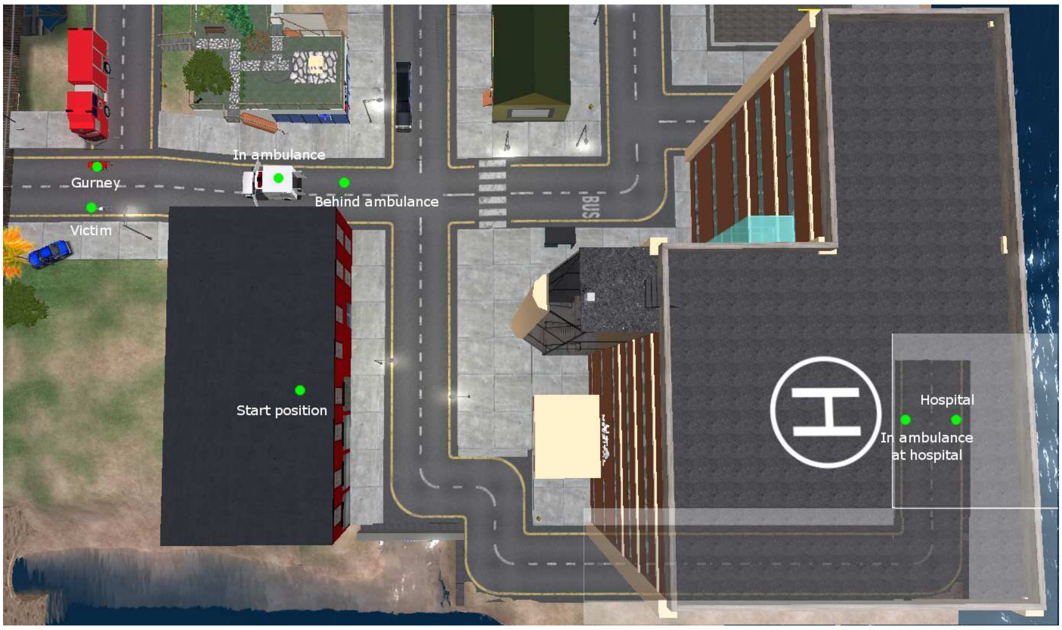


Figure 5.4: Map of scenario locations

To enable participants to experience the scenario in an immersive, realistic way, several virtual objects were required. Most of these objects – the ambulance, gurney, radio, and spineboard – were defined with a relatively limited set of available actions. These actions include choosing a destination when driving the ambulance, or moving the spineboard from the stretcher to the road. Moreover, the visibility of these actions is determined by the position or status of the object – that is, the “push” action for the gurney is not shown while it is in the back of the ambulance. Although the applicability of these actions could have been determined via feasibility constraints, this would result in showing the participant potentially irrelevant actions. In order to keep the interface as straightforward as possible for the participant, I decided instead to include additional logic in each object’s script to show or hide the potentially irrelevant actions, as appropriate. The objects are shown in Section A.3, and the actions for the non-victim objects are listed in Table A.3.

The victim object, meanwhile, is defined with a much broader range of actions. Each part of the victim (e.g., head, arms, legs) may be investigated and treated separately, or the participant may choose to treat the victim as a whole – to administer medication, for example. The victim is implemented as a hierarchially-organized collection of objects, as described in Section 4.3.1. Moreover, depending on the status of the victim and the additional medical equipment placed on the victim by the participant, additional actions – such as removing an oxygen mask – may become available. As with the other objects in the scenario, I chose to control the availability of these additional actions via the Second Life script for the patient object, rather than creating feasibility constraints. Finally, depending on the type of treatment chosen, the victim may respond with a written description of the result, a sound clip (of the victim's breathing, for example), or a change in the victim's appearance. These actions, and the associated responses, are implemented using the interactive object algorithm described in Section 4.4.3. Both the types of treatment available and the victim's response to each treatment were created based on specifications and feedback provided by the NAIT instructors.

In order for the scenario to function properly, I needed to define a set of variables to maintain the required information about the state of a particular scenario session. These variables fall into two conceptual categories: some of the variables are responsible for maintaining the status of the victim – his blood pressure, pulse, and blood glucose level. The other variables track the participant's progress through the scenario, in terms of the status of key activities. For instance, before the victim can be moved on to the spineboard, the spineboard must be moved from the gurney to the road. To address this constraint, a variable was defined to keep track of the spineboard's current position. The complete set of variables for the scenario is listed in Section A.2.

The scenario workflows define the behaviour of the various components of the scenario (blood pressure cuff, glucometer, gurney, radio, spineboard, stethoscope, thermometer, and victim) and thus, taken together, define the actions that are available in the scenario as a whole. The victim workflow, which is the most complex, is shown in Figure A.13. This workflow, following the structure described at the end of Section 4.3, processes the current action, sets relevant variables, and returns the required information to the participant. For instance, the current action might be to check the victim's pulse, in which case the workflow specifies that this action should be stored, and a message returned to the participant conveying the victim's pulse.

For the victim workflow, this processing is divided up into a few categories: checking the victim's vital signs, administering medication, and moving the victim. Within each category, a certain amount of control flow is required to determine exactly what the participant has chosen to do (e.g., *where* the victim is being moved to), and then to handle the action accordingly. The workflows for the other objects in the scenario are shown in Figures A.8 to A.12.

The constraints, meanwhile, were largely focused on avoiding unrealistic or implausible situations. For instance, once the victim has been given medication, the constraints ensure that he cannot be given that same medication again. Similarly, if the victim has been moved on to the stretcher, he cannot be placed on the stretcher again. If the participant attempts to execute an impossible action, he or she is shown an appropriate error message, through the mechanism described in Section 4.3.4. Because of this focus, most of the constraints were local (that is, specific to a single object), and fell under the category of feasibility, rather than advisibility. The complete list of constraints is shown in Table A.4.

5.1.2 Development Process

The development of the paramedic training scenario followed an iterative process, with several opportunities for demonstration and validation among the participants. Each stage of this process is described in the following paragraphs.

Initial development phase: The first phase of the process was the initial development of the scenario. Based on some preliminary conversations with paramedic instructors and other medical professionals, I elicited a set of requirements for the scenario, and a list of relevant entities and actions to be modelled. Based on this information, I created an initial version of the scenario – workflows, virtual objects, and the like – which could then act as a starting point for further discussion and development.

Pilot Session: In March of 2010, I held a pilot session with a small group of paramedic students at NAIT, along with a paramedic instructor and two medical doctors from the University of Alberta. From this session, I gathered some feedback from students, via a debriefing session and post-session questionnaires. Through observing the students' actions as they went through the scenario, I also identified several areas for improvement, both in the rescue scenario and in the MeRiTS system as a whole. In particular, some of the students found the user interface difficult to use, which then meant that they were unable to treat the victim effectively. As well, although the various workflows and components were designed to operate well together, there were some combinations of actions that the students tried which were not anticipated, and which resulted in unrealistic behaviour. Another problem that occurred was that the video recording expectations were not made sufficiently clear to the facilitators before the session, and thus there were some portions of the students' scenario experience that were not recorded. Specifically, while the interaction between the students and doctors in the ER was recorded, the students' actions in rescuing the victim at the accident scene were not fully captured.

Instructor Feedback: In October of 2010, I had a meeting with two instructors from NAIT who had not previously been involved with the project. In this meeting, and several subsequent meetings in November and December, I received specific feedback about the clinical validity of the scenario, and on areas for improvement in the user interface. Among the concerns raised during these meetings were:

- Insufficient range of treatment options: The instructors felt that the range of victim treatment options available to the student was not broad enough to allow the students to properly treat the victim.

- Level of realism: The instructors identified a few actions - such as moving the victim to the ambulance while he is on the stretcher - which were not shown in a realistic way. Similarly, the accident scene seemed empty, and lacked some of the background elements (e.g., bystanders, ambient noise) that would be present in a real-world accident scene.

- Quality of communication: Depending on the method of communication used - in person conversation or two-way radio exchange - the scenario used either voice over IP (VOIP) communication (that is, conversation with headsets and microphones), or written messages. The instructors felt that switching between these modes of conversation would be jarring for students, and that communicating via written messages would be inconsistent with what they would experience in a real-life situation. However, there was also concern that the VOIP communication needed to of sufficient quality and clarity to allow straightforward, smooth conversation among the participants.

Scenario Refinement and System Improvement: Between October, 2010 and January, 2011, I worked on implementing changes to the system, based on feedback from the paramedic instructors. Some of these changes were limited to the particulars of the scenario being developed, while others were improvements to the system as a whole. The scenario-specific changes that were implemented during this period were:

- Victim treatment: The options for treating the victim were expanded and clarified, as per guidelines from the instructors. Additional body parts (e.g., neck, abdomen) were also added, along with appropriate treatment options and responses.

- Realism: The paramedics were given appropriate uniforms (as can be seen in Figure 4), as were the nurses and doctors in the ER. This helped users feel "in character". I also added bystanders and several background elements – a barking dog, for instance – to the accident scene, to further increase the feeling of realism.

- Communication: I decided to use Second Life's built-in audio chat (that is, VOIP-based) component for conversations, rather than relying on written messages.

In addition to these scenario-specific changes, I made two key system-wide improvements:

- Synchronized movement: The mechanism for portraying synchronized movement by multiple objects was made much smoother, and thus more realistic. The details of my approach to movement synchronization are presented in Section 4.4.3.

- Usability: Rather than requiring the user to right-click to interact with some objects, and left-click for others, the interaction method was unified so that left-clicking works for all objects, and in all situations.

Training Session: Based on my experience in the pilot, I realized that providing students with training in Second Life before asking them to participate in the scenario was essential, to ensure that the students have an adequate level of familiarity with the user interface and capabilities of the system. To this end, I created a 40-minute training and demo program for students, and delivered it to fifteen paramedic students at NAIT at the end of January, 2011. In this session, I showed students how to move and communicate with their avatars, and gave them a chance to practice with simplified versions of the equipment they would be using in the rescue scenario. I also showed them a demo of the rescue scenario, to give them a better idea of what the experience of participating in the scenario would be like.

5.1.3 Study Procedure and Results

Procedure

To test this scenario, two EMT students went through a scenario session, with a Paramedic instructor supervising the rescue process, and an Emergency Medicine instructor playing the role of an ER nurse and subsequently evaluating the hand-off conversation. The two students went through the session as a team, with Participant 2 taking a lead role in the scenario, and Participant 1 adopting a supporting role. Based on these roles, it was expected that the student in the lead role would execute more actions than the student in the supporting role, since the lead role involved a greater responsibility for initiating assessment and treatment actions. However, it was expected that both students would visit the same locations, since they would both be responsible for treating the victim throughout the scenario. I was interested in evaluating the quality of the students' learning experience, and in assessing this perceived level of quality from the perspectives of both students and instructors.

Results

Through the recorded scenario traces, information was elicited about the quality and characteristics of the students' experience in the scenario. This information is summarized in 5.1.3. In this table, one can see that Participant 2 was by far the more active of the two, having executed nearly three times as many actions as Participant 1. Participant 2 also traveled much further, which indicates a greater level of activity. Since Participant 2 was asked to take on a lead role in the team, these numbers make sense, and confirm the expected level of activity of each student.

Table 5.1: Student Activity Summary

Participant	**# actions**	**Time (mins)**	**Pace (actions/min)**	**Most freq. action**	**Distance travelled (m)**
Participant 1	35	60	0.58	Palpate chest	446.67
Participant 2	111	60	1.85	Call hospital	694.89

In order to provide a more detailed analysis of the students' behaviour, their visits to landmark locations were tracked. These visit sequences are shown, alongside the "correct" sequence, in Table 5.1.3. From this table, by comparing the students' visit sequences to the "correct" sequence of locations (as dictated to us by domain experts) it is clear that the performance of Participant 2 was much better that that of Participant 1. The locations that the participants visited are also shown in Figure 5.5.

Table 5.2: Locations Visited During EMT Scenario

Participant 1	**Participant 2**	**"Correct" Sequence**
Start position (fire hall) In ambulance on site Behind ambulance on site Victim Behind ambulance at ER	Start position (fire hall) In ambulance on site Behind ambulance on site Gurney Victim Gurney In ambulance on site Behind ambulance on site Hospital Behind ambulance at ER Hospital Behind ambulance at ER	Start position (fire hall) In ambulance on site Behind ambulance on site Gurney Victim Gurney Behind ambulance on site In ambulance on site Behind ambulance on site Hospital Behind ambulance at ER

The hand-off from the EMT students to the instructor playing the role of an ER nurse was followed by a debriefing conversation among the students and instructors. The primary focus of this conversation was an evaluation of the students' performance by the instructors, although the simulation experience as a whole was also discussed. In the course of this discussion, the instructors offered students feedback about improving their communication skills, and there was also discussion by both students and instructors about the pros and cons of the simulation environment.

One such piece of feedback was that the students relay not only the assessment and treatment information, but also "pertinent negatives" – that is, potential causes for concern

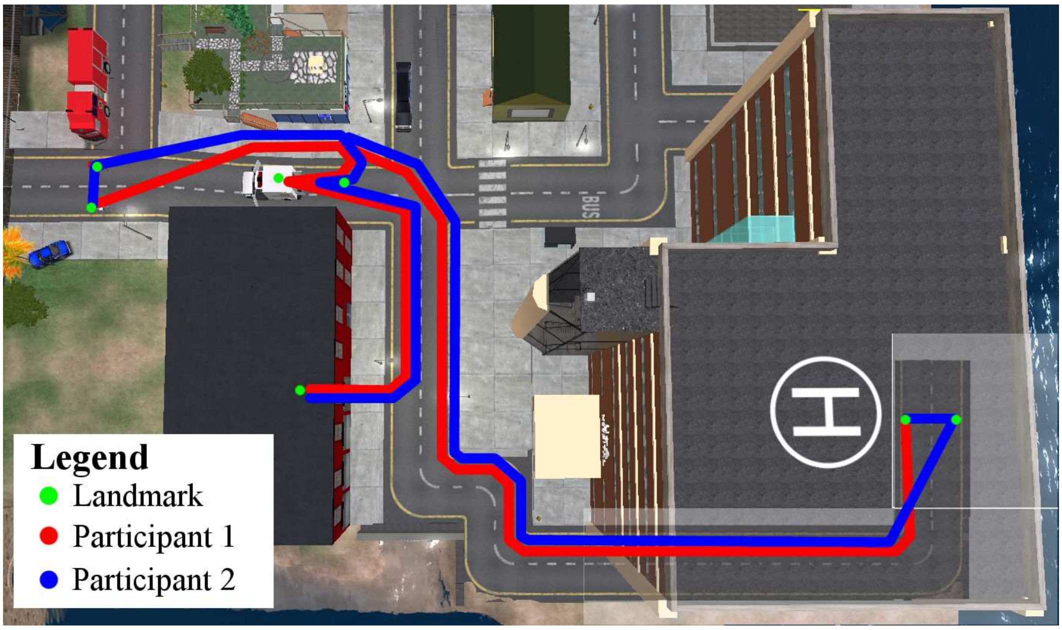


Figure 5.5: Map of Visited Locations

that were *not* present, or information (from bystanders, for example), that was not available during the patient rescue process. This instructor, who was playing the role of an ER nurse, also confirmed that the students' conversation with him at the hospital, which was much shorter than the conversation that had happened while the students were en route to the hospital, was indeed sufficient to convey the necessary information. The other instructor reminded the students of a mnemonic – "CHAT", for complaint, history, assessment and treatment – that they could use when relaying information to other participants in the rescue and hand-off process. This instructor also highlighted the importance of continuous communication between the students as they went through the rescue process, as this was an area where the students did not do very well.

The students, meanwhile, described a certain amount of confusion about what their partner was doing during the rescue, and appreciated the feedback from the instructors on their communication skills, the CHAT mnemonic, and the relevance of pertinent negatives. Participant 1, in particular, commented that

> *The pertinent negatives is actually something that I wouldn't have thought of, and that was great to hear.*

They also identified a few areas for improvement, such as the inclusion of bystanders or other participants, which would make the rescue experience richer and more complex. Participant 2 also suggested incorporating additional avatar gestures and animations, in order to visually convey the activities of each avatar. While this would certainly be a useful feature (and is discussed in Section 6.2), it was pointed out by one of the instructors that not being able to see a colleague's actions was in some ways a *more* realistic approach, since

Finally, the same student also commented that

and thus appreciated the opportunity to practice that experience in a virtual environment.

> *There's a lot of times [in a real-world situation] that things are happening, and it's more because of things that you've said or that you've delegated, than you're going to see those things happening.*

> *I've never actually given a patient off to anyone, and we didn't even practice that in scenarios at school that much.*

5.1.4 Discussion

The EMT handoff scenario provided good experience in developing a virtual world scenario, with regards to going through a collaborative, interdisciplinary development process, refining the scenario content and MeRiTS tools, and ensuring the resulting scenario was ready for use by students. The experiment, finally, provided useful initial validation of the educational utility of the scenario; students and instructors alike found the experience worthwhile, and students were able to communicate and collaborate effectively, and receive educationally relevant feedback from instructors. Specifically, based on what the students said – and what they didn't say – the instructors were able to offer the students advice on how to improve their communication, both within their paramedic team and when communicating with other disciplines involved in the rescue process.

Furthermore, through analyzing the students' scenario traces, the students' actions were evaluated on several levels. First, the instructors assessed the students' actions with respect to the roles that were assigned to the students (lead and supporting paramedic in the rescue process), and see if the students' in-world actions were consistent with these roles. The instructors were also able to compare the locations visited by each student to the "expected" sequence of location visits for the scenario, and thus assess the "correctness" of the students' actions, in terms of their movement through the scenario. In a more rigidly-defined scenario, the students' interactions with in-world objects – and perhaps even their communication – could be compared to a pre-defined "correct" set of locations, actions, and/or communication activities, to obtain a more detailed evaluation of each student's actions.

5.2 Case Study: Marketing Experiment

Marketing researchers ran an experiment with students in Second Life. While the marketing experiment was not explicitly designed for the MeRiTS framework, I made use of the opportunity to test the utility of the framework's analysis tools. This case study did not focus on the educational impact of the framework, but rather on its potential to provide a virtual space to conduct a study, its capacity to accurately record participants' actions in a virtual world, and subsequently enable instructors to analyze these actions in a variety of ways. The marketing experiment itself is described in the following paragraphs.

In the experiment, students took on the role of either a "buyer" or "seller" in a series of negotiations. In these negotiations, the seller offers to rent a stall at a market, and the buyer wants to rent the stall for their business. The buyer can expect to make \$5, and the seller, who is aware of this, can rent the stall for anywhere from \$0.01 to \$5.00. In this situation, then, for the buyer to "break even" they will want to pay at most \$2.50 for the market stall. To ensure that the participants were motivated to get a good deal, they were paid at the end of the experiment according to how much they "earned" in the experiment – that is, if a student "broke even" in each of the four rounds, he or she would have been paid a total of

$10 at the end of the experiment.

The objective of the experiment was to study how appearance influences the negotiation outcome: the buyer and the seller were assigned avatars which had the potential to bias the negotiations. The hypothesis is that an attractive (or tall) participant will be able to get a good deal from an unattractive (or short) participant. That is, if the seller is attractive (or tall), he or she will be able to get a higher price, while if the buyer is attractive (or tall), he or she will be able to negotiate a lower price. Each negotiator participated in four rounds of negotiations, where they were assigned a variety of avatars, with respect to height and attractiveness.

5.2.1 Settings and Models

The negotation space, objects, and workflows are described in Appendix B.

First, I needed to create an environment in which the negotiation could take place. This environment needed to be appropriate and realistic enough to be plausible, but not so detailed that it would distract participants from the negotiation process. I decided on creating a series of cubicles, each containing two chairs and a desk. Another important consideration was the way the cubicles were laid out in the virtual world. To avoid having the participants overhear each other in real life, Second Life's text-based chat feature was used. However, this required considering the possibility of participants overhearing other text chat conversations in the virtual world. In Second Life, a message sent by text chat will be visible to any other avatars within 20 metres. To account for this, the cubicles needed to be separated by 20 metres to avoid having the content of multiple negotiation sessions visible within a single cubicle. A screenshot of the negotiation environment is shown in Figure B.1.

In this scenario, the variables that were required were focused on the proposal and confirmation of a negotiated price. These variables included a means of identifying the buyer and seller, a proposed price, and whether the price was accepted. For a complete list of variables, see Section B.2.

The only interactive object that I created for the experiment was a contract, which the participants could click on to confirm their negotiated price. The contract object is shown in Figure B.2. This object was not strictly necessary, since the negotiation result could (usually) be inferred from the log of the text chat messages sent during the negotiation. However, the price could be extracted automatically from the log of the participants' interaction with the contract object, which greatly simplified further analysis of the negotiation sessions. Furthermore, in cases where the outcome of the negotiation was not clear from the text chat log (when the last messages from each participant indicated differing prices, for example), the contract could act as a clear indicator of an agreed-upon final price, if the participants did in fact arrive at an agreement.

In developing the contract object, I wanted to ensure that the price registered via the contract was in fact confirmed by both parties, and thus avoid the potential for abuse. To this end, I created a simple workflow to be followed by the participants, and enforced by the contract. This workflow, involving a buyer, a seller, and the contract, is described below.

1. When clicked initially, the contract asks the seller to enter the agreed-upon price or "no deal".

2. The participant playing the role of the seller (P_S) enters a price (or "no deal").

3. The contract registers the price (or “no deal”), and identifies P_S as the seller.
4. The contract then asks the buyer to confirm the price.
5. The participant playing the role of the buyer (P_B) enters “confirm” if the price is correct.
6. The contract, upon receiving “confirm” from P_B (where $P_B \neq P_S$), registers the negotiation result.

If the price entered in Step 2 is incorrect, the buyer may discuss this with the seller, and ensure that the correct price is entered. The onus, in general, is on the participants to use the object the way in which it was intended; however, the responsibility is ultimately placed on the buyer to confirm the price entered by the seller.

5.2.2 Development Process

Since the interaction in this experiment was limited to a conversation between students, there was little need for workflow modelling and definition. Rather, much of the marketing researchers’ development efforts were focused on the creation (via the avatar appearance customization tools provided by Second Life) of appropriate avatars for the participants, which were then validated using reliable “attractiveness” metrics. My efforts, meanwhile, were focused on the development of a suitable space for the participants to conduct their negotiation session.

In creating the avatars, the marketing researchers and I were guided by the structure of the experiment. That is, we needed to have avatars for each of the attractive/unattractive, tall/short, and male/female combinations. Moreover, since we were expecting between 20 and 30 participants per class, we needed to have several of each avatar ready for use.

5.2.3 Study Procedure and Results

Procedure

The participants in the marketing experiment were recruited from a second-year marketing class in the School of Business at the University of Alberta. The students’ participation in the marketing experiment occurred in two phases. First, students explored Second Life to gain some original experience with the world, creating two avatars for themselves and exploring the virtual presence of several well-known businesses. Once they were sufficiently familiar with Second Life, they took part in the experiment itself.

The experiment was conducted with approximately 30 students from two sections of an undergraduate marketing course over two hour-long sessions. The students all used a single computer lab for the experiment, but communicated solely via the in-world text chat feature, and thus they could not “overhear” each others’ conversations. Each round of negotiations lasted about ten minutes, and the students were asked to log in using a different avatar for each round.

From the point of view of my research, I was interested in assessing the recording capabilities of the MeRiTS framework – with a particular focus on the chat content – and, more importantly, the analysis capabilities of the STAT component (described in Section 4.6). That is, I wanted to find out whether the content that was recorded via the framework possessed sufficient detailed for meaningful analysis, and assessing whether the STAT component was indeed able to provide the analysis capabilities required by the marketing researchers.

Results

At the most basic level, the marketing researchers wanted to know what price (if any) the buyer and seller settled on in order to validate their hypothesis about the impact of attractiveness/height on the negotiation outcome. However, in order to gain a richer understanding of the negotiation process, they used (with my guidance) my trace analysis tool to glean some qualitative information about the negotiation process. Was one participant more talkative, or generally more active, than the other? Was the outcome affected by a difference in the relative amount of activity of the participants? Was there a difference in outcome with regards to the conversational informality? That is, was the participant who used more colloquial terms more or less likely to achieve a favourable outcome?

I computed several summary statistics for each conversation: the number of statements made by each participant; the percentage of statements made by each participant that included informal or slang terms; the minimum, maximum and average distance between the participants; the distance traveled by each participant; and the price that the participants agreed upon, showing the profit made by the buyer or seller, as appropriate. These statistics, for a single negotiation from each session, are shown in Table 5.3 below. In the columns that list the participants statistics separately (Quantity, Informality and Distance Travelled), the participant with the "seller" role is always listed first.

Table 5.3: Summary of Analysis for Marketing Experiment

Session		**Price**	**Conversation**				**Distance (m)**		
			Quantity		Informality (%)		Travelled		Inter-participant
Class 1	Session 1 (R1)	$2.50	48	62	27.10	12.90	3.74	22.70	3.45
	Session 2 (R6)	$2.50	14	14	7.14	21.40	2.61	2.05	1.31
	Session 3 (R2)	$3.75	16	10	0.00	0.00	3.64	11.70	4.10
	Session 4 (R1)	$2.50	8	5	12.50	0.00	0.98	6.13	2.46
Class 2	Session 1 (R6)	$2.50	7	6	0.00	0.00	4.89	7.93	6.55
	Session 2 (R6)	$2.00	16	15	0.00	0.00	0.00	0.29	N/A
	Session 3 (R2)	$2.50	8	11	12.50	0.00	5.51	1.53	18.55
	Session 4 (R2)	$2.50	19	19	0.00	0.00	1.07	3.95	20.58

After compiling summary statistics for each session, across both classes, I computed the correlation (R-value) between the four principal conversational summary statistics – conversational quantity (the number of statements made by each participant), informality (the percentage of each participant's conversation that consisted of slang words "lol", "haha", "rotfl" and "jk"), inter-participant distance (the distance between the two participants at any point in time), and distance traveled by each participant – and the negotiated price. For the statistics where the two participants' values were calculated separately (i.e., number of statements in a conversation and distance traveled by each participant), a summary value was calculated by subtracting the seller's value from that of the buyer's. This approach was chosen in order to ensure that the summary value would be higher for differences that favoured the landlord, and lower for differences that favoured the renter. This decision was made to ensure consistency when analyzing multiple sessions, and to facilitate analysis of the correlation between these values and the negotiated price. The results are shown in Table 5.4.

Table 5.4: Correlation of Summary Statistics with Negotiated Price

Correlation statistic	Correlation (R value)
Conversational quantity	0.19
Conversational informality	-0.04
Inter-participant distance	0.07
Distance travelled	0.04

I found that there was no correlation (that is, the R-value was too close to zero) between the price and three of the statistics: conversational informality, inter-participant distance, and distance travelled. However, for the conversational quantity, there was a slight positive correlation. That is, the participant with more statements in the course of the negotiation tended to get a slightly better price – a higher price for the seller, or a lower price for the buyer.

The feedback I received from the researcher conducting the experiment was very positive, on the whole [75]. With regards to the functionality of the system, she reported that

> *The level of possible specification on the output was useful and appropriately granular... The summary statistics were a wonderful addition. I can't enthuse over them enough!*

She also had positive feedback about the usability of the system:

> *I found the tool very straightforward... This tool generally does what it is designed to do, with a minimum of superfluous interface elements.*

She had a few suggestions as well, such as adding the ability to export directly to Microsoft Excel and improving the clarity of the output shown in the browser, all of which were subsequently implemented.

5.2.4 Discussion

The marketing experiment provided initial validation of the action recording and analysis components. Specifically, the marketing researchers and I were able to monitor the participants' movement and communication through the recording device, and then draw conclusions about the results of the experiment through careful analysis of these recorded actions. These conclusions included not only the marketing research question that was asked – the correlation between avatar appearance and bargaining outcome – but also correlations between the bargaining outcome and behavioural patterns such as conversation characteristics and movement patterns.

5.3 Experiment: Training Emergency Nurses

Based on my previous experience with developing scenarios, I created a scenario for ER nursing students, and conducted an experiment to assess the extent to which using a virtual world-based scenario enriched the classroom-based learning experience.

To this end, my development process followed two parallel tracks. In the first, in a similar manner as with the other projects described here, I went through the process of defining, developing and refining a scenario using the MeRiTS system. Meanwhile, I also developed an experimental procedure which would be used to assess the students' performance in the virtual world scenario.

5.3.1 Settings and Models

The scenario development process went through the usual stages of creating an appropriate setting for the scenario, defining workflows and constraints, developing the necessary virtual objects, and validating and refining the scenario. The scenario setting, objects, and workflows are described in Appendix C.

Since the spatial scope of the scenario was relatively limited (as compared to the EMT scenario), there was only one setting that needed to be developed: an ER ward. This setting is shown in Figure C.1. I defined a set of relevant locations for this setting, which are shown in Figure 5.6. Because of the reduced complexity of the scenario setting(s) – a single indoor location, as compared to several locations, both indoor and outdoor, for the EMT scenario – the development effort required for this aspect of the scenario was less than it had been for the EMT scenario. However, it is worth emphasizing that the required development effort was related directly to the setting's *complexity*, rather than the size or number of settings required by the scenario.

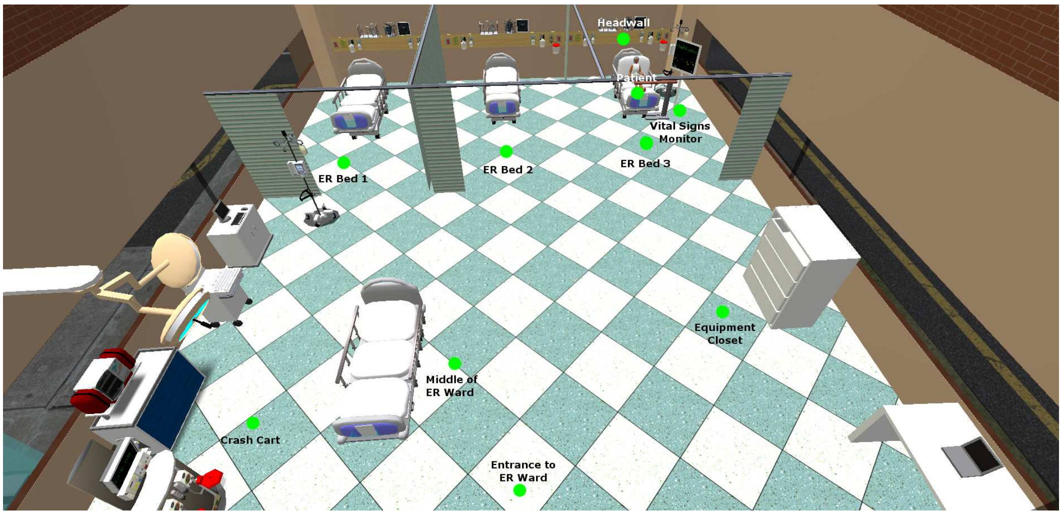


Figure 5.6: Map of ER Locations

Once the setting was defined, I then needed to define the variables that would maintain information about the state of a particular scenario session. As with the EMT scenario, I defined variables to keep track of the patient's status – her blood pressure, pulse, respiratory rate, oxygen saturation, and temperature – and to track the students' progress through the scenario, with regards to key activities. In this case, these activities included giving the patient oxygen and asthma medication. However, for this scenario I defined additional variables, which can be grouped conceptually into two categories. One set of variables kept track of the order in which the students assessed the patient, in order to provide the instructors with a means of evaluating the students' assessment skills. The other set of variables kept track of the trajectory of the patient's vital signs, as they had the potential to change over the course of a scenario session. The complete set of variables for the scenario is listed in Section C.2.

For this scenario, I needed to develop the patient, headwall equipment, and a vital signs monitor. As with the EMT scenario, the primary object was the patient itself, which in this case was a healthy, responsive, middle-aged woman. I used a similar approach to modelling this object's interactivity – the user can treat specific areas, or interact with the patient as a whole – with a few modifications for this scenario. When interacting with the patient, the user is provided with the opportunity to "talk" to the patient by choosing from a predetermined set of questions, and the patient will deliver answers (via text shown a dialog box) to these questions. The other significant change that was made for this scenario was that the mannequin used for the patient was changed to be a female, rather than male, to ensure that the scenario remained realistic and believable. The patient is shown in Figure C.2.

Another key group of objects that needed to be created was the "headwall" and associated equipment. In a standard hospital room, the headwall is a set of ports, outlets and equipment located just behind the patient's head (hence the term), and contains the equipment that nurses need to be able to access easily. This includes both frequently-used diagnostic equipment, such as a blood pressure cuff and thermometer, and equipment that would be needed in an emergency situation – an oxygen pump, an IV pump, and suction equipment. The headwall is shown in Figure C.3. In the process of creating this equipment, the mechanism for enabling the user to hold and use equipment was modified slightly. In the new approach, the user picks up an item and it is shown in the custom held items HUD, which was described in Section 4.4.3. However, to use the equipment, the user then clicks on the relevant part of the patient (e.g., the arm, to use the blood pressure cuff), and is able to choose between using the currently held object, or interacting with the selected body part, as usual. To drop the item, the user simply clicks on the image shown in the UI widget. This change to the interface required modifying the victim slightly, so that it was "aware" of which items could be used with which body parts, and also to ensure that the victim knew when the user held a particular item, so that the appropriate choices could be presented.

Finally, the scenario required a working vital signs monitor, which would draw its information from the variables stored within the MeRiTS system. To accomplish this, I created a custom web service which, using the PHP graphics library, goes through the following process:

1. The necessary vital sign data (i.e., pulse, respiratory rate) is retrieved from the MeRiTS system

2. Each vital sign value is compared to a baseline value, which is used to calculate the width of the vital sign pattern image.
   For instance, the baseline value for the pulse is 80 beats per minute (BPM), which corresponds to a pattern width of 100 pixels. If the pulse is 160 BPM, this represents a two-fold increase from the baseline value. Therefore, the width must be compressed by 50% (to 50 pixels) to generate the appropriate vital sign pattern.

3. Based on this calculation, the vital sign pattern image is repeated in order to fill the width of the vital signs monitor (800 pixels).
   Returning to the pulse example, the baseline pulse image is repeated 8 times to fill

the monitor (8*100 pixels = 800 pixels). Because the image has been compressed by 50%, this image must be repeated 16 times (16*50 pixels = 800 pixels).

4. The resulting composite pattern is stored as a PNG image.

5. The web service then returns the updated chart as a HTML page, with the generated images contained within.

The above process, therefore, takes an existing vital sign chart and updates it with a single set of new readings. However, to mimic the real-world behaviour of a vital signs monitor, which updates itself regularly (e.g., every second), this process needed to run repeatedly. To achieve this, this process is embedded in a webpage that refreshes itself with a set frequency (e.g., once per second), thus creating a constantly-updating chart. The webpage, in turn, is used by the virtual vital signs monitor as a surface texture. This means that the vital signs monitor in the virtual world is able to present the user with the vital signs chart which is updated regularly based on the values stored in the MeRiTS system. This object is shown in Figure C.4.

Therefore, in summary, I have developed several means to represent changes to scenario in the virtual world: first, and most commonly, the participant can receive feedback (either by dialog box-based messages or sound clips) that conveys relevant scenario information. Second, relevant objects can change, by moving or rotating according to the result of the action taken by the participant. Third, dynamically-updated textures can be created which important information (such as vital signs, as described above).

5.3.2 Development Process

In creating the workflows for the scenario, I started with an existing scenario – a “Shortness of Breath / Asthma Diagnosis” scenario that was created for use with a mannequin-based simulation. This was a very helpful starting point, in that the structure of the scenario, many of the clinical details (such as the patient’s vital signs), and the expected actions that would be performed by the student were clearly defined. The original version of the scenario was used for a pilot study, which is described in Section 5.3.3.

However, there were many other details that were not included in the scenario definition. For instance, in addition to the expected “correct” actions, the student might also choose to take a number of other actions which, while not impossible or entirely inappropriate, were not the focus of the scenario. For instance, the student could choose to analyze the patient’s abdomen, which would not be relevant to a breathing-related diagnosis. However, if the student is not aware that he or she is treating a patient with a breathing-related complaint, then analyzing the abdomen might be a reasonable course of action.

I worked with an ER nursing instructor to fill in these details, to refine the scenario for use within the context of her course, and to validate the clinical aspects of the scenario. Through this collaboration, I created a workflow for the ER patient that incorporated several key requirements. The patient workflow, along with the objects and ER setting, are shown in Appendix C. First, the diagnostic actions that the student can take are encoded, with the appropriate responses. Second, the workflow maintains the order in which the student has performed these actions, in order to assess the correctness of this ordering. When assessing a patient, the nurse is supposed to follow a “head-to-toe” order, to ensure that the most critical areas of the patient (e.g., head, heart, lungs) are checked first. Third, the workflow also keeps track of which actions are critical to the patient’s survival (e.g., checking the patient's level of consciousness) and which are less important (e.g., checking whether the patient has back pain). As with the "head-to-toe" ordering requirement, the student should perform critical actions before non-critical actions, to ensure that more severe problems are dealt with first. The workflow is able to, on a broader level, keep track of the feedback that should be provided to the student, based on the student's choice of actions, and the order in which those actions are performed. These feedback messages, rather than being shown immediately (as was the case in the EMT scenario), are shown to the student after he or she has finished assessing the patient.

Based on feedback from the nursing instructor, I augmented the workflow to allow the health of the patient to decline as the student goes through the scenario. While this could have been implemented as a timed event (as described in Section 4.3.5), there was concern that it would difficult to determine an appropriate amount of time to wait before causing the patient's health to decline. That is, the time required to perform an action in the virtual world is different from what it would be in real life, and the difference between the two – especially for students who are still learning how to perform these actions – is not clear. To address this concern, I created a "timing" scheme based on the number of actions taken (validated by the ER nursing instructor) to determine when the patient's health should decline. The scheme works by counting the number of actions taken, and for each n actions, the patient will decline by a given amount. This value can be adjusted based on the proficiency of the students going through the scenario – less experienced students can be given more chances to work with the patient before the decline begins, and the decline will happen more slowly. The patient's status may, of course, also change as the student goes through the scenario, and thus the patient's vital signs may stabilize or improve, depending on the student's actions.

5.3.3 Pilot Study – Procedure and Results

Procedure

In order to get some initial feedback on the scenario, I tested the original version of the scenario with students taking part in "Save Stan Saturday". This event, which took place in March, 2011, consisted of a day of interdisciplinary simulation-based training exercises for healthcare students from a range of institutions and disciplines. The students participated in groups ranging in size from one to seven students. Each group was given a ten-minute introduction to using the virtual world, and then had 15-20 minutes to go through the simulation. After they had completed the simulation, they then had a short (5-10 minute) debriefing conversation, and were then given a questionnaire to fill out.

Results

I had a total of fifteen students participate in the simulation, from the University of Alberta (10), NAIT (4), and Grant MacEwan University (1). Of these students, the largest number were nursing students (6), with others coming from pharmacy (2), medicine (2), respiratory therapy (2), diagnostic medical sonography (2), and speech pathology (1). There were 4 males and 11 females among the participants, ranging in age from 21 to 55, and an average age of 27.9. Most students were in either their first (5) or second (6) year, with 1 third-year and 3 fourth-year students.

In the questionnaire, students were asked about the level of realism in the scenario, the extent to which they were able to apply their knowledge in completing the scenario, and their overall opinion of the experience, through a series of Likert questions. For these questions, a value of 1 corresponds to “strongly disagree”, while 5 corresponds to “strongly agree”. They were also asked some free-form questions, eliciting their impressions of the scenario, their expectations of a virtual world simulation. The summarized results of the survey questions are shown in Tables 5.5 and 5.6. For complete results, see Appendix F.

Table 5.5: Questionnaire Results from Usability Testing – Likert Questions

Question	**Min**	**Max**	**Avg**
1. This experience has improved my inter-professional teamwork skills	2	5	3.40
2. The level of realism was sufficient for suspension of disbelief	1	5	3.47
3. The level of realism was sufficient to enable learning	2	5	3.80
4. I was able to apply the following knowledge and skills in completing the scenario:			
a. Knowledge of medical facts	2	5	3.79
b. Knowledge of relevant procedures	2	5	3.79
c. Communication skills	2	4	3.40
5. The experience was interesting, and I felt engaged in the experience	3	5	4.00
6. I would recommend this experience to other learners	3	5	4.07

Table 5.6: Questionnaire Results from Pilot Study – Free-form Questions

Q1. What did you like about this learning experience?
I liked interacting with the other disciplines and utilizing their expertise *I like the idea that using video game as a tool to facilitate learning. This makes learning more interesting*
Q2. What would you change or improve?
Allow for treatments and see if treatments improve or exacerbate the problem *Being able to obtain more patient history. Ask them specific questions.*
Q3. What did you learn about inter-professional collaboration and teamwork?
The more we talk the more chances of success *Teamwork is crucial in providing quality patient care* *That everyone’s opinion help in the process of diagnosing the patient*
Q5. How did this experience compare to your expectations, based on previous experience?
New and exciting *I have never experienced a health related simulation/virtual world program like this*
Q6. Do you have any other comments?
I think that computers could be used, if real people can’t meet with each other *I was initially intimidated... but was willing to try and I had fun.* *If successful, would be a great resource for students*

5.3.4 Pilot Study – Discussion

From the results shown in Table 5.5, one can see that the students felt the scenario was sufficiently realistic, and had a very positive opinion of the scenario, on the whole. From the free-form text results, one can see that students were enthusiastic about the idea of using a virtual world for education. They highlighted some areas for improvement — mostly having to do with the range of treatment options or interactivity of the patient — which were helpful in developing the customized version of the simulation. Finally, they were able to learn valuable lessons about interdisciplinary communication and teamwork, which was very much in keeping with the overall goal of the "Save Stan" event.

Based on these results, I was able to determine that the ER scenario was, with some improvements to the patient treatment options, usable and appropriate for a range of undergraduate healthcare students.

5.3.5 Collaborative Study – Procedure and Results

Following the pilot study, I planned an experiment that would assess the educational impact of the scenario for nursing students. While I was finalizing these plans, I had the opportunity – through a second offering of the "Save Stan" simulation event, conducted in February, 2012 – to work with nursing, pharmacy, and respiratory therapy (RT) students to evaluate the simulation in a collaborative setting. While the original intention had been to run the experiment individually with upper-year nursing students, I felt that the Save Stan event would provide the students with a valuable interdisciplinary learning opportunity, in addition to the original goal of providing training in patient assessment.

Collaborative Study – Procedure

The collaborative version of the experiment was run in seven rounds, where each round consisted of one "run" of the simulation with a group of 4-6 students. Within each run, my colleagues and I followed the experiment procedure – administering a pre-simulation test and questionnaire, providing a brief training session, running the scenario, conducting a debriefing session, and administering a post-simulation test and questionnaire. My colleagues enlisted a nursing instructor[1] to conduct a debriefing session with the students following each simulation session.

The participants in the experiment went through three phases, which were:

1. Pre-simulation: All participants took a short test, which asssessed their level of proficiency in going through the patient analysis and diagnosis process. The results of these tests are referred to as $PreTest$. The participants also filled out a brief questionnaire which elicited previous experience with simulation-based training and virtual worlds.

2. Simulation: The participants went through a brief training session, which gave them the required familiarity with the virtual world interface. After this, the participants went through the scenario in small groups, conducting an assessment of the patient and recording their diagnosis. These activities were recorded using the action recording tools built in to MeRiTS, and were also captured using screen-capture software

[1] Katherine Bowman, from Grant MacEwan University

(Camtasia). Following the scenario, the students went through a debriefing conversation with a nursing instructor.

3. Post-simulation: The participants were given a post-simulation questionnaire, which elicited their opinions about their simulation experience. The participants also took a second short test, similar to the pre-simulation test, to once again assess their proficiency in analyzing and diagnosing a patient. These results are referred to as $PostTest$.

From this process, several key variables can be analyzed, in order to draw conclusions about the virtual world-based learning experience:

- Absolute improvement: By comparing $PreTest$ to $PostTest$, one can determine what (if any) improvement occurred through the students' participation in the virtual world scenario. The null hypothesis is $PreTest = PostTest$.

- VW actions vs. test performance: By connecting students' actions in the scenario with their test results, one can attempt to draw two kinds of conclusions: first, one can see if there is any correlation between the students' pre-test performance and their scenario proficiency – that is, whether their prior knowledge and abilities allowed them to perform more (or less) effectively in the scenario. Second, their scenario proficiency can be correlated with their post-test performance, to see if their scenario experience had any impact on their test performance.

Due to the limited time available for each "run" – one hour, including setup time – I was not able to assess the students' attitudes towards computer-aided instruction, as I had initially intended. Additionally, because all participating students went through the VW scenario, I was not able to compare the results with a control group, as per my initial experimental design. However, there were a substantial number of participants who provided pre- and post-simulation test results, as well as informal questionnaire-based feedback. The results of the analysis of this data are described in the following section.

Collaborative Study – Results

Here are the summarized results from the version of the experiment that was run as part of the Save Stan event. For the full test contents and results, see Appendix G. An excerpt from the test that was given to the students is shown Table 5.7.

Table 5.7: Excerpt from Pre- and Post-Simulation Test

Part 1: Choose Questions to Ask
Please choose the three most important questions from the following options: 1. Do you have any allergies? ⋮ 6. Has anyone around you been sick?
Part 2: Choose Assessments
Please identify the 6 most important assessments, and suggest an order for them. a. Back pain ⋮ m. Peripheral pulses
Part 3: Choose Actions
Please identify the 7 most important actions. a. Apply oxygen ⋮ l. Sit patient up in bed

A summary of the pre- and post-simulation test results is shown in Table 5.8, with the full results shown in Tables G.2, G.3, and G.4.

Table 5.8: Pre- and Post-Simulation Test Results

		Test Components			
Participant	**Num students**	**Questions to Ask**	**Assessments**	**Actions**	**Total**
Pre-Sim Avg	27	5.44	9.85	10.08	25.15
Post-Sim Avg	24	5.57	10.21	10.42	25.96
Average Difference		0.13	0.36	0.34	0.81

In terms of individual participants' results, ten of the participants improved their results, six participants saw a decline in their results, and the remaining four stayed the same.

The pre-simulation questionnaire, shown in Table 5.9, focused on the students' previous expectations and experience with simulation-based training and virtual worlds. The full pre-simulation questionnaire results are shown in Table G.6.

Table 5.9: Pre-Simulation Questionnaire – Likert Question Results

Rating Scheme	**Min**	**Max**	**Avg**
1. Rate your previous experience with educational simulations.			
None 1 - 7 Extensive	1	7	4.36
2. Rate your previous experience with virtual worlds/online environments.			
None 1 - 7 Extensive	1	7	2.24
3. What are your expectations for the simulation?			
Low/Pessimistic 1 - 7 High/Optimistic	3	7	4.96
4. What are your expectations for the virtual world interface?			
Low/Pessimistic 1 - 7 High/Optimistic	2	6	4.44

The post-simulation questionnaire, as with the questionnaire from the usability testing session, used a five-point Likert scale, where a value of 1 corresponds to "strongly disagree", while 5 corresponds to "strongly agree". The results for the Likert questions and free-form results are summarized in Tables 5.10 and 5.11. The full results are shown in Table G.8 and Section G.4.

Table 5.10: Post-Simulation Questionnaire – Likert Question Results

Question	**Min**	**Max**	**Average**
1. This simulation has enhanced my understanding of the patient analysis and treatment process.	3	5	4.00
2. The level of realism was sufficient to enable learning in the following areas:			
a. Interacting with the patient	1	4	2.63
b. Using medical equipment	2	5	3.32
c. Treating the patient	3	5	3.79
3. I was able to apply the following knowledge and skills in completing the scenario:			
a. Knowledge of medical facts	2	5	4.00
b. Knowledge of relevant procedures	2	5	3.79
c. Communication skills	3	5	4.37
4. Throughout the scenario, I understood what I needed to do next	2	4	3.26
5. The experience was interesting, and I felt engaged in the experience	3	5	4.00
6. The scenario was well organized	3	5	3.84
7. I would recommend this experience to other students	3	5	3.89
8. Overall, this was a useful learning experience, in that it added to my:			
a. Textbook learning	2	5	3.44
b. Classroom learning	2	5	3.58
c. Practical learning	2	5	3.89

Table 5.11: Post-Simulation Questionnaire – Free-Form Question Results

Q1: What did you like about this simulation?
Never have experienced a virtual simulation before, it was different and a good experience. *It forced us to communicate because we all [had] different perspectives.* *It gave us a good opportunity to interact with other health care providers.*
Q2: What would you change or improve?
I would improve the training... before the simulation. Better patient care would result. *The questions that we can ask the patient should be more detailed.*
Q3: What did you learn about patient analysis and diagnosis?
Organization is important in assessments. *To check the vitals*
Q4: What was your experience with the virtual world interface?
It was difficult compared to reality in some aspects. *I am not very skilled with computers so I was mostly confused about how to do things.*

The other approach that I took to eliciting information from the questionnaire results was to summarize each student's pre- and post-simulation questionnaire responses into descriptive categories – low, medium, and high levels of experience/expectations for the pre-simulation questionnaire, and negative, neutral and positive opinions for the post-simulation questionnaire – based on the average of the student's scores for all of the questions in each questionnaire. For each questionnaire, the category thresholds were based on the distribution of Likert values across the range of possible answers, as shown in Table 5.12.

Table 5.12: Pre- and Post-Simulation Questionnaire Aggregation Categories

Category	**Value Range**	**Corresponding Answer(s)**
Pre-Simulation Questionnaire		
Low	$[1, 3)$	None/Low
Medium	$[3, 5)$	*middle of range*
High	$[5, 7]$	Extensive/High
Post-Simulation Questionnaire		
Negative	$[1, 3)$	Strongly Disagree/Disagree
Neutral	$[3, 4)$	Neutral
Positive	$[4, 5]$	Agree/Strongly Agree

The aggregation results for each questionnaire are shown in Table 5.13.

Table 5.13: Pre- and Post-Simulation Questionnaire Aggregation Results

Category	**Num students**	**Average score**
Pre-Simulation Questionnaire		
Low	4	2.30
Medium	7	3.56
High	12	4.83
Post-Simulation Questionnaire		
Negative	0	N/A
Neutral	12	3.41
Positive	7	4.19

Another way that one can look at these results is on a per-student basis, to see whether students' attitudes towards the simulation changed between the pre- and post-simulation questionnaires. In Table 5.14, below, shows the number of students that fall into each change category ("Low to Negative", "Low to Neutral", etc.), the trajectory for the change (downward, no change or upward), and the total number of students that fall into each trajectory category.

Table 5.14: Pre- and Post-Simulation Questionnaire Aggregation Changes

Pre-Sim Category	**Post-Sim Category**	**Trajectory**	**Num students**
Low	Negative	no change	0
	Neutral	upward	1
	Positive	upward	1
Medium	Negative	downward	0
	Neutral	no change	5
	Positive	upward	2
High	Negative	downward	0
	Neutral	downward	6
	Positive	no change	4
Trajectory Totals			
Downward			6
No change			9
Upward			4

The other type of results that I have from the experiment is the debriefing session content, which provides a more detailed picture of the students' experience. One of the biggest themes that came out of the debriefing sessions was the importance of teamwork and communication in effectively treating the patient as a group. In particular, the opportunity to work together as an interdisciplinary team, and taking advantage of each participant's skills and knowledge, was something that the students really appreciated. Some comments on this theme were:

> *Communication is key, working together as a team. Otherwise it's just going to be disjointed, and you'll have different aims. Then, essentially, the patient won't be treated as well as that patient should be.*
>
> *We communicated well, I thought, still listened to each other, bounced ideas off each other.*
>
> *There's stuff we all don't know, so we always just ask each other. I didn't know what a proper heart rate was, you guys knew it was elevated, so that's good.*
>
> [*I appreciated*] *getting to know the respiratory therapists and what they did. I didn't know they would check the respiratory rate... I had no idea what was going on with the respiratory rate, so it's good that you guys knew "oh, increase the oxygen" and even knew what meds to give them.*
>
> [*We did well at*] *delegation, we divided the tasks at the beginning, so we know who's taking care of what.*
>
> *I think at the end we kind of pulled together and started working as a team really well.*
>
> [*I appreciated*] *the team aspect where we had to bounce ideas off each other, find out about the oxygen, ventolin, atrovent, just make sure we know exactly what's going on with the patient, don't just assume things happen or don't happen.*

The facilitator also observed this, as indicated by the following comments:

> *Initially you guys were quite shy with each other, and we had pharmacy and respiratory therapy sort of in their own individual roles... and then you guys really started to collaborate nicely, clarifying what you were doing, telling each other what you were doing, clarifying "what's the dose of ventolin?"... It seemed that you guys weren't uncomfortable sharing your knowledge with each other, but you respected what someone else would say, and you would take that into account. That's all part of interdisciplinary teamwork.*
>
> *You clarified that orders were being done or interventions were being implemented appropriately. I really liked that you were constantly ... sharing your findings.*

Conversely, students were also aware of the difficulty of taking on an unfamiliar role – especially the pharmacy and respiratory therapy students who were asked to take on nursing role. One student observed that

> *It's the same with pharmacy, because we usually don't take care of vitals, we take care of the medication aspect of things, and I was like "Oh, right, I should take her body temperature" and all that kind of stuff, and going outside our scope of practice, of what you're used to.*

Another interesting theme that came up in several debriefing sessions was the importance of managing the physical space around the patient – ensuring that the students weren't getting in each others' way, and making sure that each person was able to get to the "right place" around the bed to effectively treat the patient.

> *I found at times it was hard 'cause if your avatar was standing by the bed and someone else came up you got pushed out of the way, so if you were trying to click on a hand you couldn't cause somebody else came in and sort of shoved your avatar out of the way.*
>
> *I thought it was very realistic in the way that we were getting in each other's way, 'cause it really does happen like that.*

The students also expressed that the technology aspect of the simulation was a challenge. Several students found it challenging getting used to the user interface:

> *We’re all trying to figure out how to work the system, so that lags us back.*
>
> *I didn’t realize at first that you had to keep clicking to get to stuff, and once I figured that out, it’s like “I’ll just keep clicking and see everything that I can get out of it”, and then report some of the findings.*
>
> *It was kinda clumsy to try and do things, I still don’t know where the medication is!*
>
> *I didn’t know how to work with it until it was too late.*

This was a concern which likely could have been mitigated by providing a more comprehensive training session. Because each round was only an hour long in total, the training session was limited to approximately 10 minutes, which was not sufficient for some of the students.

Other students, however, expressed reluctance or opposition with regards to technology in general:

> *I wish I just could be there in real life, and just do everything hands-on, but just involving the computer and having a different way of accessing everything made it a bit more complicated.*
>
> *Me and technology are not friends.*
>
> *If I could use my hands I’d be a lot happier.*

5.3.6 Collaborative Study – Discussion

As described in the preceding section, the results of the experiment fall into two categories: learning outcome data that indicate the degree to which students learned about the patient assessment process, and debriefing and opinion data that indicate the quality of the students’ experience.

In the first category the learning outcome data consists of the pre- and post-simulation test results. I found that there was very little difference between the pre- and post-simulation test results, which may be explained by several factors.

In the initial design of the experiment, each nursing student was to go through the simulation individually, and the simulation would then provide the student with feedback on whether his or her choices were a) made in the correct order and b) appropriate for the patient’s condition. However, because I conducted the scenario with groups of students, this feedback became less helpful (since the group of students was much less likely to perform their actions in a coordinated, organized fashion), and was replaced by the conversational feedback provided by the nursing instructor in her debriefing sessions. While this feedback was certainly useful and relevant, it did not necessarily address the particular skills tested in the pre- and post-simulation quizzes. Therefore, each group of students may not have received the full benefit of the scenario’s post-simulation feedback mechanism.

Second, as mentioned earlier, the students were from a range of disciplines (nursing, respiratory therapy, and pharmacy), and at various stages of training (from 2nd to 4th year). Because of this range of educational backgrounds, they may not have had the necessary background in patient assessment and treatment for a single simulation session to make a significant difference in their overall ability in this area. Moreover, the test itself was designed with nursing students in mind, and thus may not have been entirely appropriate for students from other disciplines.

With regards to the questionnaires, the results were generally quite positive. The questions were scored using a 5-point Likert scale, with 1 representing "strongly disagree", 3 representing a "neutral" rating, and 5 representing "strongly agree." The only question that received a score below "neutral" had to do with the realism of the patient entity. Conversely, questions concerning the overall learning outcomes, the students' ability to apply medical facts to solve the simulation, and the application of communication skills all received average scores corresponding to a rating of "agree" or better.

From the pre- and post-simulation questionnaire aggregation results, one can see that the responses from the pre-simulation indicated a generally high level of previous experience and simulation expectations, although this aggregation obscures the very low average rating for the "VW experience" question – that is, while students had a generally high level of expectations, this was *not* based on previous experience with virtual worlds. The post-simulation responses, meanwhile, showed a generally neutral-to-positive response to the simulation, with no students expressing a negative opinion, on average. Since there were slightly fewer students who completed the post-simulation questionnaire, the total number of students across the post-simulation categories is slightly less than for the pre-simulation analysis (23 students completed the pre-simulation questionnaire, compared to 19 for the post-simulation questionnaire).

From the per-student aggregation results, meanwhile, one can see that the most frequent patterns were "medium to neutral" and "high to positive", which indicate a consistent opinion, while the most frequent change was from high to neutral. This, in turn, indicates that the majority of students had their expectations at least met, if not exceeded, through participating in the simulation experience. Moreover, while a significant portion of students found that their high expectations were not fully met by the scenario, these students still had an opinion of the scenario that was neutral or better. It is also worth noting, as mentioned earlier, that these high expectations were not based on previous virtual world experience, and thus some adjustment to the reality of what virtual worlds can (and cannot) do is to be expected.

To sum up, a few key results can be gleaned from this experiment: first, students were able to gain valuable inter-professional communication skills and collaboration experience through their participation in the simulation. This was a theme that was highlighted in the students' comments in the debriefing sessions, and also appeared in students' questionnaire feedback. Second, the inter-professional, collaborative nature of the sessions was, while beneficial in some ways, a barrier to obtaining definitive pre- and post-simulation objective assessment results. The differences between students' pre- and post-test scores was, although positive, quite small, and I ame thus unable to draw any conclusions from this result. Third, the aggregated results of the students' questionnaire feedback indicates that students were, on the whole, quite happy with their simulation experience.

5.3.7 Individual Study – Procedure and Results

Individual Study – Procedure

I performed a run of the ER experiment with 4th-year nursing students in March, 2012, this time asking the participants to go through the scenario individually, rather than in a group. As with the previous experiment runs, I conducted pre- and post-simulation tests to assess the students' level of competence, and also gave the students pre- and post-simulation questionnaires, to assess their opinions about the scenario. For this run of the experiment, I also administered Allen's tool for measuring attitude toward computer assisted instruction, both before and after the simulation [2]. In this run of the experiment, I had students go through the scenario individually, in order to focus on the procedural (rather than collaborative) aspect of the scenario.

It should be mentioned that I recruited participants for a control group, who did not participate in the scenario, but continued to take part in their usual classroom-based activities. These participants were asked, like the experimental group members, to fill out questionnaires and tests at the beginning and end of the simulation period, to assess their level of competence and opinions about virtual world-based training. While the initial response was quite positive, students in the control group did not follow through on the post-simulation portion of the experiment; thus, their participation in the experiment was not useful in forming any kind of comparison with the experimental group, and has been omitted from these results.

Individual Study – Results

A summary of the pre- and post-simulation test results is shown in Table 5.15, with the full results shown in Tables H.2, H.3, and H.4.

Table 5.15: Pre- and Post-Simulation Test Results

		Test Components			
Participant	**Num students**	**Questions to Ask**	**Assessments**	**Actions**	**Total**
Pre-Sim Avg	7	5.14	10.43	10.29	25.86
Post-Sim Avg	7	5.14	10.00	10.86	26.00
Average Difference		0.00	-0.43	0.57	0.14

In terms of individual participants' results, three of the participants improved their results, three participants saw a decline in their results, and the remaining participant stayed the same.

As can be seen from these results, the students did not show any consistent improvement, in terms of the pre-simulation vs. post-simulation test scores. While some students improved their scores, an equal number earned a lower post-simulation score, with one student showing no change.

The pre-simulation questionnaire, whose results are summarized in Table 5.16, focused on the students' previous expectations and experience with simulation-based training and virtual worlds. The full pre-simulation questionnaire results are shown in Table H.6.

The post-simulation questionnaire, as with the questionnaire from the usability testing session, used a five-point Likert scale, where a value of 1 corresponds to "strongly disagree", while 5 corresponds to "strongly agree". The questions included in the question-

Table 5.16: Pre-Simulation Questionnaire – Likert Question Results

Rating Scheme	Min	Max	Avg
1. Rate your previous experience with educational simulations.			
None 1 - 7 Extensive	2	7	4.67
2. Rate your previous experience with virtual worlds/online environments.			
None 1 - 7 Extensive	1	2	1.17
3. What are your expectations for the simulation?			
Low/Pessimistic 1 - 7 High/Optimistic	4	7	5.00
4. What are your expectations for the virtual world interface?			
Low/Pessimistic 1 - 7 High/Optimistic	2	7	5.33

naire, along with the aggregated results, are shown in Table 5.17. The full post-simulation questionnaire results are shown in Table H.8.

Table 5.17: Post-Simulation Questionnaire – Likert Question Results

Question	Min	Max	Average
1. This simulation has enhanced my understanding of the patient analysis and treatment process.	3	5	4.00
2. The level of realism was sufficient to **suspend disbelief** in the following areas:			
a. Interacting with the patient	2	4	3.43
b. Using medical equipment	4	4	4.00
c. Treating the patient	4	4	4.00
d. The hospital/ER department	2	4	3.14
3. The level of realism was sufficient to **enable learning** in the following areas:			
a. Interacting with the patient	3	5	3.86
b. Using medical equipment	4	5	4.14
c. Treating the patient	4	4	4.00
4. I was able to apply the following knowledge and skills in completing the scenario:			
a. Knowledge of medical facts	4	5	4.14
b. Knowledge of relevant procedures	2	5	3.86
c. Communication skills	2	4	2.71
5. Throughout the scenario, I understood what I needed to do next	3	4	3.71
6. The experience was interesting, and I felt engaged in the experience	4	4	4.00
7. The scenario was well organized	3	4	3.86
8. The following activities were useful in facilitating my learning:			
a. Pre-simulation instructions	4	5	4.14
b. Pre-simulation practice	3	5	4.00
c. Simulation experience	4	5	4.43
9. I would recommend this experience to other students	4	5	4.29
10. Overall, this was a useful learning experience, in that it added to my:			
a. Textbook learning	2	5	3.71
b. Classroom learning	2	5	3.57
c. Practical learning	4	5	4.57

From the free-form section of the post-simulation questionnaires, some interesting opinions on the scenario were elicited.

One of the themes that emerged from this feedback was that the scenario provided the students with a low-pressure environment within which to practice their skills.

> *It is a non-threatening environment to learn about assessing and treating patients.*
>
> *Not as scary as the first day of clinical!*
>
> *A chance to try out basic interventions without "real" consequences on a real patient.*
>
> *Low-pressure learning experience.*

Another recurring theme was the overall utility of the scenario, and general appeal of the virtual world-based learning environment:

> *This was a really good experience! I think everyone should give it a try.*
>
> *I think these types of training and practise methods would be incredibly uvaluable in nursing programs.*
>
> *Would be great to have a number of these scenarios before our first clinical experience for clinical.*
>
> *This would be a good learning tool to have.*

Finally, for this run, I administered Allen's tool for measuring attitude toward computer assisted instruction [2]. The tool uses 14 bipolar adjective scales (e.g., "valuable" vs. "worthless"), and asks the student to record his or her first impression about each adjective. The summarized results of administering the tool before and after the simulation are shown in Table 5.18. The full results are shown in Tables H.10, H.11, and H.12.

Table 5.18: Pre- and Post-Simulation Attitude Assessment Results

	Pre-Simulation			**Post-Simulation**			**Difference**
Item	**Min**	**Max**	**Avg**	**Min**	**Max**	**Avg**	**Avg**
Flexible	2	7	4.83	4	7	5.43	0.83
Useful	4	7	5.33	5	7	6.43	1.00
Stimulating	4	7	5.33	5	7	6.00	0.83
Meaningful	2	7	4.67	5	7	5.71	1.17
Pleasant	4	7	4.83	4	7	5.86	1.00
Valuable	3	7	4.67	5	7	6.00	1.33
Creative	5	7	6.00	5	7	6.43	0.50
Personal	2	4	2.50	1	5	3.71	1.17
Efficient	4	6	5.00	3	7	5.29	0.17
Appropriate	4	6	5.17	4	7	5.71	0.50
Comfortable	4	7	5.33	6	7	6.29	1.00
Non-threatening	2	7	5.50	6	7	6.43	1.00
Easy to control	3	5	4.00	5	7	5.86	1.83
Timesaving	2	5	3.50	3	6	4.86	1.50
Total	52	85	66.67	69	93	80.00	13.83

5.3.8 Individual Study – Discussion

As with the collaborative scenario-based run of the scenario, students had generally high expectations going into the scenario, although these expectations were not supported by much virtual world experience. In fact, it is notable that the *highest* value given for the "previous virtual world experience" question was 2, which indicates a very low overall level of experience among the participants.

Again, many of the post-simulation questionnaire results are consistent with those from the collaborative scenario-based run; the opinions expressed by the students were generally quite positive, with especially high results for questions relating to the overall quality of the experience (Q1, Q9), application of knowledge (Q4a, Q10c), and level of engagement (Q6).

The notable exception to this trend is the question about the application of communication skills. In the multi-participant "Save Stan" run, this question received the highest response, with an average score of 4.37 out of 5. In this run – where participants went through the scenario individually – this was the only question to get an average score below 3 (2.71). This difference highlights the importance of including multiple participants in conveying communications-related lessons. That is, while there was *some* student-to-patient communication in the single-participant version of the scenario, students did not see it as having nearly the same impact as the inter-student conversations that occurred in the multiple-participant version.

From the attitude assessment results, it is clear that students' attitudes towards computer-aided instruction improved after going through the simulation. Every item in the assessment tool showed at least some improvement, with particularly large gains in the "valuable", "timesaving", and "easy to control" items. It is also notable that in several items – particularly "meaningful", "easy to control", and "non-threatening" – the lowest score increased significantly, indicating a change from a mixed or mostly negative opinion to a more consistently positive attitude for those items. These results, which are consistent with the conclusions drawn from the questionnaire, indicate that students felt quite positively about their experience with the simulation, and felt strongly about the overall value and usability of the virtual world-based scenario.

In summary, while the students did not show consistent improvement in their test scores, there were several results that were quite supportive of the value of the virtual world-based scenario. First, the numerical results from the post-simulation questionnaire indicated that students felt quite positively about the scenario, in terms of its immersiveness, practical applicability, and the overall quality of the experience. From the free-form questionnaire answers, meanwhile, students expressed appreciation for the low-stress environment provided by the virtual world scenario, and the potential of the virtual world-based approach for providing educational and practice experiences. Finally, in measuring students' attitude toward computer-aided instruction before and after the simulation, I was able to observe a consistent improvement in the students' attitude. This provides further evidence for the positive attitude shown by the students towards the virtual world-based scenario, and for the willingness shown by students to participate in, and be engaged by, this kind of learning experience.

5.4 Threats to Validity

As with any experiment, there are concerns about the validity of the experimental results that must be addressed.

First, there is the challenge of measuring the students' pre- and post-simulation knowledge and skills in an appropriate and timely fashion. If the students are tested too soon after the simulation experience, the lessons from the experience may not have had time to have an impact on the students' understanding of the skills and concepts being taught, and thus there might not be much noticeable effect, regardless of the effectiveness of the simulation. Conversely, if the post-simulation evaluation occurs too long after the simulation, the effect of the simulation experience becomes confounded with students' subsequent experiences – classroom lectures, other learning exercises, etc. – and it becomes increasingly difficult to conclusively identify the effect of the simulation experience on the students' level of knowledge. Because of the structure of my experiments – working with some students right before the end of term, recruiting others as part of a larger simulation event – I did not have as much control over the timing of the pre- and post-simulation tests as I would have liked. In the case of the experiment that was conducted as part of the Save Stan event, the post-simulation evaluation was probably too soon after the simulation experience, and this may have reduced the quality of the results.

Second, there are the shortcomings and fidelity-related issues that are an unavoidable part of creating a simulation of a real-world environment or experience. That is, in creating a virtual world representation of a particular environment, object, or behaviour, the VW representation will not be *exactly* the same as its real-world counterpart. The goal, of course, is to create a VW representation that is "close enough" to the real thing, in terms of its functionality, appearance, and relationship to the rest of the scenario. To this end, I consulted extensively with experts and educators in creating each scenario, pilot-tested the scenarios with students, and generally attempted to ensure that my VW scenes, objects and behaviours were *sufficiently realistic* for a given scenario. However, this process is not foolproof, and something that was an irrelevant detail to one student may well have seemed like a glaring error to another.

Third, it is possible that there is some inherent quality or characteristic of the virtual world experience that is detrimental to learning. That is, there could be some aspect of the user interface, for example, that causes students to have difficulty learning, no matter how well-designed the scenario might be, or how realistic the objects and interactive behaviours. Hopefully by conducting further standardized, repeatable experiments with VW-based scenarios, and comparing these results to equivalent "traditional" exercises, I can either dispel this concern or, at worst, understand it better and work to mitigate it.

5.5 Towards a Scenario Development Methodology

Based on my experience with the case studies and experiments described in this section, I propose the following broadly-applicable guidelines.

5.5.1 Facilitator and Instructor Involvement

When running an educational program, co-ordination with facilitators and instructors is crucial, on several levels. On a practical level, it is essential that everyone involved in the project understands their role, their responsibilities, and the way that they will fulfill their role within the virtual world. This kind of common understanding is essential to running a scenario as smoothly as possible.

From an educational point of view, the instructors should be consulted at the start of the development process to ensure that the learning objectives are clear, and that they can be met through the proposed scenario. It is, after all, the instructors who will assess students based on these learning objectives, so it is important that there is a shared understanding of these objectives and how they are to be met by the students. As the development of the scenario proceeds, the instructors should be consulted regularly, to ensure that the implementation of the scenario is consistent with the initial educational vision, and that the details of the scenario (setting, appearance of objects, required tasks, etc.) will be consistent with students' expectations, based on the students' classroom experiences.

5.5.2 Mandatory Virtual World Training

As I have found, the ability to use a virtual world effectively is not acquired instantaneously, even for students of the "digital generation." Rather, most students will require some training in using their avatar, and mastering the fine points of movement, communication, and camera control. Moreover, students will, typically, not realize that they need this training until they are "stuck" - they are unable to communicate, for instance, or their avatar ends up in an unexpected location.

From this experience, I have found that offering a voluntary training session is insufficient, since the students who need to attend the training session may not go, assuming that they know enough already. This will put some students at a disadvantage when it comes to running the program, and may cause delays for the rest of the students, as well. Rather, it is important to make the session mandatory, to ensure that everyone has the same set of skills, and level of comfort, in using the virtual world. These sessions should include instructors and facilitators, as well as students – *all* of the scenario participants need to be sufficiently comfortable with the virtual world to fulfill their roles effectively and efficiently. Finally, one should make sure that the tasks students are attempting in the training session are related to what they'll be doing in the education session; however, the tasks should not be so similar as to "give away" any new experiences or challenges that will be presented during the session.

One way to facilitate this kind of training session is to create a separate training area, which contains simplified items *similar to*, but not exactly the same as, those that will be used in the scenario. In this training area, a scenario facilitator can provide students with training in the basics of controlling their avatars, and also give the students a chance to familiarize themselves with particular virtual world objects that will be used in the scenario. This approach was used in the context of the EMT handoff scenario to allow students to become comfortable with relevant activities – driving an ambulance, pushing a gurney and interacting with a victim – without actually exposing the students to the accident rescue scenario.

5.5.3 Planning for Technical Problems

As virtual worlds are an emerging technology, there are a number of technical problems that may arise, and one should be as prepared as possible for these difficulties. First, be sure to test, test, and then test again! Test the code in individual components, the interaction between components, and the computers and networks used to run the virtual world software. As much as possible, try to duplicate the "real" environment when testing, in order to obtain a realistic idea of how the system will behave when the time comes to run the educational program. This includes factors such as the software, hardware, peripherals, network connection, and number of participants in the virtual space, each of which can have an effect on overall performance.

On this note, one should be conservative when assessing whether a computer can run virtual world software. Barely meeting the minimum requirements, whether in terms of memory, graphics capabilities, or internet connection speed, may result in a frustrating experience for students and facilitators, and a poor outcome for the program.

Another potential problem is space requirements for creating a digital recording of a program. These recordings, while comprehensive and accurate, are very large, often taking many gigabytes of space for quite brief segments of video. In order to be adequately prepared, one should ideally have an extra external hard drive on hand, in order to store the recording of the sesion.

One should allow users time to get set up on their computers, log into the virtual world, and generally get ready to participate in the educational program. These steps all take time, and if this is not accounted for, students will feel rushed and facilitators will end up feeling behind schedule and disorganized.

Finally, one should have backup plans in place for as many of the program components as possible. People may not log in on time – or at all – computers might crash, and internet connections can fail. If one has contingency plans, then these obstacles can be dealt with smoothly and effectively, rather than becoming insurmountable obstacles.

5.6 Summary

In this chapter, I have presented three empirical evaluations, each of which dealt with a different educational context, had its own distinct characteristics, and provided insight into different aspects of the MeRiTS framework. The following table sums up these evaluations.

5.6.1 Connection to Educational Theories

The last column in the table, Related Theories, ties the evaluations presented in this chapter to the educational theories discussed in Chapter 2.

First, Kolb's theory of experiential learning [50], and de Freitas' related exploratory learning theory [17], are connected to the EMT and ER evaluations. Both of these evaluations followed an experiential/exploratory learning approach, in that the participants experienced the scenario, and were then able to, through a debriefing conversation, reflect on the experience and (potentially) revise their mental models. The repeatability of the scenarios meant that, had time permitted, the participants could have revisited the scenarios and tested their learning. Finally, the emphasis by de Freitas on the social aspect of learning is reinforced by the co-operation shown in both the EMT handoff scenario and the collaborative version of the ER nursing experiment.

The Cognitive Apprenticeship model proposed by Collins [14] is also quite relevant here, as the aim in both evaluations was to provide the participants with authentic experiences that could help them master the relevant skills (in this case, decision-making and

Table 5.19: Summary of Empirical Evaluations

Name and Focus	**Characteristics**	**Lessons Learned**	**Related Theories**
EMT: accident scene rescue and handoff	roles: two EMTs (lead and supporting), one ER staff multiple scenes, outdoor (accident site) and indoor (hospital) objects: victim, ambulance, gurney, spineboard, medical equipment workflows specify victim behaviour, interaction with equipment participants: 2 EMT students from NAIT	Can qualitatively assess student performance, role adoption	Experiential learning [50], Cognitive apprenticeship [14], Educational co-creation [59]
Marketing: negotiation exercise with variety of avatars	roles: buyer and seller single indoor scene: simple cubicle object: contract to confirm negotiation workflow specifies negotiation confirmation process participants: 30 undergraduate business students from U of A	Can record participants' actions and provide useful analysis	Discourse [9], non-verbal communication [58]
ER Nursing: assessing and treating patient	roles: nurse, RT and pharmacist single indoor scene: hospital ER ward objects: patient, medical equipment workflows specify patient status, trajectory, reactions to treatment participants: 27 students in Nursing, RT, Pharmacy from U of A, NAIT, Grant MacEwan	Can teach procedural, teamwork and collaboration skills	Collaborative learning [45], Exploratory learning [17]

communication skills). Again, the repeatability of the scenarios is crucial, since it allows the students to practise these authentic skills, and enables the instructors to participate in a variety of ways – as an expert modelling the “correct” approach, as a confederate joining in the activity, and finally as an observer.

Finally, the importance of communication is a common theme in all three experiments, which is connected to several educational theories – educational co-creation [59], discourse [9], non-verbal communication [58] and collaborative learning [45]. Messinger's proposal that educational co-creation can be achieved through virtual a world was supported by the EMT handoff scenario, which involved collaboration among students and instructors. The theories of discourse and non-verbal communication, while not an intentional focus of the marketing experiment, were certainly relevant to that situation. Finally, as implied in the first paragraph of this section, the collaborative nature of the ER nursing experiment is connected to Johnson's work on collaborative learning.

To sum up, the educational theories at the core of my research – experiential and exploratory learning, cognitive apprenticeship, and theories related to communication – not only informed the definition of the models used in the MeRiTS framework, but were also strongly tied to the empirical evaluations that I conducted.

5.6.2 Conclusions

There are several broad conclusions that may be drawn from the results of my case studies and experiments.

First, when training students in a procedure for which there is a clearly-established "correct" sequence of actions (e.g., go to X, perform action Y, and then to go Z), a MeRiTS-based VW simulation provides an effective environment for practicing this procedure, recording this experience, and analyzing the action trace to compare each student's actions to the correct sequence.

Second, the capability for students to enter the VW and adopt specific roles – that of an EMT, for instance, or an ER nurse – enables the students to gain valuable experience in interdisciplinary communication and collaboration skills. This experience is difficult to obtain in real-world training exercises, and the use of a VW breaks down many of the logistical and practical barriers that are associated with real-world exercises of this type.

Third, the action recording and analysis capabilities of the MeRiTS framework allow the instructor to analyze students' actions from a variety of perspectives, and to understand these actions both on an individual and group interaction level.

Finally, the VW environment – and the MeRiTS framework specifically – provides a flexible platform for the creation of a variety of scenarios, teaching procedural or communication skills, for either individuals or groups of students. Students have shown a willingness to try technology which may be unfamiliar to them and, especially when provided with sufficient training, have expressed largely positive feelings about their experience.

Chapter 6

Contributions and Future Work

Virtual worlds are increasingly being adopted as platforms for conducting simulations in the context of competency-based training programs. In my work in this area, I have identified the need for a virtual world-based training system that allows instructors to not only deliver educational experiences to students, but also record, analyze and understand the students' actions as they go through the training session. To meet this need, I have developed a framework for developing and enacting virtual world-based scenarios, which supports the recording and analysis of participants' in-world behaviour.

6.1 Contributions

In my work, my central thesis is that virtual worlds offer a cost-effective platform for simulation-based training. Through the MeRiTS framework, I have made several several contributions which support this thesis.

The first contribution is the creation of **the MeRiTS framework** itself, which is described in detail in Chapter 4. The framework takes advantage of the potential of the virtual world platform by supporting the development, enactment and analysis of simulations for educational purposes. The framework is based on a generic model of in-world user activity, the Avatar Behaviour Model (ABM), described in Chapter 3. This model describes, in an implementation-independent manner, the educationally-relevant actions that a user can take within a virtual world. Specifically, based on relevant pedagogical theories, I have focused on three types of actions: movement and exploration, experiencing the world, and social interaction. The ABM provides a solid foundation for the framework, upon which I have developed scenario definition, enactment, and recording tools. Based on this underlying foundation, I have developed models for the specification of virtual world-based scenarios (described in Section 4.3), and designed a scenario execution engine that drives the enactment of these modelled scenarios (see Section 4.5). I have also had extensive experience in a particular virtual world (Second Life), which has been accumulated through the process of implementing several scenarios. Through this process, I have developed solutions for several virtual world-based design challenges (discussed in Section 4.4), which are applicable to other virtual worlds, as well.

There has been similar work in this area by Taylor et al., who have also developed a generic framework for virtual world-based simulation [77], and Nakanishi et al., who have developed a language for specifying participant-driven scenarios in a virtual world [62]. However, Taylor's framework is limited to describing virtual patient behaviour, and does not apply to other kinds of interactive objects. Nakanishi's Q language, meanwhile, lacks some features included in MeRiTS – timed events and constraints – which make the scenario behaviour definition capabilities provided by MeRiTS significantly more expressive and comprehensive.

The second contribution is **a comprehensive recording component and accompanying set of analysis tools**. The recording tools (described in Section 3.3) are based on the ABM, and thus can be implemented in any virtual world. In implementing the recording tool, I have drawn on research from educational psychology to ensure that the tool is not only technically cohesive, but also pedagogically appropriate. The analysis tools (presented in Section 4.6), meanwhile, provide instructors with a powerful means of understanding students' actions in a virtual world, and answering a wide range of research questions. At this point, the toolkit includes three tools. First, statistical-analysis tools provide instructors with a succinct, meaningful summary of the students' activities, both individually and as a group, by distilling a comprehensive record of each student's actions. Second, the pattern-analysis and recognition tool enables instructors to analyze patterns of behaviour and thus draw conclusions about groups of students, or particular students over several sessions. Third, the video annotation component enables instructors to view a digital recording of all students' behaviour, augmented by annotations that describe each student's object interaction, movement, and communication activities.

Relevant work has been done in this area by Hurst, who identified three types of recorded action data that should be captured for virtual world-based scenarios [42]. These three types – reflective data, machinima, and environment data – are consistent with the kind of information that is captured by the behaviour recording and analysis tools provided by MeRiTS. However, Hurst's work was limited to identifying these types of recorded data, and there was no effort to actually implement a system that would capture this data. Another relevant project is the "River City" environment developed by Nelson et al., a custom-built, game-oriented virtual world which includes an integrated tool to track students' actions [64]. This tracking system is used to inform a student guidance system which, in turn, provides the students with customized hints as they participate in the game. While the tracking system has an educationally-oriented focus, and the recorded actions are used for pedagogical purposes, there are several key differences as compared to my work. First, the set of actions that are recorded are focused on the "River City" environment, and are not necessarily applicable to any other context or virtual world platform. For example, the "talking to town residents" action is not necessarily applicable to other educational contexts, and the "reading signs" action may not be possible in another virtual world. Another difference is that the recorded data is used as input to another component (the student guidance system), rather than being shown to instructors or students as a part of the educational process. One other project worth noting is by Marty, who integrated a variety of recording capabilities into a "pedagogical dungeon" [54]. This work shares some characteristics with my efforts, including the creation of recording tools based on a foundation of several basic action types, and a set of analysis tools that allow these basic types to be aggregated into more readily understandable activities, from an educational point of view. However, Marty's recording tools were created for a custom game platform, and thus cannot be applied to a general-purpose virtual world. Moreover, the analysis capabilities offered by Marty's tools are not nearly as extensive as those provided by MeRiTS.

Third, to validate the utility of the framework, **I have, in consultation with instructors, developed several scenarios using the MeRiTS framework (presented in Chapter**

5), and evaluated these scenarios in a variety of ways. In the most rigorous experiment, I conducted an experiment (see Section 5.3) which assessed the educational effectiveness of the virtual world simulation. In this experiment, I assessed the educational outcomes for students participating in the simulation, provided qualitative evaluation of their performance through an instructor-guided debriefing session, and also elicited feedback from the students on their experience using the virtual world simulation. I also conducted two other supporting case studies (described in Sections 5.1 and 5.2) which provide validation for the communication skills training and behaviour recording and analysis aspects of my work.

A few other researchers, such as Dev and Greci, have undertaken similar studies of the effectiveness of virtual world-based training [19], [31]. These studies indicate that, in the contexts analyzed by the researchers (emergency medicine and crisis response, respectively), virtual world-based training does indeed allow students to meet educational goals. Dev showed that, by taking part in a series of virtual medical cases, students were able to improve their team performance skills, which were rigorously evaluated by expert observers. Greci used training exercises along with reflective learning sessions to provide students with both a realistic experience and meaningful reflection on this experience improved communication and decision-making skills. In both of these studies, however, the focus of the evaluation was on virtual world scenarios that were designed for specific contexts. While the researchers provide ample evidence of the effectiveness of these particular scenarios, they do not provide the means for these results to be extended to other contexts via the virtual world being evaluated. In my work, I provide both evaluation of several scenarios implemented using MeRiTS, as well as a generally-applicable framework that can be applied to a wide range of other educational contexts.

Finally, through my scenario development experience, I have gained insights into the process of developing and delivering a virtual world-based scenario, which are relevant for other researchers or educators going through this process. These guidelines, presented in Section 5.5, are general in nature, and can be applied to projects from a range of educational contexts and virtual world implementation situations. Some of these guidelines are echoed by Cai, in his high-level analysis of the use of virtual worlds for education and training [8]. In this work, Cai presents a seven-stage process for designing v-learning solutions, including describing learning objectives, developing the game, and conducting a post-deployment evaluation. However, Cai does not advocate, as I do, for the involvement of instructors throughout the development process, and his analysis does not include specific recommendations for the deployment and execution of the project.

6.2 Future Work

I am considering several areas for future work. One of these is developing additional statistical analysis tools, to provide instructors with further insights into students' in-world behaviour. For instance, I am investigating the analysis of conversations, to understand which students frequently participate in conversations, whether one person tends to interrupt another, and whether some students are excluded. I want to further develop the gaze analysis tool to allow the instructor to determine what the student is looking at, and whether this gaze target is relevant for the student's current activity, rather than just reporting on changes in gaze direction.

Another area that warrants further investigation is the implementation of the virtual world client – and, particularly, the recording tools – for another virtual world. At this time, I have focused my development efforts on Second Life, as it is the virtual world with which I have had the most experience. However, there are other worlds, such as Open Wonderland and Sirikata, which may be used for educational simulation. I could implement my client and associated recording tool within these virtual worlds, and see if there are any differences in efficiency or recording capabilities which emerge from this implementation process.

A third area that I am interested in exploring is the automatic incorporation of avatar gestures into certain kinds of interaction. For instance, when taking a patient's blood pressure, a student's avatar could automatically perform a gesture that would show the avatar leaning over patient's arm and affixing a blood pressure cuff, and then after a few seconds removing the cuff. This would be very useful in multi-participant scenarios, where each participant is provided with a very limited set of visual cues to understand what other participants are doing during the enactment of the scenario. This was an issue that was raised during the evaluation of the EMT scenario, and also would have been helpful in the "Save Stan"-based ER nursing experiment. While Second Life offers the ability to create custom gestures (which is, of course, an essential requirement for this feature), there are other barriers to be overcome in implementing this feature. The most significant of these is that, in order for the gesture to be executed in a realistic manner, the avatar and the relevant object must be rotated appropriately. In the blood pressure cuff example, the avatar must be facing the patient's arm before he or she leans over to affix the cuff, otherwise the gesture will not make any sense to other participants.

Finally, my colleagues and I are beginning to investigate the integration of input from outside sources, such as mobile phones and gaming systems, into the enactment of a virtual world-based scenario. The integration of input from smartphones is particularly appealing in terms of the potential for developing MeRiTS-based augmented reality games (ARGs), which combine virtual-world and real-world experiences into a cohesive, "blended" game. Typically, these games involve smartphone-based activity in a real-world setting (scanning QR codes, GPS-tracked movement, gyroscope-tracked phone orientation), with an additional layer of information or interactivity conveyed through a virtual world. Incorporating input from gaming systems, meanwhile, will allow us to take advantage of the kinesthetic intutitiveness of motion-based systems such as the Nintendo Wii and Microsoft Kinect, which treat the user's body position, weight, and movement as input. By integrating input from these systems, we can enable participants to move in the real world, and see those movements reflected in the virtual world. For situations involving physical coordination among multiple participants – a surgical team managing the physical space around a patient, for example – this kind of kinesthetic input could be significantly more intuitive and appealing than the standard keyboard- and mouse-based input offered by a standard computer interface, while requiring much less expense and overhead than a full-fledged virtual reality system.

6.3 Conclusion

In conclusion, I have presented a novel, robust virtual world-based educational platform, supported by relevant educational psychology research. From a functional perspective, the framework provides the necessary scenario definition and execution capabilities for creating and delivering a wide range of virtual world-based training scenarios. From a design point of view, the framework is based on my model of avatar behaviour which, in turn, is informed by research on educationally-relevant activities in a virtual world context. The components built upon this framework, meanwhile, include a set of recording and analysis tools that enable the instructors to assess students' behaviour in a comprehensive manner. Through these components, the instructor is able to not only provide students with an immersive, interactive experience, but also determine whether, through this experience, the students have met the educational goals of the scenario.

The empirical evaluations that are presented provide evidence of the framework's utility and educational relevance, through several case studies and experiments in a range of contexts. Through a case study in an EMT training context, I provide support for the functionality of the framework in delivering a collaboration-focused training scenario, and of the capabilities of the recording and analysis tools in assessing the students' behaviour while participating in this scenario. In a marketing case study, meanwhile, I provide further demonstration of the range of analysis capabilities incorporated into the system, including conversational content analysis and correlation of a key outcome with various experiment session characteristics. Finally, working within an ER context, I provide evidence of the educational impact of a scenario developed with the framework. Through several runs of the scenario with varying groups of participants, I show – through analysis of test results, questionnaires, and debriefing conversations – that students learned about teamwork, interdisciplinary communication, and patient assessment through participating in the scenario. Moreover, the virtual world setting provided students with a safe, low-pressure environment in which to practice clinical skills, and learn about the patient assessment process.

Bibliography

[1] R. Agrawal and R. Srikant. Fast algorithms for mining association rules. In *Proceedings of the 20th International Conference on Very Large Data Bases*, pages 487–499, 1994.

[2] Lois Ryan Allen. Measuring attitude toward computer assisted instruction: The development of a semantic differential tool. *Computers in Nursing*, 4(4):144–151, 1986.

[3] Toshitaka Amaoka, Hamid Laga, and Masayki Nakajima. Modeling the personal space of virtual agents for behavior simulation. In *International Conference on Cyberworlds*, 2009.

[4] John Seely Brown, Allan Collins, and Paul Duguid. Situated cognition and the culture of learning. *Educational Researcher*, pages 1–11, 1989.

[5] Steven A. Brown and Hilmi A. Lahoud. An examination of innovative online lab technologies. In *Proceedings of the 6th Conference on Information Technology Education (SIGITE '05)*, pages 65–70, New York, NY, USA, 2005.

[6] Ann Buchanan. Ann Myers Medical Centre. Website, 2009.

[7] Martha Burkle and Kinshuk. Learning in virtual worlds: The challenges and opportunities. In *International Conference on Cyberworlds*, 2009.

[8] Hong Cai, Bo Sun, Patty Farh, and Meng Ye. Virtual learning services over 3D internet: Patterns and case studies. In *IEEE International Conference on Services Computing (SCC '08)*, volume 2, pages 213 – 219, 2008.

[9] Christopher N. Candlin. *The Construction of Professional Discourse*, chapter General Editor's Preface. London: Longman, 1997.

[10] Baek-Hwan Cho, Jeonghun Ku, Dong Pyo Jang, Saebyul Kim, Yong Hee Lee, In Young Kim, Jang Han Lee, and Sun I. Kim. The effect of virtual reality cognitive training for attention enhancement. *Cyberpsychology and Behavior*, 5(2):129–137, 2002.

[11] David Chodos, Eleni Stroulia, and Parisa Naeimi. An integrated framework for simulation-based training on video and in a virtual world. *Journal of Virtual Worlds Research*, 2(1):1–28, 2009.

[12] Matthew Church, Eric Graves, Jason Duncan, Adel Lari, Robin Miller, Neesha Desai, Richard Zhao, Mike Carbonaro, Jonathan Schaeffer, Nathan Sturtevant, and Duane Szafron. A demonstration of ScriptEase II. In *Proceedings of the Seventh Artificial Intelligence and Interactive Digital Entertainment Conference (AIIDE)*, pages 215–216, October 2011.

[13] Teresa Coffman and Mary Beth Klinger. Utilizing virtual worlds in education: The implications for practice. *International Journal of Social Sciences*, 2(1):29–33, 2007.

[14] Allan Collins. *Educational Values and Cognitive Instruction: Implications for Reform*, chapter Cognitive Apprenticeship and Instructional Technology, pages 121–138. North Central Regional Educational Laboratory, 1991.

[15] Linda Creanor, Kathryn Trinder, Doug Gowan, and Carol Howells. LEX: The learner experience of e-learning. Glasgow Caledonian University, August 2006.

[16] Sara de Freitas. Serious virtual worlds: A scoping study. Serious Games Institute, JISC e-Learning Programme, November 2008.

[17] Sara de Freitas and Tim Neumann. The use of exploratory learning for supporting immersive learning in virtual environments. *Computers and Education*, 52:343–352, 2009.

[18] Andrea DeLucia, Rita Francese, Ignazio Passero, and Genoveffa Tortora. Development and evaluation of a virtual campus on Second Life: The case of SecondDMI. *Computers and Education*, 52:220–233, 2009.

[19] Parvati Dev, Patricia Youngblood, W. LeRoy Heinrichs, and Laura Kusumoto. Virtual worlds and team training. *Anesthesiology Clinics*, 25:321–336, 2007.

[20] John Dewey. *Democracy and Education*. New York: The Free Press, 1916.

[21] Michele M. Dickey. Three-dimensional virtual worlds and distance learning: two case studies of Active Worlds as a medium for distance education. *British Journal of Educational Technology*, 36(3):439–451, 2005.

[22] Don M. Dini, Michael van Lent, Paul Carpenter, and Kumar Iyer. Building robust planning and execution systems for virtual worlds. In *AIIDE*, pages 29–35, 2006.

[23] Thomas M. Duffy and David H. Jonassen. *Constructivism and the Technology of Instruction*, chapter Constructivism: New Implications for Instructional Technology. 1992.

[24] Heidi L. Frankel, Paul L. Rogers, Rajesh R. Gandhi, Eugene B. Freid, Orlando C. Kirton, and Michael J. Murray. What is taught, what is tested: Findings and competency-based recommendations of the UMEC of the SCCM. *Critical Care Medicine*, 32(9):1949–1956, 2004.

[25] Kristoffer Getchell, Alan Miller, J. Ross Nicoll, Rebecca J. Sweetman, and Colin Allison. Games methodologies and immersive environments for virtual fieldwork. *IEEE Transactions on Learning Technologies*, 3(4):281–293, October - December 2010.

[26] Carina Girvan and Timothy Savage. Identifying an appropriate pedagogy for virtual worlds: A communal constructivism case study. *Computers and Education*, 55:342–349, 2010.

[27] Gernot Goebbels and Vali Lalioti. Co-presence and co-working in distributed collaborative virtual environments. In *Proceedings of the 1st International Conference on Computer Graphics, Virtual Reality and Visualisation (AFRIGRAPH '01)*, pages 109–114, New York, NY, USA, 2001. ACM.

[28] Erika A. Goldstein, Carol F. MacLaren, Sherilyn Smith, Terry J. Mengert, Ramoncita R. Maestas, Hugh M. Foy, Marjorie D. Wenrich, and Paul G. Ramsey. Promoting fundamental clinical skills: A competency-based college approach at the U of washington. *Academic Medicine*, 80(5):423–433, 2005.

[29] R.S. Grabinger and J.C. Dunlap. *The changing face of learning technology*, chapter Rich environments for active learning: a definition, pages 8–38. University of Wales Press, 2000.

[30] Shawn Graham. Second lives: online worlds for archaeological teaching and research. *European Journal of Archaeology*, 10:77–79, 2007.

[31] Laura S Greci, Samantha Hurst, Karen Garman, Ricky Huang, Helene Hoffman, Micha Cardenas, Michael Gates, Kristen Kho, Todd Porteous, Erin Higginbotham, and Zia Agha. Multi-user virtual environment boot camp and patient surge triage practice. Presentation given at Medicine Meets Virtual Reality 18, Newport Beach, CA, February 8-12, 2011., February 2011.

[32] Laura S Greci, Rameshsharma Ramloll, Samantha Hurst, Karen Garman, Jaishree Beedasy, Eric B Pieper, Ricky Huang, Helene Hoffman, Micha Cardenas, Michael Gates, Kristen Kho, Todd Porteous, and Zia Agha. Pandemic flu patient surge planning and practice in a virtual environment. In *Innovations in Emergency Medicine Education, Society for Academic Emergency Medicine Annual Meeting Abstracts*, 2010.

[33] Chris Greenhalgh, Martin Flintham, Jim Purbrick, and Steve Benford. Applications of temporal links: Recording and replaying virtual environments. In *IEEE Virtual Reality Conference*, 2002.

[34] Chris Greenhalgh, Jim Purbrick, Steve Benford, Mike Craven, Adam Drozd, and Ian Taylor. Temporal links: recording and replaying virtual environments. In *Proceedings of the eighth ACM international conference on Multimedia*, pages 67–74, 2000.

[35] Fátima Gutiérrez, Jennifer Pierce, Victor M. Vergara, Robert Coulter, Linda Saland, Thomas P. Caudell, Timothy .E. Goldsmith, and Dale C. Alverson. The effect of degree of immersion upon learning performance in virtual reality simulations for medical education. *Studies in Health Technology and Informatics*, 125:155–160, 2007.

[36] Edward T. Hall. A system for the notation of proxemic behaviour. *American Anthropologist*, 65(5):1003–1026, 1965.

[37] Raija Hämäläinen, Sanna Järvelä, and Tony Manninen. Computer-supported collaboration in a scripted 3-D game environment. In *Computer Supported Collaborative Learning*, pages 504–508, 2005.

[38] William LeRoy Heinrichs, Patricia Youngblood, Phillip M. Harter, and Parvati Dev. Simulation for team training and assessment: Case studies of online training with virtual worlds. *World Journal of Surgery*, 32:161–170, 2008.

[39] M. Hobbs, E. Brown, and M. Gordon. Using a virtual world for transferable skills in gaming education. *ITALICS, The Higher Education Academy*, pages 1–13, 2006.

[40] Hsui-Mei Huang, Ulrich Rauch, and Shu-Sheng Liaw. Investigating learners' attitudes toward virtual reality learning environments: Based on a constructivist approach. *Computers and Education*, 55:1171–1182, 2010.

[41] Ken Hudson and Kathryn Degast-Kennedy. Canadian border simulation at Loyalist College. *Journal of Virtual Worlds Research*, 2(1):1–11, 2009.

[42] Samantha Hurst. Towards high-resolution ethnography for evaluation of team-oriented virtual reality training for medicine. In *Medicine Meets Virtual Reality 18*, 2011. Presentation.

[43] Leslie Jarmon. An ecology of embodied interaction: Pedagogy and homo virtualis. *Journal of Virtual Worlds Research*, 2:1–9, 2009.

[44] Sanna Järvelä. The cognitive apprenticeship model in a technologically rich learning environment. *Learning and Instruction*, 5:237–259, 1995.

[45] David W. Johnson and Frank P. Johnson. *Joining Together: Group Theory and Group Skills*. Boston: Allyn & Bacon, 1991.

[46] Blagica Jovanova, Marius Preda, and Françoise Preteux. The role of interoperability in virtual worlds. *Journal of Virtual Worlds Research*, 2(3):1–19, October 2009.

[47] Iris A. Junglas and Douglas J. Steel. The virtual sandbox. *SIGMIS Database*, 38(4):26–28, 2007.

[48] Bill Kapralos, Sayra Cristancho, Mark Porte, David Backstein, Alex Monclou, and Adam Dubrowski. Serious games in the classroom: Gauging student perceptions. In *Medicine Meets Virtual Reality 18*, pages 254–260, 2011.

[49] Ton Koenraad. How can 3D virtual worlds contribute to language education? focus on the language village format. Unpublished manuscript: Utrecht University of Applied Sciences, 2008.

[50] David Allen Kolb and Ronald Eugene Fry. *Toward and Applied Theory of Experiential Learning*. New York: John Wiley & Sons, 1975.

[51] Frederick W. Kron, Craig L. Gjerde, Ananda Sen, and Michael D. Fetters. Medical student attitudes toward video games and related new media technologies in medical education. *BMC Medical Education*, 10(50):1–11, 2010.

[52] Susan M. Land and Michael J. Hannafin. *Theoretical Foundations of Learning Environments*, chapter Student-Centered Learning Environments, pages 1–23. Lawrence Erlbaum Associates, 2000.

[53] Stefan Marks, John Windsor, and Burkhard Wünsche. Evaluation of game engines for simulated surgical training. In *GRAPHITE*, pages 273–280, 2007.

[54] Jean-Charles Marty and Thibault Carron. Observation of collaborative activities in a game-based learning platform. *IEEE Transactions on Learning Technologies*, 4(1):98–110, January - March 2011.

[55] Claudia L. McDonald. The Pulse!! collaboration: Academe & industry, building trust. In *Medicine Meets Virtual Reality 17*, 2009. Presentation.

[56] David McDonald, Alice Leung, William Ferguson, and Talib Hussain. An abstraction framework for cooperation among agents and people in a virtual world. In *AIIDE*, pages 54–59, 2006.

[57] M. McNaughton, M. Cutumisu, D. Szafron, J. Schaeffer, J. Redford, and D. Parker. ScriptEase: Generative design patterns for computer role-playing games. Technical Report 05, University of Alberta, 2004.

[58] Albert Mehrabian. *Nonverbal Communication*. Chicago: Aldine-Atherton, 1972.

[59] Paul R. Messinger, Xin Ge, Glen Mayhew, Run Niu, and Eleni Stroulia. *Handbook of Research on VEs for Corporate Education*, chapter Facilitating a Hierarchy of Education: Corporate Education in Virtual Worlds, pages 1–24. 2010.

[60] Tassos A. Mikropoulos and Antonis Natsis. Educational virtual environments: A ten-year review of empirical research (1999-2009). *Computers and Education*, 56:769–780, 2011.

[61] Teresa Monahan, Gavin McArdle, and Michela Bertolotto. Virtual reality for collaborative e-learning. *Computers and Education*, 50:1339–1353, 2008.

[62] Hideyuki Nakanishi and Toru Ishida. FreeWalk/Q: Social interaction platform in virtual space. In *Proceedings of the ACM Symposium on Virtual Reality Software andTechnology (VRST '04)*, pages 97–104, New York, NY, USA, 2004. ACM.

[63] Jeffrey C. Neely, K. Westmoreland Bowers, and Mathew W. Ragas. Virtual possibilities: A constructivist examination of the educational applications of second life. *Journal of Interactive Learning Research*, 21(1):93–110, 2010.

[64] B.C. Nelson and D.J. Ketelhut. Exploring embedded guidance and self-efficacy in educational and multi-user virtual environments. *Computer-Supported Collaborative Work*, 3:413–427, 2008.

[65] Diana G. Oblinger and James L. Oblinger. Educating the net generation. *EduCause*, 2005.

[66] Alexandra Petrakou. Interacting through avatars: Virtual worlds as a context for online education. *Computers and Education*, 54:1020–1027, 2010.

[67] Mark Prensky. Digital natives, digital immigrants. *On the Horizon*, 9(5):1–6, 2001.

[68] Colin Price. *Games-based Learning: Advancements for Multi-sensory Human Computer Systems*, chapter The Path between Pedagogy and Technology: Establishing a Theoretical Basis for the Development of Educational Game Environments, pages 191–213. 2009.

[69] Amith Tudur Raghavendra. Event-based data collection engine for serious games. In *Proceedings of the 9th International Conference on Entertainment Computing*, ICEC'10, pages 294–301, Berlin, Heidelberg, 2010. Springer-Verlag.

[70] M. Scardamalia and C. Bereiter. Computer support for knowledge-building communities. *The Journal of Learning Sciences*, 3(3):265–283, 1994.

[71] Roger C. Schank and Robert P. Abelson. *Scripts, Plans, Goals and Understanding; An inquiry into human knowledge structures*. Lawrence Erlbaum, 1977.

[72] Ralph Schroeder. Defining virtual worlds and virtual environments. *Journal of Virtual Worlds Research*, 1(1):1–3, 2008.

[73] Laurel M. Sheppard. Virtual surgery brings back smiles. *IEEE Computer Graphics and Applications*, 25:6–11, 2005.

[74] Gary Small and Gigi Vorgan. *iBrain: Surviving the Technological Alteration of the Modern Mind*. HarperColins, 2008.

[75] Kristen Smirnov. Feedback on MeRiTS log analysis tools. Personal conversation.

[76] Neha Sugandh, Santiago Ontanon, and Ashwin Ram. On-line case-based plan adaptation for real-time strategy games. In *AAAI Conference on Artificial Intelligence*, 2008.

[77] D. Taylor, V. Patel, D. Cohen, R. Aggarwal, K Kerr, N. Sevdalis, N. Batrick, and A. Darzi. Single and multi-user virtual patient design in the virtual world. *Studies in Health Technology and Informatics*, 163:650–652, 2011.

[78] Olle ten Cate and Fedde Scheele. Competency-based postgraduate training: Can we bridge the gap between theory and clinical practice? *Academic Medicine*, 82(6):542–547, 2007.

[79] Ruck Thawonmas and Keita Iizuka. Visualization of online-game players based on their action behaviors. *International Journal of Computer Games Technology*, 2008:1–9, 2008.

[80] Julita Vassileva. Toward social learning environments. *IEEE Transactions on Learning Technologies*, 1(4):199–214, 2008.

[81] Victor Vergara, Thomas Caudell, Timothy Goldsmith, Panaiotis, and Dale Alverson. Knowledge-driven design of virtual patient simulations. *Innovate: Journal of Online Education*, 5(2):1–6, 2008.

[82] Gustav Verhulsdonck and Jacquelyn Ford Morie. Virtual chironomia: Developing non-verbal communication standards in virtual worlds. *Journal of Virtual Worlds Research*, 2(3):1–10, 2009.

[83] Richard A. Voorhees. Competency-based learning models: A necessary future. *New Directions for Institutional Research*, 110:5–13, 2001.

[84] Natalia Voynarovskaya, Roman Gorbunov, Emilia Barakova, Rene Ahn, and Matthias Rauterberg. Nonverbal behavior observation: collaborative gaming method for prediction of conflicts during long-term missions. In *Proceedings of the 9th International Conference on Entertainment Computing*, ICEC'10, pages 103–114, Berlin, Heidelberg, 2010. Springer-Verlag.

[85] Victoria L. Walker. 3D virtual learning in counselor education: Using Second Life in counselor skill development. *Journal of Virtual Worlds Research*, 2(1):1–14, April 2009.

[86] David Waller, David Knapp, and Earl Hunt. Spatial representations of virtual mazes: The role of visual fidelity and individual differences. *Human Factors*, 43(1):147–158, 2001.

[87] Stephen West, Ross A. Brown, and Jan C. Recker. Collaborative business process modeling using 3D virtual environments. In *16th Americas Conference on Information Systems : Sustainable IT Collaboration around the Globe*, 2010.

[88] W. Winn and M. Windschitl. Learning science in virtual environments: the interplay of theory and experience. *Themes in Education*, 1:373–389, 2000.

[89] Maja Wrzesien and Mariano Alcañiz Raya. Learning in serious virtual worlds: Evaluation of learning effectiveness and appeal to students in the E-Junior project. *Computers and Education*, 55:178–187, 2010.

[90] WS-BPEL. Web services business process execution language technical committee., 2009.

Appendix A

Scenario Definition – EMT Handoff

A.1 Settings

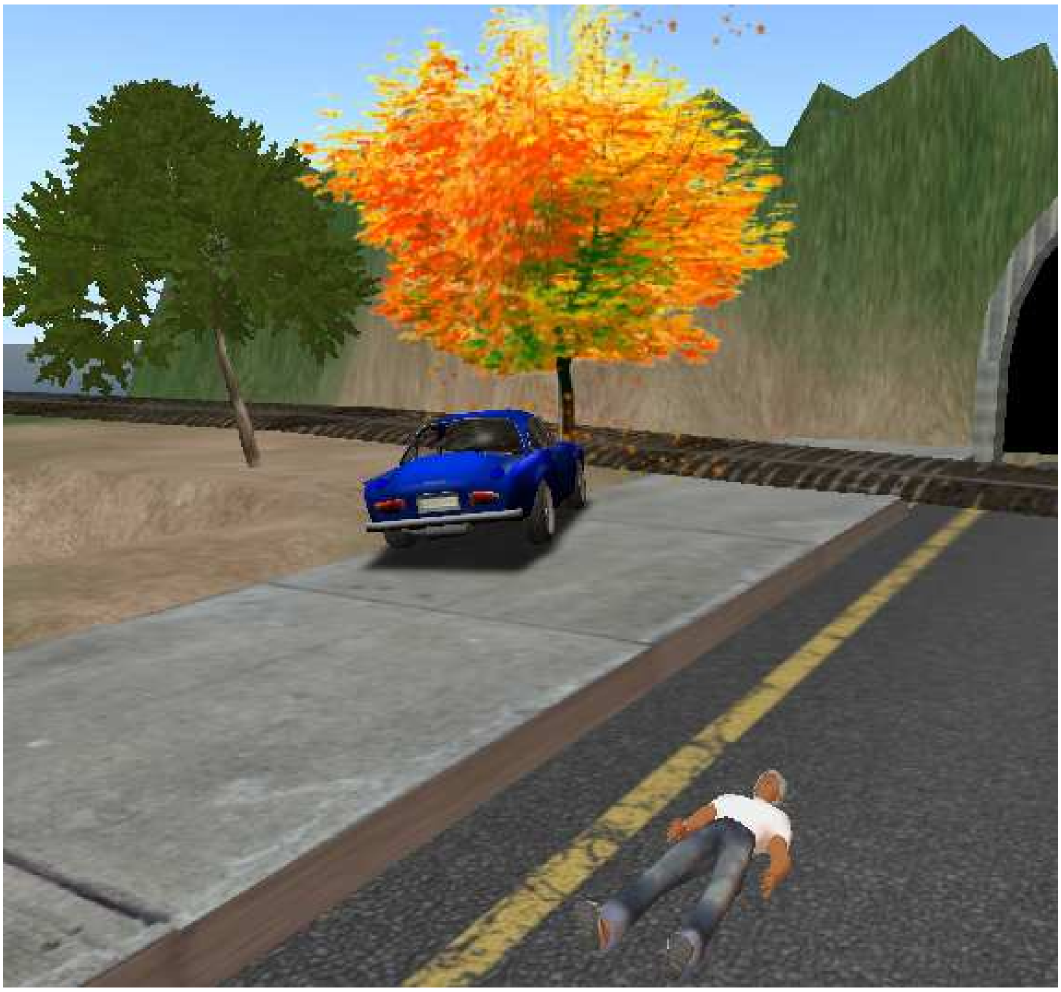


Figure A.1: Accident Scene

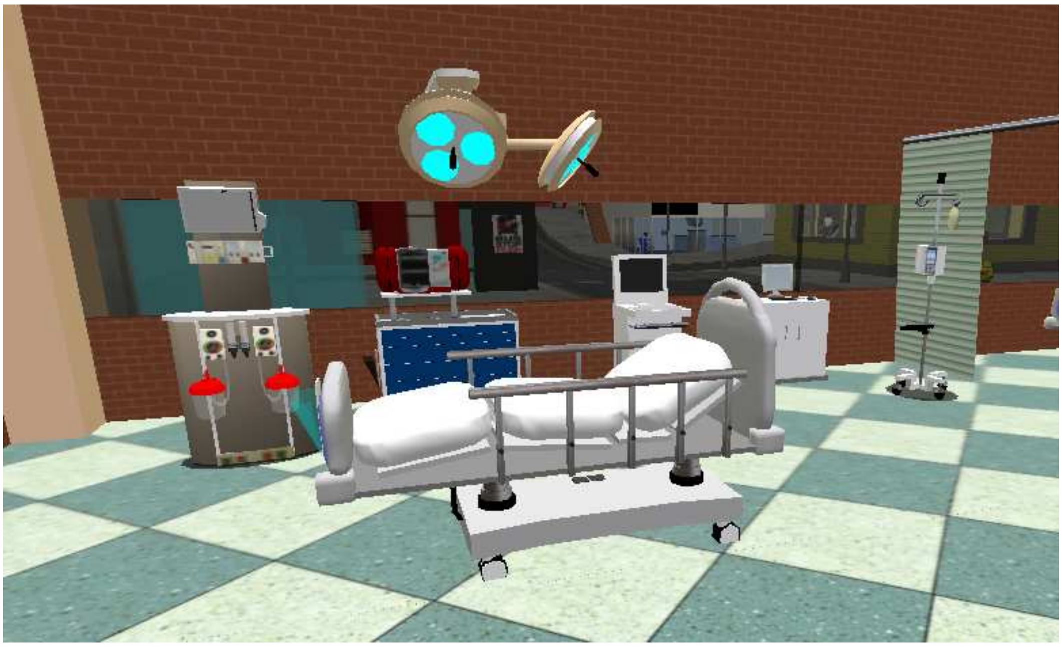


Figure A.2: Emergency Room

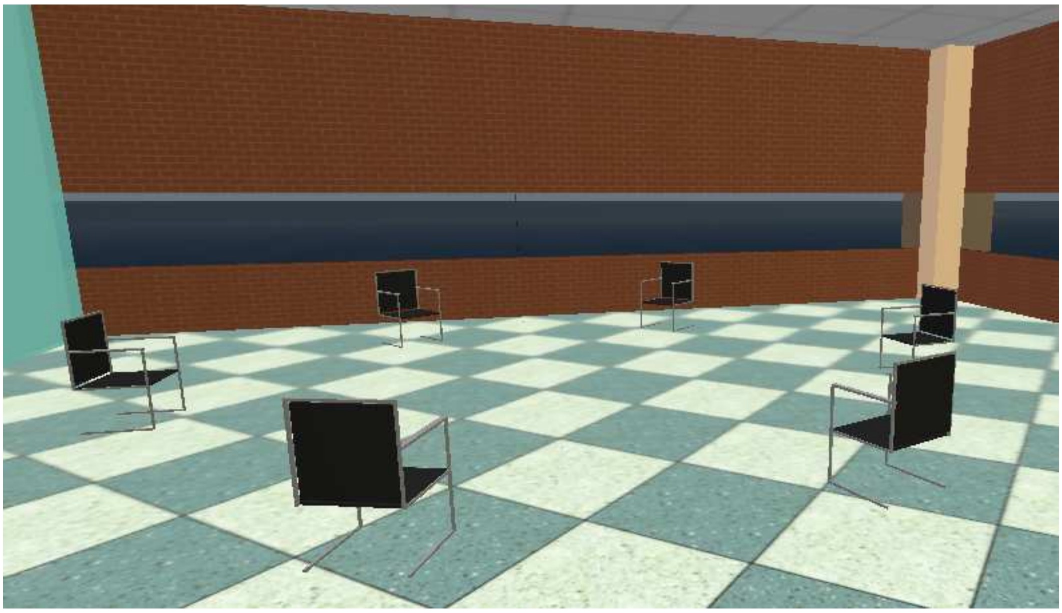


Figure A.3: Debriefing Area

A.2 Variables

Table A.1: Variables

Name	Type	Default Value
Systolic BP	integer	100
Pulse Checked	boolean	false
Called Hospital	boolean	false
On Spineboard	boolean	false
Goggles	string	off
Diastolic BP	integer	60
Pulse	integer	110
Pulse Description	string	weak
Saline Given	boolean	false
Spineboard Location	string	on gurney
Transport Status	string	on ground
Blood Glucose	real	2.1
Blood Glucose Checked	boolean	false
Dextrose Given	boolean	false
Ambulance Location	string	fire hall
Gurney Location	string	fire hall
Gurney Status	string	in ambulance

A.3 Objects

Table A.2: Object Actions for the EMT handoff scenario

Action	Description	Constraint
Ambulance		
Drive to accident scene	Drive from the starting location to the accident scene	None
Drive to hospital	Drive from the accident scene to the hospital	Victim must be in ambulance
Gurney		
Push to accident scene	Push from the ambulance to the accident scene	None
Push to ambulance	Push from the accident scene to the ambulance	Victim must be on the gurney
Spineboard		
Move to victim	Move the spineboard to the ground beside the victim	Spineboard must be on gurney
Radio		
Call hospital	Call ER staff in the hospital	None
End call	End the call to the ER staff	The call must have been started

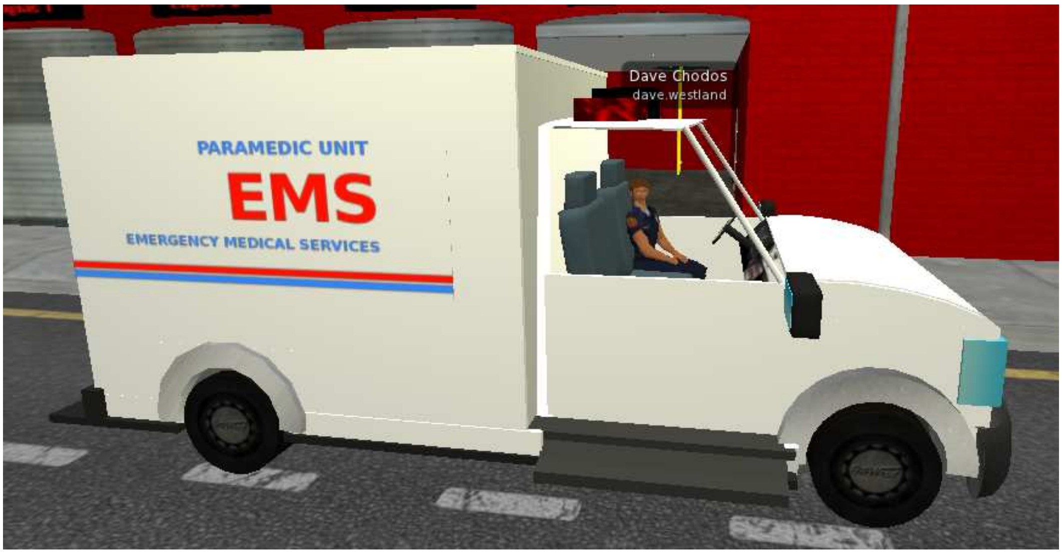


Figure A.4: Ambulance

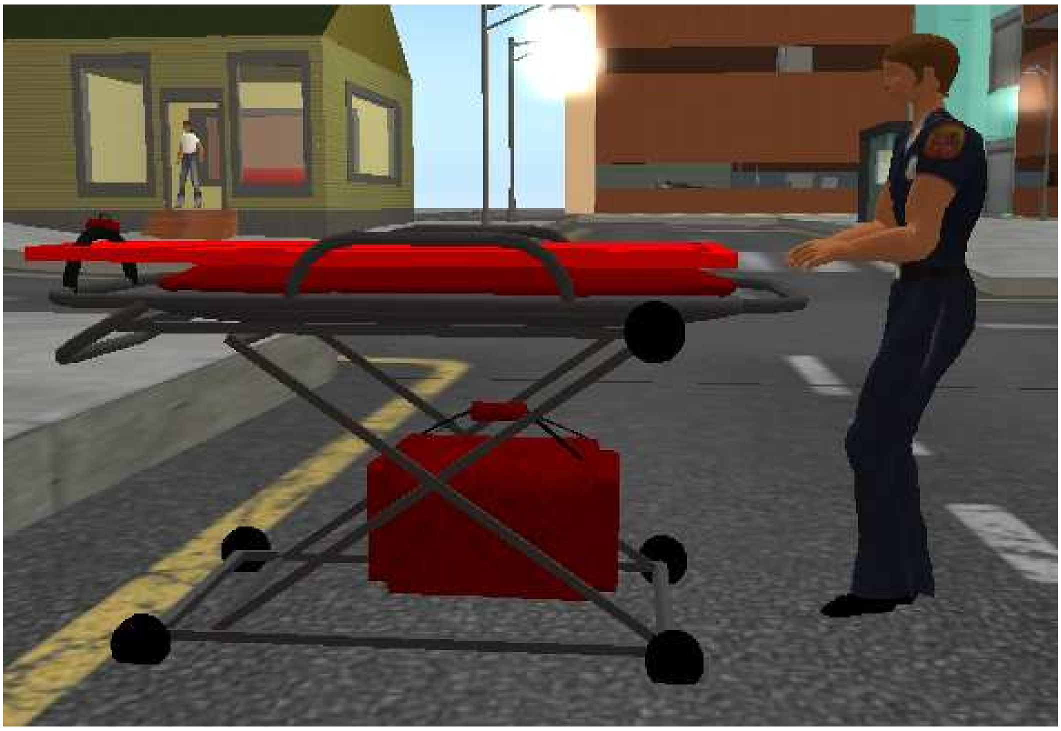


Figure A.5: Gurney and Spineboard

Figure A.6: Radio

Figure A.7: Victim

A.4 Workflows and Constraints

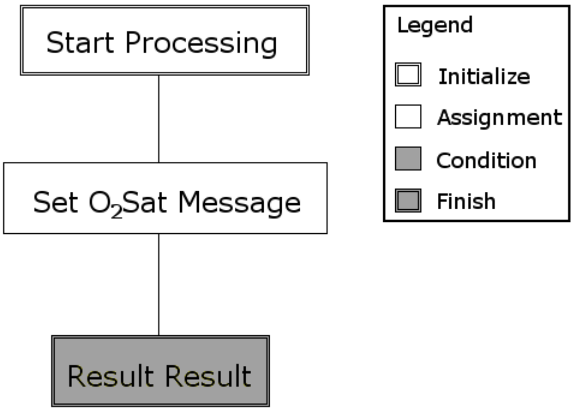


Figure A.8: Pulse Oximeter Workflow

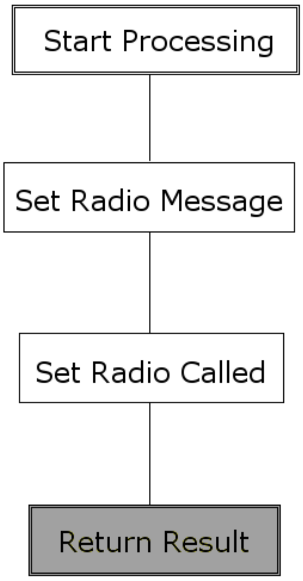


Figure A.9: Radio Workflow

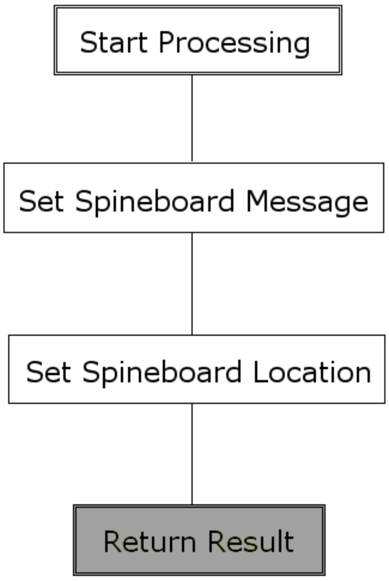


Figure A.10: Spineboard Workflow

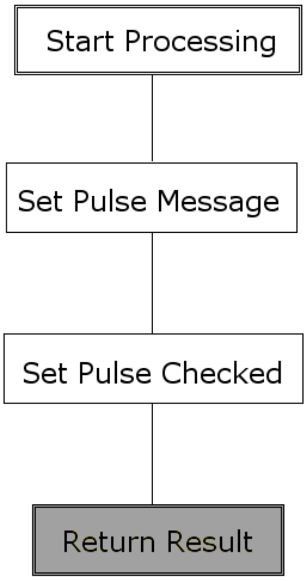


Figure A.11: Stethoscope Workflow

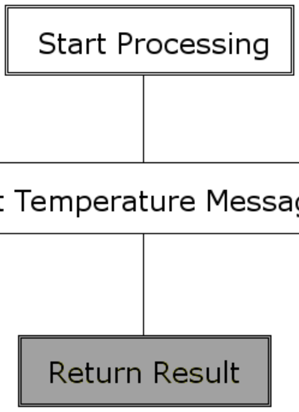


Figure A.12: Thermometer Workflow

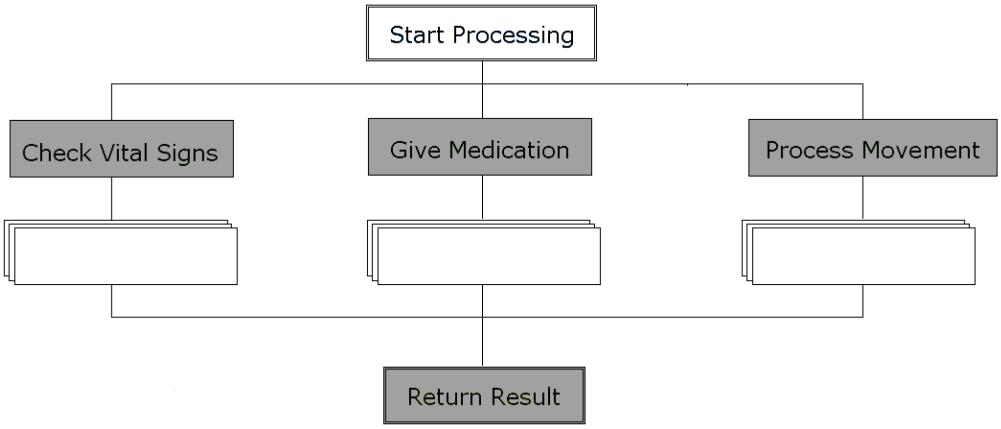


Figure A.13: Victim Workflow

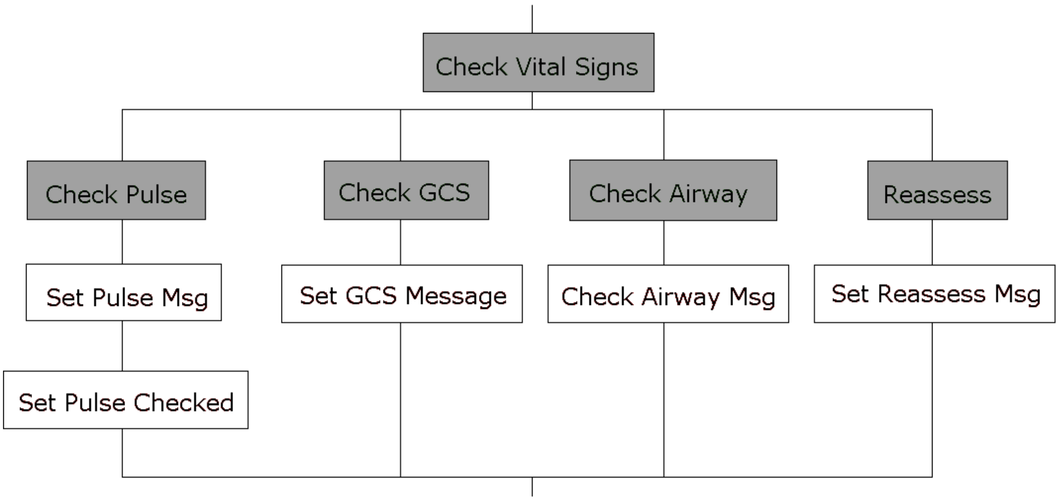


Figure A.14: Victim Workflow - Vital Signs

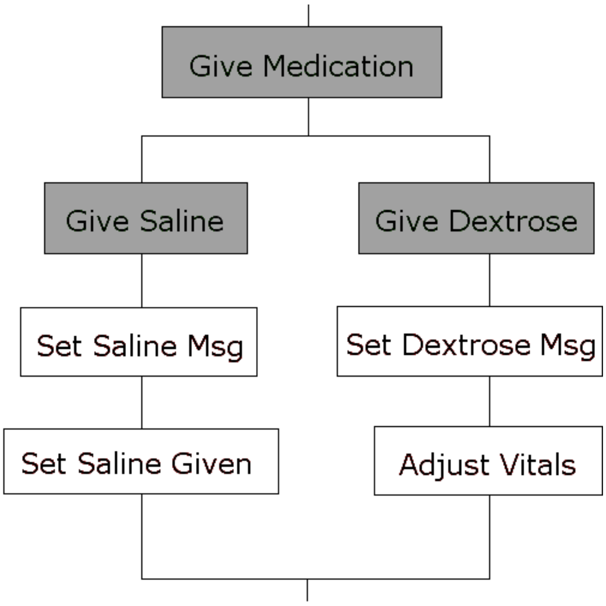


Figure A.15: Victim Workflow - Give Medication

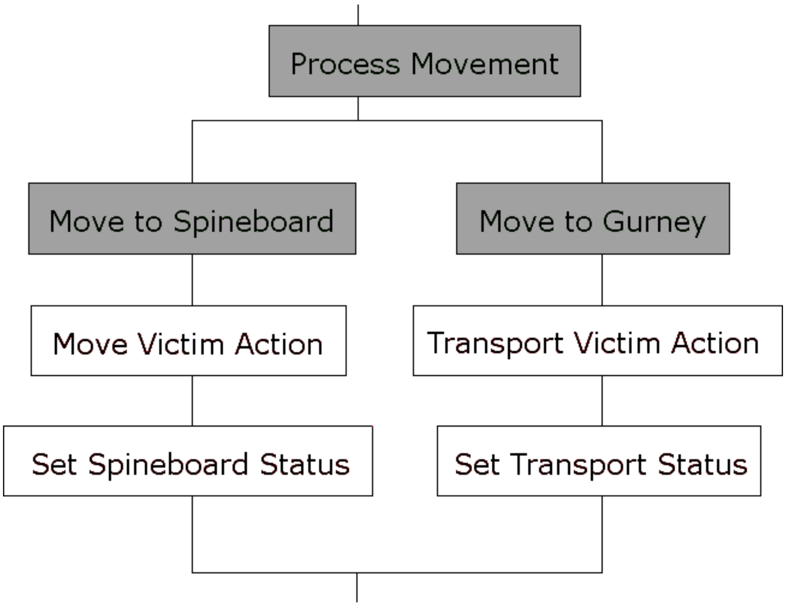


Figure A.16: Victim Workflow - Movement

Table A.3: Constraints for the EMT handoff scenario

Name	**Type**	**Scope**	**Constraint**
Victim vital signs			
Basic vital signs	Feasibility	Global	SystolicBP $\geq$ 65 and DiastolicBP $\geq$ 45
Vital signs and movement	Advisibility	Global	SystolicBP $\geq$ 75 and DiastolicBP $\geq$ 55
Other clinical constraints			
Should wear PPE during treatment	Advisibility	Local (Victim)	When moving victim, Goggles = on
Should check pulse before moving victim	Advisibility	Local (Victim)	When moving victim, CheckedPulse = yes
Must start IV before administering dextrose	Advisibility	Local (Victim)	When giving dextrose, SalineGiven = yes
Moving victim to spineboard and gurney			
Spineboard on ground	Feasibility	Local (Victim)	Before putting victim on spineboard, SpineboardLocation = onGround
Victim on spineboard	Feasibility	Local (Victim)	Before moving victim to gurney, OnSpineboard = yes
Victim already in transit	Feasibility	Local (Victim)	Before moving victim to gurney, OnSpineboard = yes
Actions to be taken once			
PPE can only be put on once	Feasibility	Local (PPE)	Before putting on goggles, Goggles = off
Hospital can only be called once	Feasibility	Local (Radio)	Before calling hospital, CalledHospital = no
Victim can only be moved to spineboard once	Feasibility	Local (Victim)	When moving victim to spineboard, OnSpineboard = no
Can only give victim saline once	Feasibility	Local (Victim)	Before giving victim saline, SalineGiven = no
Can only give victim dextrose once	Feasibility	Local (Victim)	Before giving victim dextrose, DextroseGiven = no

Appendix B

Scenario Definition – Marketing

B.1 Settings

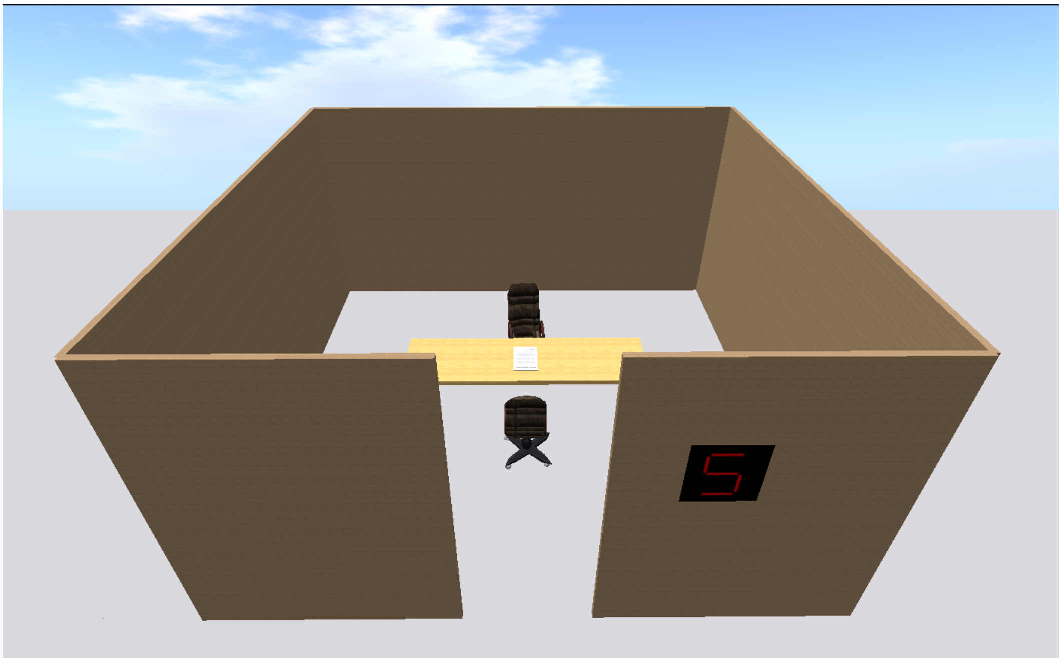


Figure B.1: Marketing Negotiation Cubicle

B.2 Variables

Table B.1: Variables

Name	Type	Default Value
Seller ID	integer	0
Buyer ID	integer	0
Proposed Price	real	0.00
Price Confirmed	boolean	false

B.3 Objects

Table B.2: Object Actions for the Marketing Experiment

Action	Description	Constraint
	Contract	
Enter price	Seller enters price	
Confirm price	Buyer confirms price	Buyer ID is different from seller ID
Reject price	Buyer rejects price	Buyer ID is different from seller ID

Rental Agreement

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Etiam et lorem ut dui dignissim consectetur. Sed fringilla consequat sodales. Sed neque nunc, consectetur vel ultrices eget, convallis et velit. Fusce nec massa vel odio varius varius. Suspendisse consequat blandit lobortis.

Cras ullamcorper porta nibh at adipiscing. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae; Suspendisse fermentum est ut felis feugiat et tempor leo mattis. Donec non condimentum odio.

Fusce fermentum tempor bibendum. Sed sollicitudin quam ut tortor convallis mattis eleifend felis gravida. Nunc sed turpis et lorem ultrices auctor. Pellentesque mi lacus, tempor ac varius ac, feugiat sed ipsum. Ut venenatis justo sed mi semper eget auctor metus scelerisque. Nunc semper nisl sit amet est rutrum sed blandit eros pharetra.

Fusce quis sagittis massa. Sed vulputate iaculis libero, a pharetra magna rhoncus auctor. Morbi orci massa, mollis quis ultrices vel, eleifend eget eros.

Landlord ____________________

Renter ____________________

Figure B.2: Contract

B.4 Workflows and Constraints

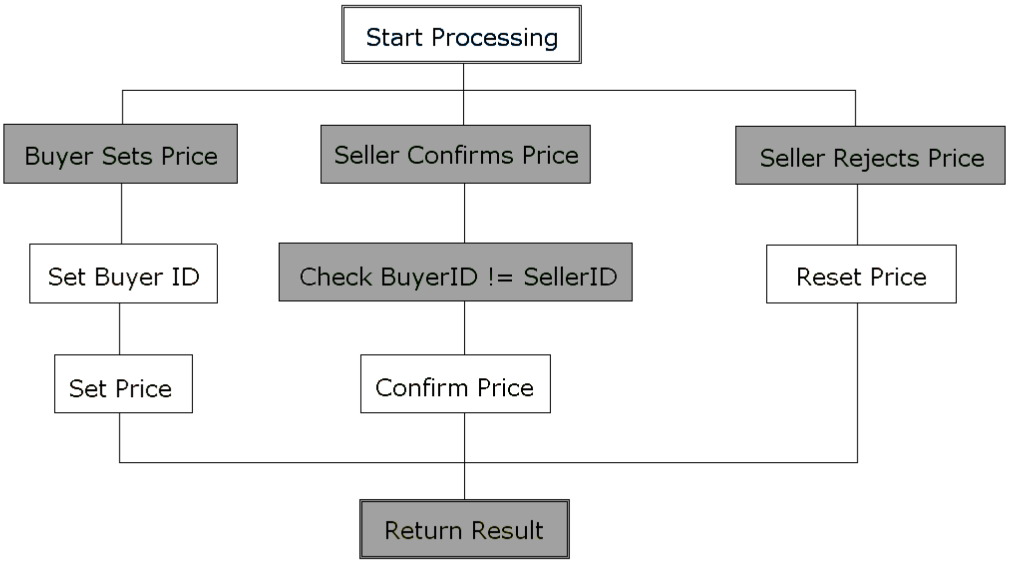


Figure B.3: Contract Workflow

Appendix C

Scenario Definition – ER Nursing

C.1 Settings

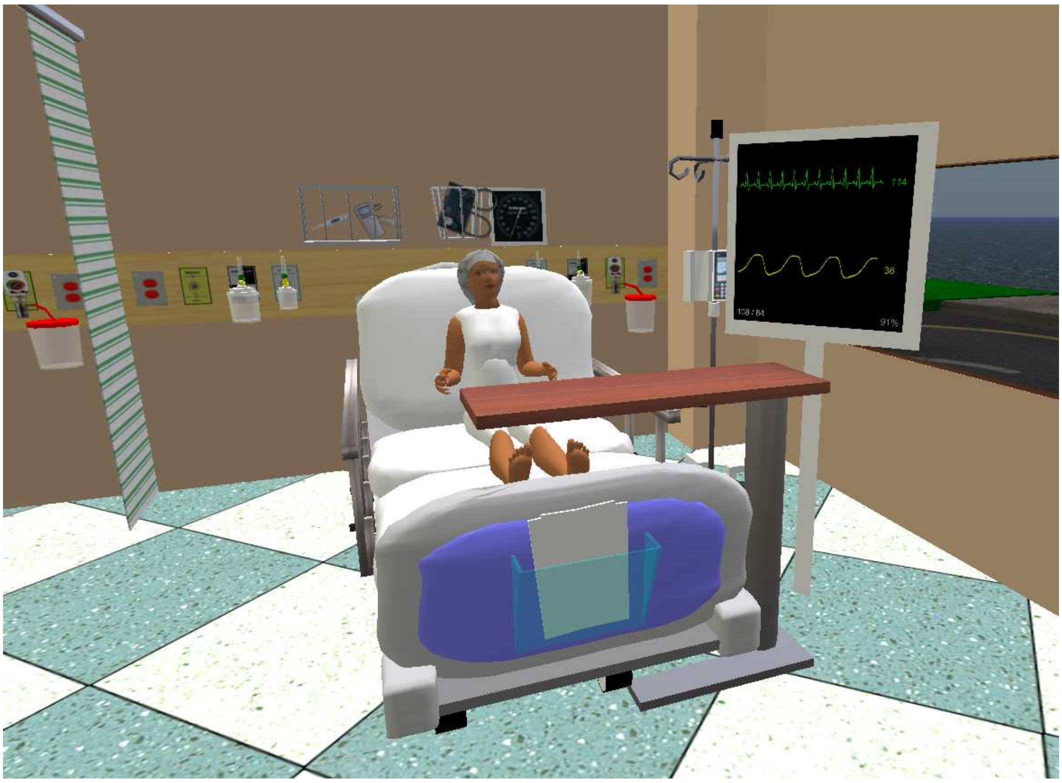


Figure C.1: Emergency Room

C.2 Variables

Table C.1: Variables

Name	**Type**	**Default Value**
SystolicBP	integer	108
DiastolicBP	integer	64
HeartRate	integer	114
RespiratoryRate	integer	36
OxygenSaturation	integer	91
Temperature	integer	37
CheckedChest	boolean	false
CheckedHead	boolean	false
NebulizerMedication	string	none
EvaluationMessage	string	*blank*
CriticalChecked	boolean	false
NonCriticalChecked	boolean	false
CheckedAbdomen	boolean	false
CheckedPelvis	boolean	false
CheckedExtremities	boolean	false
OxygenRate	integer	0
OxygenGiven	boolean	false
NumActions	integer	0
VitalsTrajectory	string	downward
MedicationGiven	boolean	false
VitalsStatus	string	downward
IVGiven	boolean	false

C.3 Objects

Table C.2: Object Actions for the ER Nursing Experiment

Action	**Description**	**Constraint**
	Patient	
Perform action	Perform an assessment or treatment action	Assessment is not already finished
Finish assessment	Complete the assessment	Assessment is not already finished
	Oxygen Pump	
Set O_2 rate	Set the rate of oxygen flow	Another pump is not already being used
Administer O_2	Administer oxygen to the patient	O_2 rate is set, is between 2 and 4 L/min
	Medical Air Pump	
Set medication	Set the medication to be administered	Another pump is not already being used
Administer medication	Administer medication to the patient	medication is set to either “ventolin” or “atrovent”

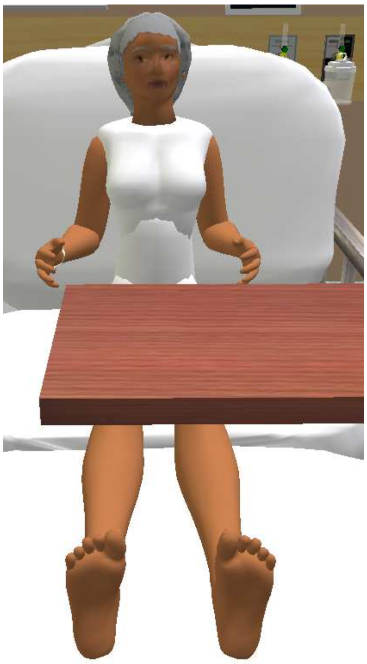


Figure C.2: ER Patient

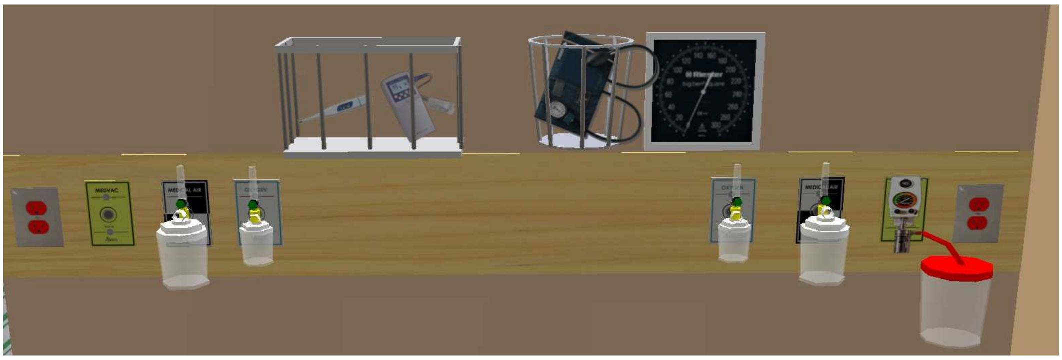


Figure C.3: Headwall

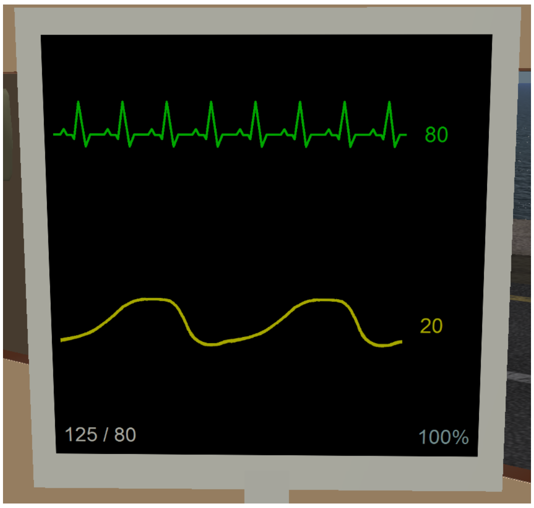


Figure C.4: Vital Signs Monitor

C.4 Workflows and Constraints

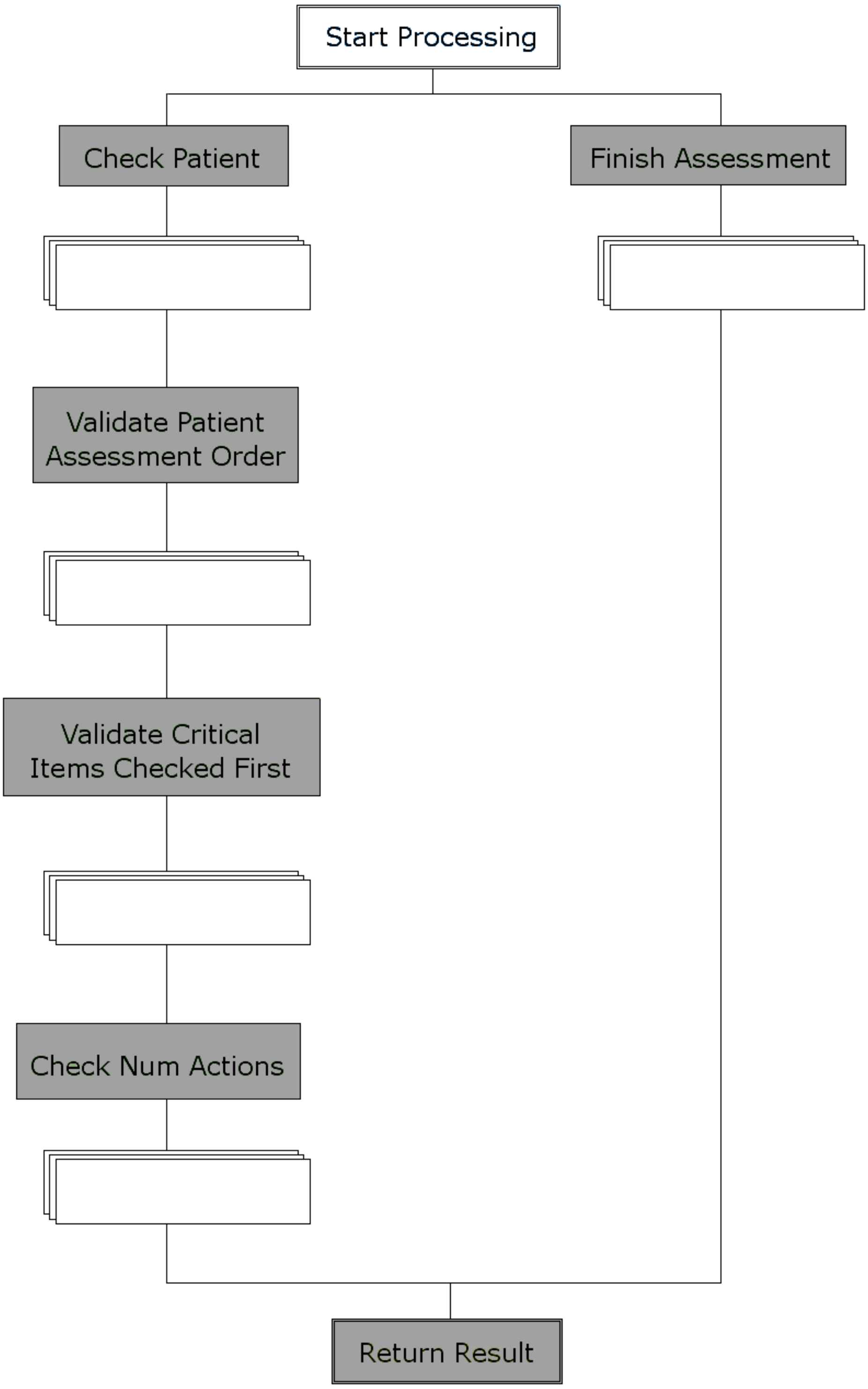


Figure C.5: Patient Workflow

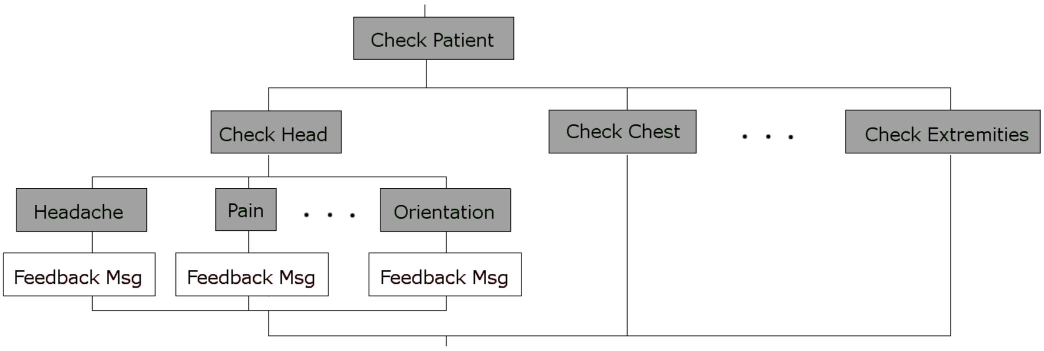


Figure C.6: Patient Workflow - Check Patient

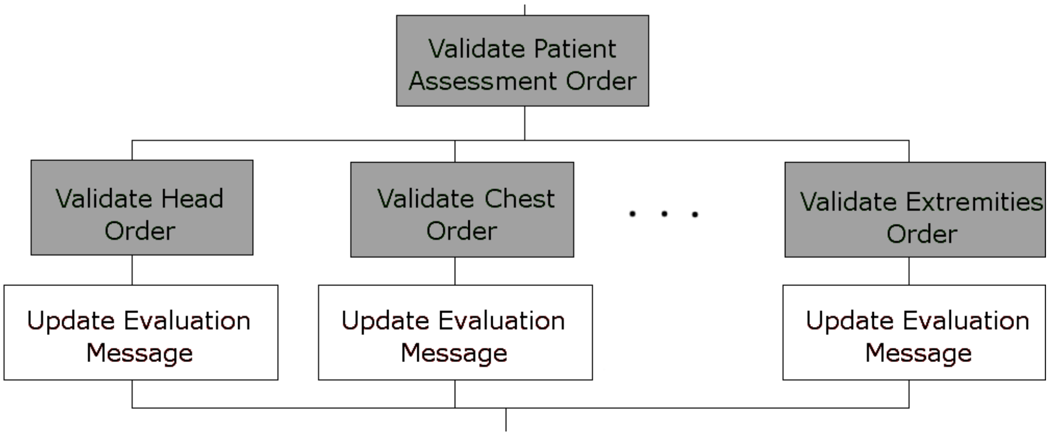

Validate Patient Assessment Order
Validate Head Order
Validate Chest Order
. . .
Validate Extremities Order
Update Evaluation Message
Update Evaluation Message
Update Evaluation Message

Figure C.7: Patient Workflow - Assessment Order

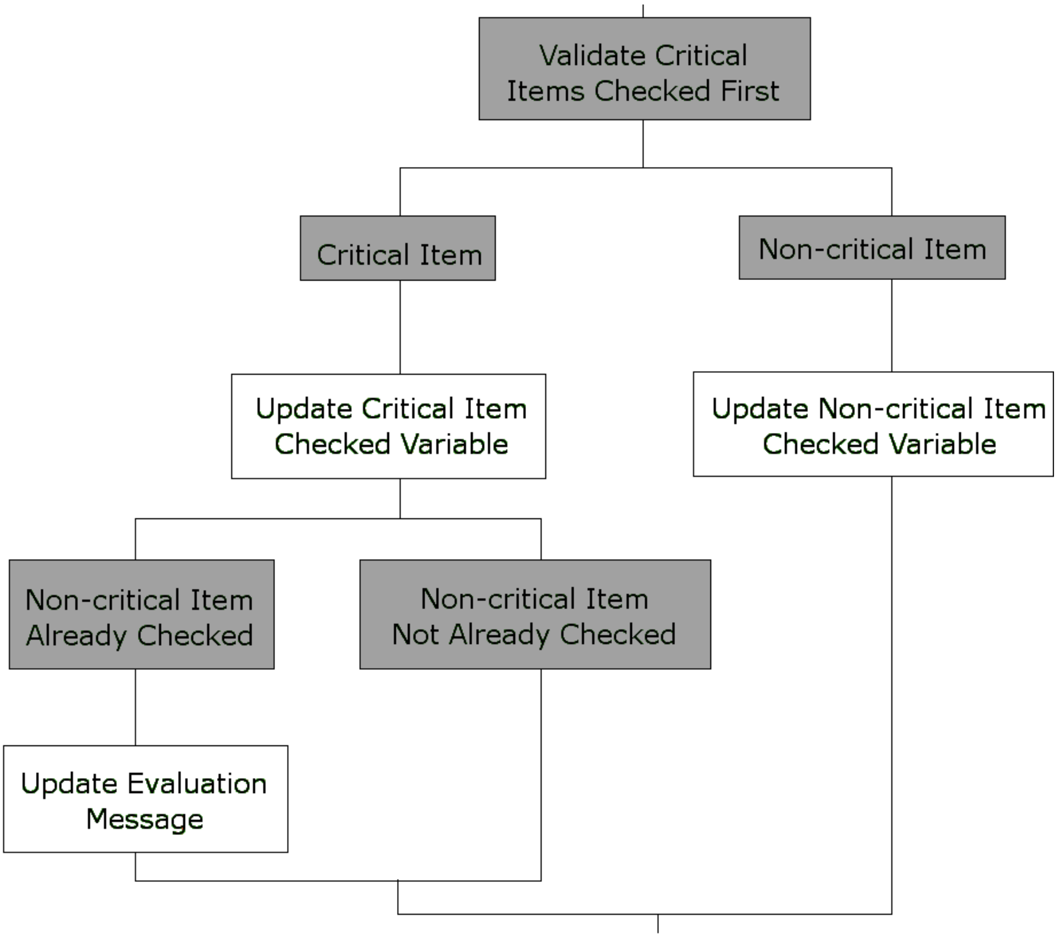

Validate Critical Items Checked First
Critical Item
Non-critical Item
Update Critical Item Checked Variable
Update Non-critical Item Checked Variable
Non-critical Item Already Checked
Non-critical Item Not Already Checked
Update Evaluation Message

Figure C.8: Patient Workflow - Critical Items Checked First

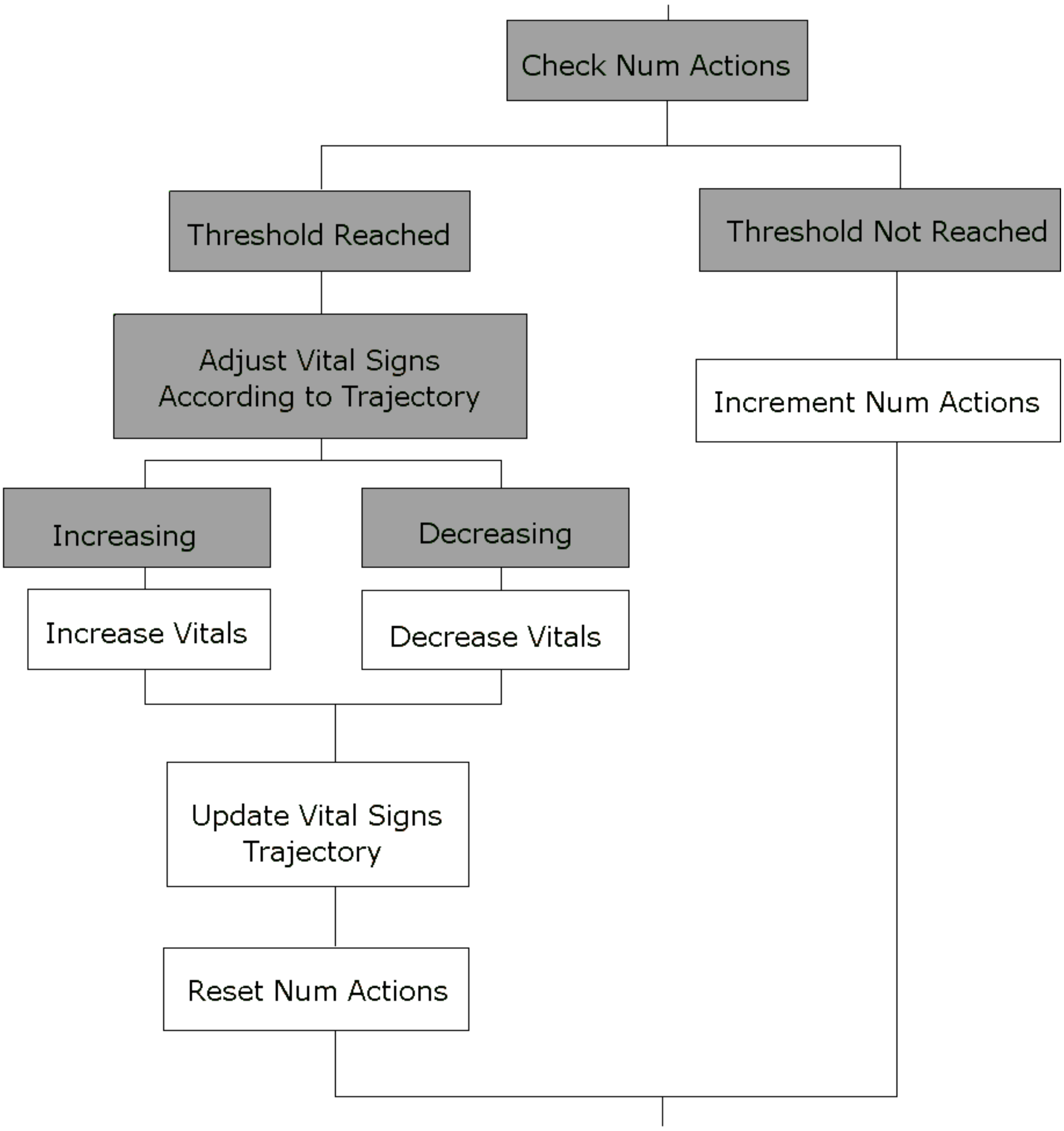


Figure C.9: Patient Workflow - Update Num Actions

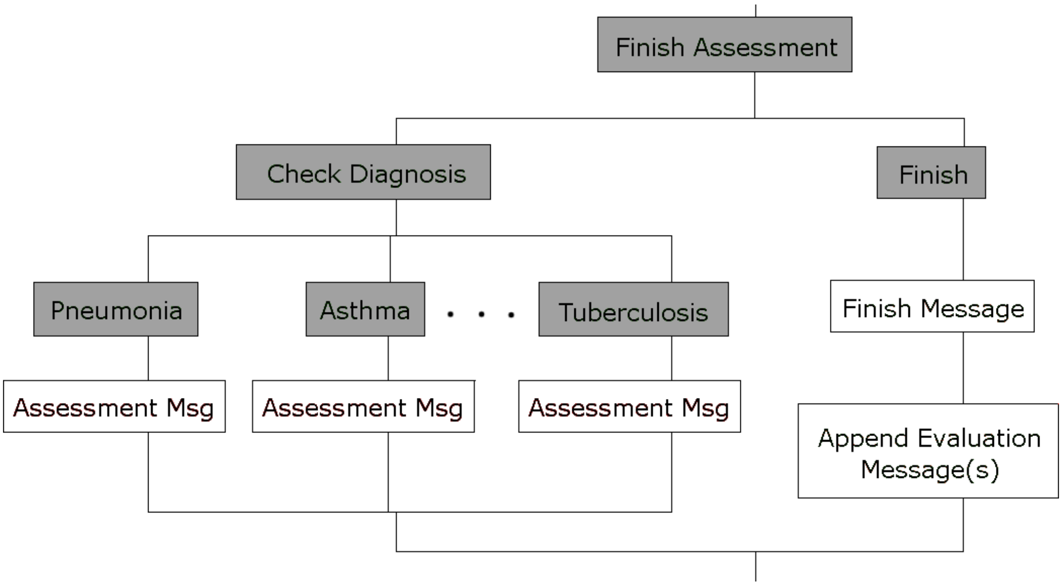


Figure C.10: Patient Workflow - Finish Assessment

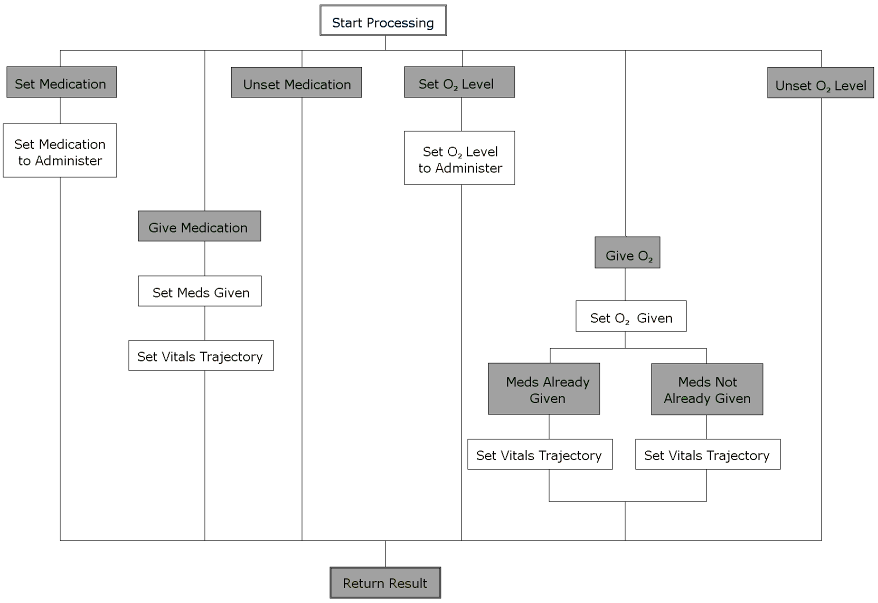


Figure C.11: Oxygen and Medical Air Pump Workflow

Appendix D

Dynamic Object Composition

This composite entity is then disassembled into its constituent parts upon arrival. More specifically, the mechanism involves m old parent entities $P_1, \cdots, P_m$, n child entities $C_{m,1}, \cdots, C_{m,n}$ for each old parent entity P_m, a new parent entity N (where $N \in P_1, \cdots, P_m$), and a relay entity R. These entities interact as follows, and as shown in Figure D.1:

1. P_i sends a "disassemble" message to R, for some $1 \leq i \leq m$
2. R sends a "start disassembly" (SD) message to each of $P_1, \cdots, P_m$
3. Upon receipt of SD, each P_i sets its status to "in transition", and sends a "confirm disassembly" (CD) message to R
4. Upon receipt of CD, from P_i, R sends a "detach child" (DC) message to $C_{i,j}$ for $1 \leq j \leq n$
5. Upon receipt of DC, $C_{i,j}$ detaches itself from its parent P_i
6. Once all of the children of P_i have been detached, P_i sends a "finished disassembly" (FD) message to R
7. Upon receipt FD from P_i, R sends a "start reassembly" (SR) message to N, with the expected number of children E
   Note that $E = P_1 + \cdots P_m + (C_{i,1} + \cdots + C_{i,n}$ for each $1 \leq i \leq m)$
8. Upon receipt of SR, N sets E and responds with a "confirm reassembly" (CR) to R
9. Upon receipt of CR, R sends an "attach to N" (AN) message to P_i and $C_{i,j}$
10. Upon receipt of AN, $C_{i,j}$ and P_i attach themselves to N
11. When N reaches the expected number of children E, it sends a "finished reassembly" (FR) message to R
12. Upon receipt of FR, R sends a "finished assembly" (FA) message to each of $P_1, \cdots, P_m$
13. Upon receipt of FA, each P_i sets its status to "finished"

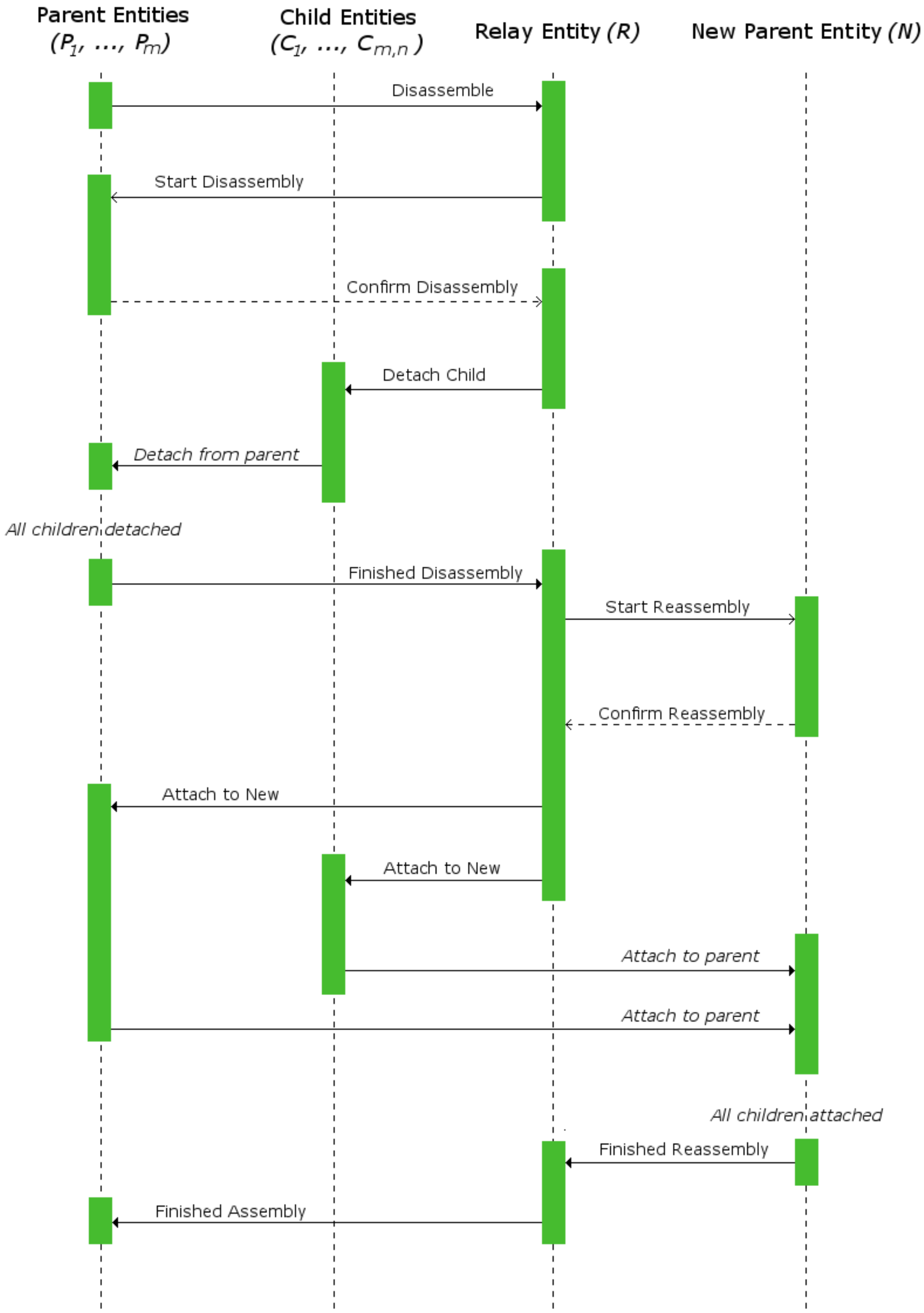


Figure D.1: Entity Disassembly and Re-assembly Sequence Diagram

Appendix E

Scenario Trace and Parsed Actions

E.1 Scenario Trace

```
 1.  movement 233.678,49.9461,27.5207 Dave Westland 2012-01-16 10:23:09 ERsim
 2.  movement 233.291,53.1195,27.5207 Dave Westland 2012-01-16 10:23:10 ERsim
 3.  movement 232.929,56.2452,27.8446 Dave Westland 2012-01-16 10:23:11 ERsim
 4.  movement 232.794,59.0332,27.5426 Dave Westland 2012-01-16 10:23:12 ERsim
 5.  movement 232.785,59.1530,27.5453 Dave Westland 2012-01-16 10:23:13 ERsim
 6.  interaction What would you like to do with the chest? respirations, ...
      cough, ..., pulse Dave Westland 2012-01-16 10:23:15 ERsim
 7.  interactionResponse cough The patient has a dry cough.  Dave Westland
     2012-01-16 10:23:18 ERsim
 8.  movement 232.571,61.3411,27.5477 Dave Westland 2012-01-16 10:23:23 ERsim
 9.  movement 232.657,62.7528,27.5207 Dave Westland 2012-01-16 10:23:24 ERsim
10.  hold PulseOximeter Dave Westland 2012-01-16 10:23:26 ERsim
11.  movement 232.489,61.2485,27.5655 Dave Westland 2012-01-16 10:23:27 ERsim
12.  movement 232.429,58.7718,27.5403 Dave Westland 2012-01-16 10:23:28 ERsim
13.  movement 232.429,58.7723,27.5409 Dave Westland 2012-01-16 10:23:29 ERsim
14.  movement 232.429,58.7724,27.5423 Dave Westland 2012-01-16 10:23:30 ERsim
15.  movement 232.521,58.9850,27.5493 Dave Westland 2012-01-16 10:23:31 ERsim
16.  movement 232.864,59.6740,27.5507 Dave Westland 2012-01-16 10:23:32 ERsim
17.  interaction Would you like to use the PulseOximeter, or assess the right
     arm?  Dave Westland 2012-01-16 10:23:33 ERsim
18.  interactionResponse PulseOximeter Oxygen saturation is 91% Dave Westland
     2012-01-16 10:23:37 ERsim
19.  movement 232.856,59.6460,27.5508 Dave Westland 2012-01-16 10:23:42 ERsim
20.  movement 232.784,61.1403,27.5507 Dave Westland 2012-01-16 10:23:43 ERsim
21.  movement 232.781,61.3374,27.5207 Dave Westland 2012-01-16 10:23:44 ERsim
22.  drop PulseOximeter Dave Westland 2012-01-16 10:23:44 ERsim
23.  movement 232.798,60.4471,27.6013 Dave Westland 2012-01-16 10:23:46 ERsim
24.  movement 232.837,58.7482,27.5507 Dave Westland 2012-01-16 10:23:47 ERsim
25.  textChat hello there Dave Westland 2012-01-16 10:23:50 ERsim
26.  movement 232.818,58.8056,27.5507 Dave Westland 2012-01-16 10:24:02 ERsim
27.  movement 232.874,58.8669,27.5507 Dave Westland 2012-01-16 10:24:03 ERsim
28.  movement 232.566,56.2748,27.5507 Dave Westland 2012-01-16 10:24:04 ERsim
29.  movement 232.014,53.2010,27.5207 Dave Westland 2012-01-16 10:24:05 ERsim
30.  movement 231.225,50.3208,27.5207 Dave Westland 2012-01-16 10:24:06 ERsim
31.  movement 230.914,49.1631,27.5207 Dave Westland 2012-01-16 10:24:07 ERsim
```

E.2 Parsed Actions

Movement: **Dave Westland** moved from **Start position** to **equipment closet** (2012-01-26 10:23:09) [Source: lines 1-2 of trace]

Movement: **Dave Westland** moved from **equipment closet** to **ER bed 3** (2012-01-26 10:23:11) [Source: lines 2-4]

Movement: **Dave Westland** moved from **ER bed 3** to **ER patient** (2012-01-26 10:23:12) [Source: lines 4-5]

Interaction: **Dave Westland** was shown **“What would you like to do with the chest?”** with options **“respirations,skin colour,patient,cough,lung sounds,pulse”** (2012-01-26 10:23:15) [Source: line 6]

Interaction: **Dave Westland** chose **“cough”** and was presented with the response **“The patient has a dry cough.”** (2012-01-26 10:23:18) [Source: line 7]

Movement: **Dave Westland** moved from **ER patient** to **headwall** (2012-01-26 10:23:23) [Source: lines 8-9]

Held Item: **Dave Westland** picked up a **PulseOximeter** (2012-01-26 10:23:26) [Source: line 10]

Movement: **Dave Westland** moved from **headwall** to **ER patient** (2012-01-26 10:23:27) [Source: lines 11-16]

Interaction: **Dave Westland** was shown **“Would you like to use the PulseOximeter, or assess the right arm?”** with options **“PulseOxim,RightArm”** (2012-01-26 10:23:33) [Source: line 17]

Interaction: **Dave Westland** chose **‘PulseOximeter”** and was presented with the response **‘Oxygen saturation is 91%”** (2012-01-26 10:23:37) [Source: line 18]

Movement: **Dave Westland** moved from **ER patient** to **headwall** (2012-01-26 10:23:43) [Source: lines 19-20]

Movement: **Dave Westland** moved from **headwall** to **ER patient** (2012-01-26 10:23:44) [Source: lines 20-21]

Dropped Item: **Dave Westland** dropped a **PulseOximeter** (2012-01-26 10:23:44) [Source: line 22]

Chat: **Dave Westland** said **“hello there”** (2012-01-26 10:23:50) [Source: line 25]

Movement: **Dave Westland** moved from **ER patient** to **ER bed 3** (2012-01-26 10:24:04) [Source: lines 23-24 and 26-31]

Appendix F

ER Nursing – Pilot Study

F.1 Post-Simulation Questionnaire – Likert Questions

The following table shows the Likert-style questions that were asked of the students participating in the initial usability study for the ER nursing scenario, after they had taken part in a run of the simulation.

Table F.1: Post-Simulation Questionnaire - Likert Questions

Question	**Strongly Disagree**	**Disagree**	**Neutral**	**Agree**	**Strongly Agree**	**Not applicable**
1. This experience has improved my inter-professional teamwork skills						
2. The level of realism was sufficient for suspension of disbelief						
3. The level of realism was sufficient to enable learning						
4. I was able to apply the following knowledge and skills in completing the scenario:						
a. Knowledge of medical facts						
b. Knowledge of relevant procedures						
c. Communication skills						
5. The experience was interesting, and I felt engaged in the experience						
6. I would recommend this experience to other learners						

The following table presents the students' resposes to the Likert-style questions listed previously.

Table F.2: Post-Simulation Questionnaire – Likert Question Results

Participant	Q1	Q2	Q3	Q4a	Q4b	Q4c	Q5	Q6
Student 1	3	4	4	5	5	4	4	4
Student 2	5	5	5	2	2	4	5	5
Student 3	2	4	4	4	4	3	4	4
Student 4	3	4	4	4	4	3	4	4
Student 5	4	2	4	4	4	4	3	4
Student 6	4	4	4	4	4	4	5	5
Student 7	3	3	4	4	4	4	3	3
Student 8	4	4	4	4	4	4	5	4
Student 9	4	4	4	–	–	4	4	4
Student 10	3	4	4	4	4	3	3	4
Student 11	3	2	2	3	3	3	4	3
Student 12	4	4	4	3	3	4	3	4
Student 13	4	4	3	4	4	2	5	5
Student 14	3	3	3	3	3	3	4	4
Student 15	2	1	4	5	5	2	4	4
Average	3.40	3.47	3.80	3.79	3.79	3.40	4.00	4.07

F.2 Post-Simulation Questionnaire – Free-Form Questions

The students were, in addition to the Likert-style questions listed previously, asked for their response to six free-form questions. These questions, along with the students' responses, are shown below.

1. What did you like about this learning experience?
Made you systematically think about the problem
Shows my strengths and weaknesses about what I need to work on in the profession
This new program is an alternative way of learning, it's very interactive
It's fascinating how technology can be used in helping us visually see an ER setting
It was easy to do, and was somewhat realistic.
There was no pressure
I had never used computers for something like this before
I liked interacting with the other disciplines and utilizing their expertise
Great way to learn through virtual simulations!
Virtual, can access from computer
I like the idea that using video game as a tool to facilitate learning. This makes learning more interesting
I liked how I could interact with other nurses
Ability to do health assessment
I liked the virtual effect, use of how to treat a patient
Having to interact with the environment in order to perform tests/examinations

2. What would you change or improve?

Allow for treatments and see if treatments improve or exacerbate the problem

Would add more organization of the goal and roles for all of the players, also a list of what we know and don't know

More info and R&D on more functions and user interactions

More realism to the scenario; not much information about the patient was provided. Having a look at the chart would help

Have more information (patient) available and results of treatment

Some speech-related details

Hard to navigate

Some technical change of the programming of the game

Incorporation of more diagnostic imaging modalities would be great!

Have a brief history of patient - helps with focusing assessment. As I take vital signs, it would be nice for them to appear on a virtual clipboard so I don't have to remember them

Being able to obtain more patient history. Ask them specific questions.

More patient information and medical history available. Perhaps create an entire profile of patient (HPI, Med Hx, Medications, FHx, Allergies, Smoking, Alcohol, etc.) but have the info revealed with the correct questions being asked → kind of like Choose Your Own Adventure.

3. What did you learn about inter-professional collaboration and teamwork?

N/A as there were not others with me. See potential though with the headphones

Communication is a valuable element, communication with client is important

I'm not too sure how this program was supposed to be interdisciplinary

I did not get a chance to communicate with other health professionals as I was the only one who attended at that particular time for this simulation

It's good to talk to each other

The more we talk the more chances of success

It can be done with computers, which is an alternative

It was great listening to input from the other disciplines

Communication is key

Teamwork is crucial in providing quality patient care

Each discipline has a significant role to play with their specific knowledge

That everyone's opinion help in the process of diagnosing the patient.

I didn't communicate with anyone else during the simulation

4. Describe your previous experience with simulations and with virtual worlds.

None

No previous experience, computer games maybe

My first experience and only experience with simulations prior to this was in the biopsy team. Very realistic.

None

1 sim lab, 0 virtual

None

0 virtual worlds. I have used sim labs at MacEwan

No previous experience
Sims: in school and lab work
This is a pretty real life experience, but I had a hard time using the buttons on keyboard because I don't play virtual games
None with virtual worlds. Minimal with simulation (two scenarios in the nursing program)
0

5. How did this experience compare to your expectations, based on previous experience?
Had no previous experience, better than some video games I have heard of
New and exciting
I have never experienced a health related simulation/virtual world program like this
Very different, never considered using the "Save Stan" system to experience an ER situation. Was hoping there were more health professionals available for me to communicate with using this system.
It was interesting
No previous experience in virtual
I had no idea what to expect.
Novel, interesting
It wasn't very specific to my field of study
It is an interesting way to learn - it takes a little skill and practice for it to work in interdisciplinary program
This experience was less intense than the other simulation
No expectations

6. Do you have any other comments?
This may be difficult with people who are not very oriented with the directional changes on a computer.
I think that computers could be used if real people can't meet with each other. I think that it's always better to talk to and see real people.
I was initially intimidated by the computer experience because I don't play computer games but was willing to try and I had fun. Thank you!
The scenario felt beyond the scope of my practice, didn't feel like I had much to contribute.
It was interesting to have the opportunity to work with nurses, etc.
If successful, would be a great resource for students

Appendix G

ER Nursing – Collaborative Study

G.1 Pre- and Post-Simulation Tests

The students participating in the evaluation of the ER scenario were given a test before and after they participated in a run of the simulation. The test is shown in Table G.1.

Table G.1: Pre- and Post-Simulation Test

Part 1: Questions to Ask
Please choose the three most important questions from the following options: 1. Do you have any allergies? 2. Are you on a special diet? 3. Do you have a family history of diabetes? 4. Do you have any medical conditions? 5. Are you taking any medications? 6. Has anyone around you been sick?
Part 2: Assessments
Please identify the 6 most important assessments, and suggest an order in which they should be taken. a. Back pain b. Blood pressure c. Lung sounds d. Level of consciousness e. Jaundice f. Neuro-vital signs g. Psychosocial presentation of patient h. Recent injuries or illnesses i. Respiratory rate j. Temperature k. Pain l. Bowel sounds m. Peripheral pulses
Part 3: Actions
Please identify the 7 most important actions. a. Apply oxygen b. Suction the patient c. Insert oral airway d. Call respiratory therapy e. Consult physician f. ECG g. Draw bloodwork (including CBC, cardiac enzymes) h. Calm patient i. Administer medication j. Start IV k. Prepare code cart l. Sit patient up in bed

The test results are presented in Tables G.2, G.3, and G.4.

Table G.2: Pre-Simulation Test Results

	Test Components			
Participant	**Questions to Ask**	**Assessments**	**Actions**	**Total**
Pre-Sim 1	6	11	10	27
Pre-Sim 2	4	11	10	25
Pre-Sim 3	6	9	12	27
Pre-Sim 4	6	9	12	27
Pre-Sim 5	6	11	10	27
Pre-Sim 6	6	10	6	22
Pre-Sim 7	6	10	12	28
Pre-Sim 8	6	10	10	26
Pre-Sim 9	6	10	10	26
Pre-Sim 10	4	15	10	29
Pre-Sim 11	6	6	8	20
Pre-Sim 12	6	10	12	28
Pre-Sim 13	6	10	12	28
Pre-Sim 14	4	11	12	27
Pre-Sim 15	6	9	12	27
Pre-Sim 16	6	10	12	28
Pre-Sim 17	4	9	8	21
Pre-Sim 18	4	10	12	26
Pre-Sim 19	6	9	12	27
Pre-Sim 20	6	12	8	26
Pre-Sim 21	–	9	8	17
Pre-Sim 22	6	9	8	23
Pre-Sim 23	4	10	8	22
Pre-Sim 24	6	7	8	21
Pre-Sim 25	4	9	10	23
Pre-Sim 26	6	11	10	27
Pre-Sim 27	6	10	10	26
Pre-Sim Avg	5.44	9.85	10.08	25.15

Table G.3: Post-Simulation Test Results

	Test Components			
Participant	**Questions to Ask**	**Assessments**	**Actions**	**Total**
Post-Sim 1	6	8	10	24
Post-Sim 2	4	11	10	25
Post-Sim 3	6	13	14	33
Post-Sim 4	6	14	10	30
Post-Sim 5	6	11	10	27
Post-Sim 6	6	9	12	27
Post-Sim 7	6	10	12	28
Post-Sim 8	4	10	10	24
Post-Sim 9	6	10	8	24
Post-Sim 10	4	9	12	25
Post-Sim 11	4	8	10	22
Post-Sim 12	6	9	12	27
Post-Sim 13	6	12	10	28
Post-Sim 14	–	10	8	18
Post-Sim 15	6	12	8	26
Post-Sim 16	6	8	10	24
Post-Sim 17	6	9	10	25
Post-Sim 18	6	9	10	25
Post-Sim 19	6	12	10	28
Post-Sim 20	6	10	10	26
Post-Sim 21	6	10	10	26
Post-Sim 22	4	8	12	24
Post-Sim 23	6	11	10	27
Post-Sim 24	6	12	12	30
Post-Sim Avg	5.57	10.21	10.42	25.96

Table G.4: Difference Between Pre- and Post-Simulation Test Results

	Test Components			
Participant	**Questions to Ask**	**Assessments**	**Actions**	**Total**
Student 1	0	-3	0	-3
Student 2	0	0	0	0
Student 5	0	0	0	0
Student 6	0	0	0	0
Student 7	0	1	0	1
Student 8	-2	0	-2	-4
Student 9	0	0	2	2
Student 10	0	0	4	4
Student 11	0	-2	-2	-4
Student 12	0	0	0	0
Student 13	0	0	2	2
Student 14	0	1	0	1
Student 15	0	3	0	3
Student 16	2	-2	2	2
Student 17	0	2	2	4
Student 18	2	0	0	2
Student 19	0	1	0	1
Student 20	0	1	-2	-1
Student 21	0	0	-2	-2
Student 23	0	1	-2	-1

G.2 Pre-Simulation Questionnaire

In addition to the pre-simulation quiz, the students were also given a pre-simulation questionnaire, which addressed their previous experience with educational simulations and virtual worlds. The questions included in the questionnaire and the students' results are shown in the following tables. Note that the students are grouped into aggregate categories ("low", "medium", and "high") based on the average of their Likert scores.

Table G.5: Pre-Simulation Questionnaire – Questions

1. Rate your previous experience with educational simulations.

None 1 2 3 4 5 6 7 Extensive

2. Rate your previous experience with virtual worlds/online environments.

None 1 2 3 4 5 6 7 Extensive

3. What are your expectations for the simulation?

Low/Pessimistic 1 2 3 4 5 6 7 High/Optimistic

4. What are your expectations for the virtual world interface?

Low/Pessimistic 1 2 3 4 5 6 7 High/Optimistic

Table G.6: Pre-Simulation Questionnaire – Results

Participant	**Q1**	**Q2**	**Q3**	**Q4**	**Average**
Low level of experience/expectations					
Student 15	1	1	3	3	2.00
Student 25	1	1	4	4	2.50
Student 1	3	1	4	3	2.75
Student 22	4	1	3	3	2.75
Medium level of experience/expectations					
Student 5	4	1	4	3	3.00
Student 3	4	1	4	4	3.25
Student 14	2	1	5	5	3.25
Student 12	3	1	5	5	3.50
Student 20	5	3	4	2	3.50
Student 9	6	1	4	4	3.75
Student 18	5	1	7	2	3.75
Student 7	3	1	6	6	4.00
Student 21	4	2	5	5	4.00
High experience/expectations					
Student 2	4	3	5	5	4.25
Student 4	5	1	7	5	4.50
Student 6	5	2	6	5	4.50
Student 17	7	1	5	5	4.50
Student 11	6	3	5	5	4.75
Student 19	5	4	5	5	4.75
Student 10	5	4	6	5	5.00
Student 13	6	2	6	6	5.00
Student 16	4	7	5	4	5.00
Student 23	5	6	4	5	5.00
Student 24	6	2	6	6	5.00
Student 8	6	5	6	6	5.75
Average	4.36	2.24	4.96	4.44	

G.3 Post-Simulation Questionnaire – Likert Questions

In addition to the post-simulation quiz, the students were also given a post-simulation questionnaire, which elicited their thoughts on the quality and effectiveness of the simulation. The questions included in the questionnaire and the students' results are shown in the following tables. Note that, as with the pre-simulation questionnaire, the students are grouped into aggregate categories ("negative", "neutral", and "positive") based on the average of their Likert scores.

Table G.7: Post-Simulation Questionnaire – Likert Questions

Question	**Strongly Disagree**	**Disagree**	**Neutral**	**Agree**	**Strongly Agree**	**Not applicable**
1. This simulation has enhanced my understanding of the patient analysis and treatment process.						
2. The level of realism was sufficient to **enable learning** in the following areas:						
a. Interacting with the patient						
b. Using medical equipment						
c. Treating the patient						
3. I was able to apply the following knowledge and skills in completing the scenario:						
a. Knowledge of medical facts						
b. Knowledge of relevant procedures						
c. Communication skills						
4. Throughout the scenario, I understood what I needed to do next						
5. The experience was interesting, and I felt engaged in the experience						
6. The scenario was well organized						
7. I would recommend this experience to other students						
8. Overall, this was a useful learning experience, in that it added to my:						
a. Textbook learning						
b. Classroom learning						
c. Practical learning						

Table G.8: Post-Simulation Questionnaire – Likert Question Results

Student	**Q1**	**Q2a**	**Q2b**	**Q2c**	**Q3a**	**Q3b**	**Q3c**	**Q4**	**Q5**	**Q6**	**Q7**	**Q8a**	**Q8b**	**Q8c**	**Average**
Negative (*none*)															
Neutral															
Student 15	4	3	2	3	3	3	4	2	3	3	3	3	3	3	3.00
Student 2	3	1	2	4	4	4	5	2	4	3	3	2	2	4	3.07
Student 9	4	2	3	3	4	4	3	3	3	3	3	3	3	3	3.14
Student 10		1	2	3	4	4	4	3	4	4	3	4	3	2	3.15
Student 16	3	3	3	3	4	4	4	3	3	3	3	3	3	3	3.21
Student 7	3	2	3	4	2	3	5	3	4	4	4	2	3	4	3.29
Student 3	4	2	3	3	4	3	5	3	4	4	4	2	3	4	3.43
Student 11	4	2	3	3	4	3	4	3	4	4	4	3	4	4	3.50
Student 14	3	2	2	4	4	4	4	4	4	3	4	4	4	4	3.57
Student 12	4	2	3	4	4	4	5	3	4	4	4	4	4	4	3.79
Student 19	4	2	4	4	4	4	5	3	4	4	4		4		3.83
Student 8	4	2	4	3	5	5	5	4	4	3	4	4	4	4	3.93
Positive															
Student 13	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4.00
Student 1	4	4	4	4	4	4	5	4	5	4	5	3	3	4	4.07
Student 5	5	3	4	4	4	2	3	3	5	5	5	5	5	5	4.14
Student 17	5	4	5	5	4	4	4	3	4	4	4	4	3	5	4.14
Student 18	4	4	4	4	4	4	5	4	4	5	4	4	4	4	4.14
Student 4	5	3	4	5	5	4	5	4	4	4	4	4	4	4	4.21
Student 6	5	4	4	5	5	5	4	4	5	5	5	4	5	5	4.64
Avg	4.00	2.63	3.32	3.79	4.00	3.79	4.37	3.26	4.00	3.84	3.89	3.44	3.58	3.89	3.70

G.4 Post-Simulation Questionnaire – Free-Form Questions

The students were, in addition to the Likert-style questions listed previously, asked for their response to five free-form questions. These questions, along with the students' responses, are shown below.

1. What did you like about this learning experience?
Never have experienced a virtual simulation before, it was different and a good experience.
It forced us to communicate because we were all looking at things from different perspectives.
It gave us a good opportunity to interact with other health care providers.
The technology of simulation was amazing.

2. What would you change or improve?
I would improve the training of how to interact with the patient and how to perform procedures on the patient before the simulation. Better patient care would result.
It may need to be a bit longer in order to allow people to get used to the system and be comfortable with actions they need to take.
The questions that we can ask the patient should be more detailed.
The lag time of the computers

3. What was your experience with the virtual world interface?
It was difficult compared to reality in some aspects.
I am not very skilled with computers so I was mostly confused about how to do things.

4. What did you learn about patient analysis and diagnosis?
Organization is important in assessments.
To check the vitals

5. Do you have any other comments?
Great!

Appendix H

ER Nursing – Individual Study

H.1 Pre- and Post-Simulation Test Results

The students participating in the evaluation of the ER scenario were given a test before and after they participated in a run of the simulation. The test is shown in Table G.1.

Table H.1: Pre- and Post-Simulation Test

Part 1: Questions to Ask
Please choose the three most important questions from the following options: 1. Do you have any allergies? 2. Are you on a special diet? 3. Do you have a family history of diabetes? 4. Do you have any medical conditions? 5. Are you taking any medications? 6. Has anyone around you been sick?
Part 2: Assessments
Please identify the 6 most important assessments, and suggest an order in which they should be taken. a. Back pain b. Blood pressure c. Lung sounds d. Level of consciousness e. Jaundice f. Neuro-vital signs g. Psychosocial presentation of patient h. Recent injuries or illnesses i. Respiratory rate j. Temperature k. Pain l. Bowel sounds m. Peripheral pulses
Part 3: Actions
Please identify the 7 most important actions. a. Apply oxygen b. Suction the patient c. Insert oral airway d. Call respiratory therapy e. Consult physician f. ECG g. Draw bloodwork (including CBC, cardiac enzymes) h. Calm patient i. Administer medication j. Start IV k. Prepare code cart l. Sit patient up in bed

The test results are presented in Tables H.2, H.3, and H.4.

Table H.2: Pre-Simulation Test Results

Participant	Test Components			Total
	Questions to Ask	**Assessments**	**Actions**	
Pre-Sim 1	6	10	12	28
Pre-Sim 2	6	11	10	27
Pre-Sim 3	4	9	10	23
Pre-Sim 4	4	11	12	27
Pre-Sim 5	6	14	8	28
Pre-Sim 6	6	8	10	24
Pre-Sim 7	5	10	10	24
Pre-Sim Avg	5.14	10.43	10.29	25.86

Table H.3: Post-Simulation Test Results

Participant	Test Components			Total
	Questions to Ask	**Assessments**	**Actions**	
Post-Sim 1	6	8	12	26
Post-Sim 2	6	11	10	27
Post-Sim 3	4	10	14	28
Post-Sim 4	4	8	10	22
Post-Sim 5	6	11	10	27
Post-Sim 6	6	10	10	26
Post-Sim 7	4	12	10	26
Post-Sim Avg	5.14	10.00	10.86	26.00

Table H.4: Difference Between Pre- and Post-Simulation Test Results

Participant	Test Components			Total
	Questions to Ask	**Assessments**	**Actions**	
Student 1	0	-2	0	-2
Student 2	0	0	0	0
Student 3	0	1	4	5
Student 4	0	-3	-2	-5
Student 5	0	-3	2	-1
Student 6	0	2	0	2
Student 7	0	2	0	2

H.2 Pre-Simulation Questionnaire

In addition to the pre-simulation quiz, the students were also given a pre-simulation questionnaire, which addressed their previous experience with educational simulations and virtual worlds. The questions included in the questionnaire and the students' results are shown in the following tables. Note that the students are grouped into aggregate categories ("low", "medium", and "high") based on the average of their Likert scores.

Table H.5: Pre-Simulation Questionnaire - Likert Questions

1. Rate your previous experience with educational simulations.								
None	1	2	3	4	5	6	7	Extensive
2. Rate your previous experience with virtual worlds/online environments.								
None	1	2	3	4	5	6	7	Extensive
3. What are your expectations for the simulation?								
Low/Pessimistic	1	2	3	4	5	6	7	High/Optimistic
4. What are your expectations for the virtual world interface?								
Low/Pessimistic	1	2	3	4	5	6	7	High/Optimistic

Table H.6: Pre-Simulation Questionnaire - Likert Question Results

Participant	**Q1**	**Q2**	**Q3**	**Q4**	**Average**
Student 1	6	1	5	6	4.50
Student 2		—		—	
Student 3	6	2	6	6	5.00
Student 4	7	1	4	7	4.75
Student 5	4	1	4	2	2.75
Student 6	2	1	7	7	4.25
Student 7	3	1	4	4	3.00
Average	4.67	1.67	5.00	4.04	

H.3 Post-Simulation Questionnaire – Likert Questions

In addition to the post-simulation quiz, the students were also given a post-simulation questionnaire, which elicited their thoughts on the quality and effectiveness of the simulation. The questions included in the questionnaire and the students' results are shown in the following tables. Note that, as with the pre-simulation questionnaire, the students are grouped into aggregate categories ("negative", "neutral", and "positive") based on the average of their Likert scores.

Table H.7: Post-Simulation Questionnaire - Likert Questions

Question	**Strongly Disagree**	**Disagree**	**Neutral**	**Agree**	**Strongly Agree**	**Not applicable**
1. This simulation has enhanced my understanding of the patient analysis and treatment process.						
2. The level of realism was sufficient for **suspension of disbelief** in the following areas:						
a. Interacting with the patient						
b. Using medical equipment						
c. Treating the patient						
c. The hospital/ER department						
3. The level of realism was sufficient to **enable learning** in the following areas:						
a. Interacting with the patient						
b. Using medical equipment						
c. Treating the patient						
4. I was able to apply the following knowledge and skills in completing the scenario:						
a. Knowledge of medical facts						
b. Knowledge of relevant procedures						
c. Communication skills						
5. Throughout the scenario, I understood what I needed to do next						
6. The experience was interesting, and I felt engaged in the experience						
7. The scenario was well organized						
8. The following activities were useful in facilitating my learning:						
a. Pre-simulation instructions						
b. Pre-simulation practice						
c. Simulation experience						
9. I would recommend this experience to other students						
10. Overall, this was a useful learning experience, in that it added to my:						
a. Textbook learning						
b. Classroom learning						
c. Practical learning						

Table H.8: Post-Simulation Questionnaire - Likert Question Results

Student	**Q1**	**Q2a**	**Q2b**	**Q2c**	**2d**	**Q3a**	**Q3b**	**Q3c**	**Q4a**	**Q4b**	**Q4c**	**Q5**	**Q6**	**Q7**	**Q8a**	**Q8b**	**Q8c**	**Q9**	**Q10a**	**Q10b**	**Q10c**
Student 1	4	4	4	4	4	5	5		4	4	4	4	4	4	4	4	4	4	4	4	5
Student 2	3	2	4	4	3	4	4	4	4	4	2	4	4	4	4	4	4	4	4	2	4
Student 3	5	3	4	4	3	3	4	4	4	4	3	3	4	4	4	4	5	5	3	4	5
Student 4	5	4	4	4	2	4	4	4	4	2	3	4	4	4	4	5	5	5	5	5	5
Student 5	4	4	4	4	4	4	4		5	5	2	4	4	4	5	3	5	4	4	4	5
Student 6	3	3	4	4	3	3	4		4	4	2	4	4	3	4	4	4	4	2	2	4
Student 7	4	4	4	4	3	4	4	4	4	4	3	3	4	4	4	4	4	4	4	4	4
Average	4.00	3.43	4.00	4.00	3.14	3.86	4.14	4.00	4.14	3.86	2.71	3.71	4.00	3.86	4.14	4.00	4.43	4.29	3.71	3.57	4.57

H.4 Post-Simulation Questionnaire – Free-Form Questions

The students were, in addition to the Likert-style questions listed previously, asked for their response to five free-form questions. These questions, along with the students' responses, are shown below.

1. What did you like about this learning experience?
A chance to try out basic interventions without "real" consequences ona real patient.
Easy to understand
Interactive and different
Hands-on, clinical approach
Not as scary as first day of clinical!
It was straight forward and well explained
Engaging/easy to use simulation activity
Gave you an opportunity to think through what might actually be going wrong with the patient if you followed the vital signs.
It encouraged me to think about what assessments were relevant for treating the patient
It is a non-threatening environment to learn about assessing and treating patients

2. What would you change or improve?
More of a sense of beginning or end...
I would like more assessment questions to gain a better understanding of the patient's condition... more subjective questions please
Medical air pump -¿ have always used inhalers in past
Ask patient questions more directly
Needed to ask about smoking -¿ emphysema?
More interaction with patient
I would keep the question box up and if you wanted to assess all those areas you can continually click and go. Otherwise you have to jump back and forth between the questions and patient to ask the next assessment question.
I would like to see more options for interacting with the patient in terms of talking and getting the patient to move. I would like to get some feedback while doing the scenario.
If the change in the patient condition could be quicker.

3. What did you learn about patient analysis and diagnosis? Consolidation of nursing process
Air pump for atrovent/ventolin
Possibly asthma - wheezing
I learnt that you can do some assessment skills to get leads into what should be done next.
even though the computer told me at the end I should have assessed this or that, at the time I felt I didn't need to, and just went looking for what my gut told me was right. I believe I was able to make the right diagnosis in the end.
It is important to think through what the symptoms indicate
Assessing and treating needs to be more systematic.

4. What was your experience with the virtual world interface?
Low-pressure learning experience

None
Interesting!
I tried it during Save Stan, but it was hard competing with the other students that day to figure out who has what medical apparatus and how it actually worked. I spent most of my time that day trying to just take my avatar for a walk around the ER.
It was easy to manipulate and understand

5. Do you have any other comments?
Definitely a useful tool for learning clinical drills. I think these types of training and practise methods would be incredibly valuable in running programs. Please keep this up!
Would be great to have a number of those scenarios before our first clinical experience for clinical 1.
Was awesome! Thanks
This was a really good experience! I think everyone should give it a try.
I really enjoyed the experience, would have liked to see a definite conclusion to the treatment
This would be a good learning tool to have.

H.5 Attitude Assessement – Computer-Assisted Instruction

I administered Allen's tool for measuring attitude toward computer assisted instruction [2]. The tool uses 14 bipolar adjective scales (e.g., "valuable" vs. "worthless"), and asks the student to record his or her first impression about each item. The tool is shown in Table H.9, and the results of administering the tool before and after the simulation are shown in Tables H.10, H.11, and H.12. Note that the numerical values shown in Table H.9 are not actually shown in the tool, but are presented here in order to clarify how values are assigned to each item in the tool.

Table H.9: Attitude Assessment Tool

Rigid	1	2	3	4	5	6	7	Flexible
Useful	7	6	5	4	3	2	1	Useless
Stimulating	7	6	5	4	3	2	1	Boring
Meaningless	1	2	3	4	5	6	7	Meaningful
Pleasant	7	6	5	4	3	2	1	Unpleasant
Valuable	7	6	5	4	3	2	1	Worthless
Creative	7	6	5	4	3	2	1	Stifling
Impersonal	1	2	3	4	5	6	7	Personal
Efficient	7	6	5	4	3	2	1	Inefficient
Inappropriate	1	2	3	4	5	6	7	Appropriate
Comfortable	7	6	5	4	3	2	1	Uncomfortable
Non-threatening	7	6	5	4	3	2	1	Threatening
Overpowering	1	2	3	4	5	6	7	Easy to control
Timesaving	7	6	5	4	3	2	1	Time-consuming

Table H.10: Pre-Simulation Attitude Assessment

Student	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**Total**
Student 1	2	6	4	2	5	3	5	3	5	6	4	2	3	2	52
Student 3	6	6	6	5	4	6	7	2	6	6	4	7	5	5	75
Student 4	7	7	7	7	7	7	7	4	5	6	7	7	5	2	85
Student 5	4	4	4	4	5	4	6	2	4	4	7	7	4	4	63
Student 6	6	5	5	5	4	4	6	2	6	5	5	5	3	4	65
Student 7	4	4	6	5	4	4	5	2	4	4	5	5	4	4	60
Average	4.83	5.33	5.33	4.67	4.83	4.67	6.00	2.50	5.00	5.17	5.33	5.50	4.00	3.50	66.67

Table H.11: Post-Simulation Attitude Assessment

Student	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**Total**
Student 1	7	7	7	7	7	7	7	5	7	7	6	6	7	6	93
Student 2	4	7	5	5	6	6	6	4	6	6	6	6	6	4	77
Student 3	5	7	6	6	7	6	7	4	5	6	6	7	5	6	83
Student 4	7	7	7	7	6	7	7	1	5	7	7	7	7	3	85
Student 5	6	6	5	5	5	5	7	3	6	5	7	7	6	6	79
Student 6	5	5	6	5	6	6	6	5	5	4	6	6	5	4	74
Student 7	4	6	6	5	4	5	5	4	3	5	6	6	5	5	69
Average	5.43	6.43	6.00	5.71	5.86	6.00	6.43	3.71	5.29	5.71	6.29	6.43	5.86	4.86	80.00

Table H.12: Pre vs. Post-Simulation Attitude Assessment

Student	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**Total**
Student 1	5	1	3	5	2	4	2	2	2	1	2	4	4	4	41
Student 3	-1	1	0	1	3	0	0	2	-1	0	2	0	0	1	8
Student 4	0	0	0	0	-1	0	0	-3	0	1	0	0	2	1	0
Student 5	2	2	1	1	0	1	1	1	2	1	0	0	2	2	16
Student 6	-1	0	1	0	2	2	0	3	-1	-1	1	1	2	0	9
Student 7	0	2	0	0	0	1	0	2	-1	1	1	1	1	1	9
Average	0.83	1.00	0.83	1.17	1.00	1.33	0.50	1.17	0.17	0.50	1.00	1.00	1.83	1.50	13.83