Ph.D Year 1: Transfer Report (RD5)
Towards a Patient Simulator Framework for Evaluation of e-Health Environments: Modeling, Techniques, Validation and Implementation

Owen Lo

February 2011
# Contents

1 Introduction 1
   1.1 Introduction ........................................... 1
   1.2 Research Questions ................................... 1
   1.3 Aim and Objectives ................................. 1
   1.4 Current Research Overview ......................... 2
   1.5 Report Structure .................................. 3

2 Background & Theory 4
   2.1 Introduction ........................................... 4
   2.2 E-Health ............................................. 4
      2.2.1 History and Definitions ......................... 4
      2.2.2 Advantages of e-Health .......................... 5
      2.2.3 E-Health in the Cloud ......................... 6
      2.2.4 Technical Concerns of e-Health ................. 7
      2.2.5 Organisational Concerns of e-Health ........... 7
   2.3 Theory of Simulation ................................ 8
      2.3.1 Introduction .................................... 8
      2.3.2 Definition and History .......................... 9
      2.3.3 Emulation vs Simulation ......................... 9
      2.3.4 Taxonomy of Simulation ......................... 10
      2.3.5 Challenges in Successful Simulation .......... 11
   2.4 Conclusion ........................................... 12

3 Literature Review 14
   3.1 Introduction ......................................... 14
   3.2 E-Health Review ..................................... 14
      3.2.1 Sensor Based .................................... 14
      3.2.2 DACAR Project .................................. 15
      3.2.3 Microsoft HealthVault .......................... 17
5.4.2 Experimentation ................................................. 42
5.4.3 Summary and Evaluation ................................. 43
5.5 Additional Experimental Work: WCF Service Implementation ........... 45
  5.5.1 Advantages .................................................. 45
  5.5.2 Drawbacks .................................................. 46

6 Proposed Novelty and Future Work .................................. 47
  6.1 Novelty of Research ........................................... 47
  6.2 Time Plan and Mile Stones .................................. 48
    6.2.1 Time Plan ................................................. 48
    6.2.2 Abstraction of Patient Model .................................. 50
    6.2.3 Refinement of Patient Simulator Framework .................. 50
    6.2.4 Validation of Patient Simulator Model and Implementation ...... 50
    6.2.5 Evaluation Metrics Formalisation .......................... 51
    6.2.6 Knowledge Transfer with Health Professional ............... 51
    6.2.7 Evaluation Workshop with Chelsea and Westminster Hospital ... 51
    6.2.8 Experimentation with DACAR project ....................... 51
    6.2.9 Six and Twelve Months Progress Review .................... 51
    6.2.10 Journal and Conference Selection .......................... 52
# List of Figures

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1</td>
<td>Simulation Core Components Taxonomy [1]</td>
<td>10</td>
</tr>
<tr>
<td>2.2</td>
<td>Simulation Interface Taxonomy [1]</td>
<td>11</td>
</tr>
<tr>
<td>3.1</td>
<td>Sensor Based e-Health Service [2]</td>
<td>15</td>
</tr>
<tr>
<td>3.2</td>
<td>Overview of the DACAR e-Health service platform</td>
<td>16</td>
</tr>
<tr>
<td>3.3</td>
<td>Health Vault Overview [3]</td>
<td>18</td>
</tr>
<tr>
<td>3.4</td>
<td>Google Health [4]</td>
<td>19</td>
</tr>
<tr>
<td>3.5</td>
<td>Discrete vs Continuous Event Based Simulation</td>
<td>21</td>
</tr>
<tr>
<td>3.6</td>
<td>Deterministic Behavior Example</td>
<td>22</td>
</tr>
<tr>
<td>3.7</td>
<td>Probabilistic Behavior Example</td>
<td>22</td>
</tr>
<tr>
<td>3.8</td>
<td>SimMan®Robotic Mannequin [5]</td>
<td>26</td>
</tr>
<tr>
<td>3.9</td>
<td>HPS and VP Integration using HIDEP exchange</td>
<td>26</td>
</tr>
<tr>
<td>3.11</td>
<td>Input Parameters for Type 1 Diabetes Simulation [7]</td>
<td>29</td>
</tr>
<tr>
<td>4.1</td>
<td>Patient Simulator Framework</td>
<td>32</td>
</tr>
<tr>
<td>4.2</td>
<td>Patient Model</td>
<td>33</td>
</tr>
<tr>
<td>5.1</td>
<td>Patient Simulator First Prototype</td>
<td>37</td>
</tr>
<tr>
<td>5.2</td>
<td>Patient Simulator Current Prototype</td>
<td>37</td>
</tr>
<tr>
<td>5.3</td>
<td>CareMagic - Blood Pressure Chart</td>
<td>38</td>
</tr>
<tr>
<td>5.4</td>
<td>CareMagic - EWS Chart</td>
<td>38</td>
</tr>
<tr>
<td>5.5</td>
<td>Normal Distribution of Heart Rate</td>
<td>41</td>
</tr>
<tr>
<td>5.6</td>
<td>Simulation of Normal Patient</td>
<td>42</td>
</tr>
<tr>
<td>5.7</td>
<td>Comparison of Simulated Heart Rate with Real-Life Data</td>
<td>44</td>
</tr>
<tr>
<td>5.8</td>
<td>Abstract Model of the Patient Simulator as a WCF Services</td>
<td>46</td>
</tr>
<tr>
<td>6.1</td>
<td>Year 2 Time Plan</td>
<td>49</td>
</tr>
<tr>
<td>6.2</td>
<td>Year 3 Time Plan</td>
<td>50</td>
</tr>
</tbody>
</table>
List of Tables

3.1 Categories of Simulation in Second Life . . . . . . . . . . . . . . . . . 28

5.1 Mean and Standard Deviation of Vital Signs . . . . . . . . . . . . . . 42
5.2 Minimum and Maximum Heart Rate Values . . . . . . . . . . . . . . 43
Abstract

TrADITIONALLY, record keeping of patient data within a healthcare environment has been conducted using paper based systems. With the exponential growth in modern technology, electronic systems and highly sophisticated medical devices have started replacing this traditional method. Furthermore, with the Internet now a common everyday commodity in most developed parts of the world, the healthcare industry has began migrating their systems to take advantages of modern communication infrastructures.

Known as e-Health, this concept is the delivery of healthcare services on a mass scale. It enables cross-communication of medical data along with the delivery of health care services straight from the Internet itself. No longer are healthcare records isolated as e-Health enables the sharing of medical data from all facilities, from the largest hospitals to the smallest drop-in clinics, using distributed computing techniques such as cloud computing.

However, with the growth in the development of e-Health, including both platforms and services, the issue of how to evaluate these patient centric environments remains unanswered. Privacy of Data laws make it difficult to use real patient data, whilst live deployment of e-Health environments require overcoming many more legal and ethical requirements. Thus, in this report, it is proposed that the research and development of a patient simulator framework - using computer based simulation techniques - enables the evaluation of e-Health environments under some early defined metrics including efficiency, reliability, security and scalability.

As part of this report, the main aim and objectives of this work is outlined along with a comprehensive literature review of relevant subjects. The proposed novelty of this framework is presented and future work to be conducted is outlined.
Chapter 1

Introduction

1.1 Introduction

This chapter introduces the topic of this research. It presents the research questions which this work aims to investigate. The aim and objectives is outlined. An outline of research conducted on e-Health environments is provided. The difficulties in evaluating such implementations is highlighted thus reinforcing the novel contribution this research aims to provide: a patient simulator for the evaluation of e-Health environments. Finally, the structure of this report is provided.

1.2 Research Questions

The main research questions are as follows:

- What are the current techniques used in simulation and, in particular, methodologies for the simulation of patients in a healthcare environment?
- In the provision of a patient simulator, how does one go about providing a model of a patient? More specifically, in the simulation of patients catered towards evaluating e-Health environments, what are the key attributes which a patient must consist of?
- Can a patient simulator be used in place of real life patients to evaluate e-Health environments and what are the metrics of evaluation to be applied?
- Prior to the evaluation of e-Health environments using a patient simulator, how can the patient simulator first be validated to ensure both its accuracy and realism in the model of a patient? What quantitative and qualitative validation techniques can be used to conduct such an evaluation?

1.3 Aim and Objectives

The main aim of this research is as follows:

- Provision of a patient simulator framework which enables the evaluation of e-Health environments under the metrics of efficiency, reliability, security and scalability.
In order to meet this aim, the following objectives are defined:

- Explore existing methodologies in the simulation of patients in healthcare environment. Furthermore, a thorough review of past and present techniques used in computer simulation needs to be conducted in order to have the most up to date knowledge on this discipline.

- Data gathering for the provision of a patient model. Key attributes of a patient need to be established.

- Conduct testing and validation of e-Health environments in order to assess the viability of this approach.

1.4 Current Research Overview

The application of e-Health has brought about a change in the way health care services interact with patient data. Traditionally a paper-based system [8], the migration to electronic based records, e.g. Electronic Health Records (EHR), has provided a far more efficient method of storing patient data. Not only has electronic records removed the need for physical storage space, it brings about the ability for medical staff to quickly and easily look up a patient’s medical record and - in the case of a patient moving to another hospital - such information can be easily transmitted to the new facility with very little delay.

Thus, it can clearly be stated that e-Health brings about an ease of communication for the health care industry as a whole [9]. Unfortunately, there is still a lack of trust in computer based systems. A prime example of this is in 2009, when the English National Health Service (NHS) lost thousands of medical records [10] due to a lack of security in their computer systems. One could easily justify that patients may still feel rather uneasy about medical facilities storing their personal data in an e-Health environment.

As it stands, there is currently no clear method in evaluating how effective e-Health platforms and services are in regards to security, scalability, auditability and efficiency. The ideal scenario for the evaluation of e-Health technologies would be live deployment in hospital and clinical environments. This has certainly been achieved in the past, with recent research including [11, 12, 13] all being carried out on real life patients. However, a fundamental theme found in such clinical trials is the fact that they are heavily funded by health care organisations [14]. This fact itself is not a problem but, for researchers without the lack of funding, attempting to evaluate their e-Health platform and services is made far more difficult due to the lack of resources available. Furthermore, with the vast amount of legal and ethical ramifications in performing real-life clinical trials [15, 16] using actual patients may not even be desirable until researchers feel that their work has progressed to a point whereby live deployment would not produce any risk, e.g. lost of privacy, to persons participating in a clinical trial.

In order to provide for e-Health infrastructure which is commercially and technically viable, there is a need to carry out extensive clinical trials and evaluations. But, in order to carry out such trials, researchers require the backing of health care organisations and must also overcome the legal and ethical barriers imposed in clinical trials.
Hence, in this report, it is proposed that research into the concept of simulating patient’s within a e-Health environment be carried out. Through the simulation of virtual patients, extensive testing and validation of e-Health platforms and services may be conducted whilst overcoming both legal and ethical issues in live clinical trials. Furthermore, it mitigates the need to use real-life patient data, thus overcoming the legal and ethical barriers which are imposed in attempting to carry out live clinical trials.

1.5 Report Structure

- **Introduction** - This chapter introduces the topic of this research along with outlining the research questions and aim and objectives of the research.

- **Background and Theory** - The background of healthcare, including the concepts of e-Health, is provided in this chapter. Furthermore, it outlines the current difficulties faced in the deployment of e-Health environments and how research into the patient simulator helps overcome this problem. Finally, as the scope of this research is within the field of computer simulation, the theoretical concepts behind this discipline is provided.

- **Literature Review** - In the literature review, a concise analysis of both e-Health environments and simulation techniques and validation is provided. In order to provide justification for the novelty of this research, a review of existing healthcare simulations is also conducted and it is shown that the patient simulator differs from the goals these works.

- **Proposed Patient Simulator Framework** - The framework of the patient simulator is provided in this chapter. It provides an abstract view of the patient simulator which is comprised of four main components: user interface, patient models, simulation engine and communication interface.

- **Work to Date** - The work which has been conducted, in relation to this research, is provided in this chapter. It presents the implementation prototypes of the patient simulator, early work conducted in testing the e-Health service CareMagic developed by Kodit [17] and a description on some of the papers which have been submitted this year. Furthermore, a WCF implementation of the patient simulator is also presented.

- **Proposed Novelty and Future Work** - The final chapter concludes this report by proposing the novel aspect of the patient simulator. In particular, it is proposed that the patient simulator will enable the evaluation of e-Health environments under the metrics of efficiency, reliability, security and scalability. Furthermore, the future work which is to be conducted along with an inclusion of a time plan for the next 2 years is provided.
Chapter 2

Background & Theory

2.1 Introduction

This chapter discusses the background and theory to the topics which are covered in this report. Two main sections make up this chapter: e-Health and Computer Simulation. It provides the background knowledge necessary for the literature review and following chapters of this report.

2.2 E-Health

In the first section, the history of healthcare and definition of e-Health is provided. It can be stated that e-Health has potential in bringing about a positive change in healthcare environments therefore, the advantages of e-Health is presented. However, many issues still need to be overcome in the implementation of e-Health environments. This section presents both technical issues (from the developers perspective) and organisational issues (from the healthcare industry perspective) which need to be overcome in order for e-Health to be widely accepted.

2.2.1 History and Definitions

Beginning in the early 70s, the term Medical and Nursing Informatics was first coined to describe the concept of applying computer based systems for the provision of health care services [9]. With the exponential advancement of technology and communication networks, healthcare environments began to replace traditional paper-based record keeping services with electronic counter-parts such as EHRs. It is now common for almost all healthcare environments, from small drop-in clinics to the large complex hospitals, to use computer systems in their provision of health care.

However, the definitions provided for using such modern technologies for the provision of health care services is still debatable. In particular, there has been much ambiguity between the usage of the terms "e-Health" (alternatively defined as 'eHealth' in American English) and "Health Informatics". In a technical report produced by the Internation Technology Union, they state that e-Health and Health Informatics may be used interchangeably [18] whilst the work of both [19, 20] argue that e-Health refers to the provision of health care services which are predominantly Internet based.
To further complicate matters, some researchers have proposed that Health Informatics (alternatively known as Medical Informatics) is concerned with the systematic processing of data, information and knowledge in medicine and health care via computational based systems [21, 22]. Alternatively, Health Informatics is concerned with the handling and carrying out tasks in regards to medical data using computers.

From all definitions provided here, one key point may be highlighted. Prior to the introduction of modern communication infrastructures, i.e. the Internet, it is apparent that Medical Informatics referred to the process of using computer systems to manage and process medical data. However, when the Internet become widely used by the general population, the term “e-Health” was coined which (at the time) specifically meant using the Internet as the primary means in the provision of health care services.

In this report, similar to the view stated by [18], it is argued that health informatics and e-Health are exactly the same concept. To justify, it can be argued that nearly all computer systems have some form of modern communication method, i.e. TCP/IP protocol, therefore it is not felt that a distinction between the two terms should, or can, be made. Thus, for the scope of this report, the use of the term “e-Health” refers to the practice of applying modern computing and communication technologies for the provision of health care services though this term may be used interchangeably with "Health Informatics".

Furthermore, another definition which must be clarified is the terms "health care" and "healthcare". Health care, as two words, refers to the act of providing some form of service, e.g treatment, counseling, advice, to a patient whilst healthcare, one word, refers to the system or systems which provide the delivery of health care.

2.2.2 Advantages of e-Health

Many advantages can be found in applying e-Health to a healthcare environment. One key example is EHRs. Prior to the introduction of EHRs, all medical information regarding a patient were recorded on paper. Using EHRs allows healthcare environments to now store such information electronically, a practice which allows for a far more simple and efficient method of storing data [8]. Further backing is provided by Grogan [23] in which he states that there is evidence to suggest that EHRs provide a more complete and error free method for the storage of patient data.

The work of Mullner and Chung provides additional support for this argument, in which the authors state that “paper records contain too many errors and inefficiencies, and they hinder the communication between health care providers” [24]. Such a quote highlights another very important advantage in using electronic records: ease of communication. Having all records stored electronically allows for medical staff to easily look up a patients medical record and - in the case of a patient moving to another hospital - such information can be easily transmitted to the new hospital with very little delay.

In the migration of paper based systems to modern communication infrastructures, e-Health has also taken advantage of the cutting edge Internet technology available: cloud computing. Via the use of cloud computing, e-Health environments can be deployed under a centralised location and be accessed anywhere in the world so long as a Internet connection is available. This opens up the possibility of cross communication of medical data between hospitals along with delegating the support and
maintenance of computing resources to a third-party, i.e. the cloud provider, along
with cutting down costs involved in supporting these infrastructures [25].

It has been found, from the literature review, that most implementations of e-Health
environments take advantage of cloud computing. Hence, the next section provides
a brief background to this emerging technology.

2.2.3 E-Health in the Cloud

A common theme found in the literature is that many researchers of both e-Health
services and platforms have proposed their infrastructure to be hosted on the cloud.
It should be made explicitly clear that e-Health does not necessarily need to be part
of a cloud network but the advantages of doing so are described in this following
section.

First proposed in the 60s by computer scientist John McCarthy, cloud computing
is the idea that computing resources, including hardware and software, could be
provided to end users at a very low cost. John McCarthy referred to this idea as
utility computing [26] though it is now commonly referred to it as cloud computing.
A huge variety of definitions have been given to this term [27] but, in essence, cloud
computing is providing computer based services over the Internet. More technically,
it is the provision of services to end users using a distributed model. In other
words, it is the provision of applications, platforms and hardware and software on
an on-demand basis [28]. The term “cloud” comes from the fact that end users need
to know the physical location in which the services are being accessed from.

Three main models of service may be provided in a cloud environment: Software as
a Service (SaaS), Platform as a Service (PaaS) and Infrastructure as a Service (IaaS). A
brief description of each model is provided as follows:

- **Software as Service** is the delivery of software services to a end user remotely.
  Rather than a user purchasing, and installing, the software locally, the software
  may be accessed on-line instead. This type of service may be implemented as
  a subscription based service, or pay per use service. There are even instances
  of this type of service being free to use. An example of SaaS is Google Docs
  [29], which allows users to create text, spreadsheet and presentation documents
  on-line. SaaS will usually use a "thin-client" model, which is a client which of-
 fers very little functionality except for establishing a connecting with the server
  in which the SaaS is hosted on. An example of a thin-client model is a web
  browser.

- **Platform as a Service** is the provision of a platform for the users to run their
  applications on. Typically, this will be a stack within a virtual machine. An ex-
  ample of PaaS is Amazon’s Elastic Cloud Computing (EC2) [30]. Amazon’s EC2
  is a paid service which provides users with virtual servers to run applications
  on. Based on the cost, the specifications of the platform may vary, however,
  users will not be aware of the underlying software or hardware architecture.

- **Infrastructure as a Service** is the provision of low level services such as CPU,
  Operating System and Storage facilities. An example of IaaS is Amazon’s EC2
  storage service. This allows users to upload and download data objects via
  web services/application programming interfaces (API) including, for exam-
  ple, Java API’s and Simple Object Access Protocol (SOAP) messages. This al-
  lows developers to have secure access and storage to objects.
In practice, there is much ambiguity over which category of service is being offered in a cloud environment. For example, to provide for SaaS, there must some form underlying platform for software services to be offered in the first place. Furthermore, some form of infrastructure will also be required in order to carry out the processing and storage of data. Therefore, it should be noted that the categories presented in the list above are simply used to provide a abstract overview of cloud computing and, in practice, all three services are used in conjunction to provide for a cloud based environment.

Though hosting e-Health environments in a cloud based network certainly has its advantages, regardless of whether or not this choice is made, both technical and organisational concerns in the implementation and deployment of e-Health environments still need to be addressed. The next two sections provide the background as to what the key concerns are.

2.2.4 Technical Concerns of e-Health

As with most emerging and new technologies, the risk of security implications in e-Health environments has been widely discussed. It can be stated that the key security concerns in e-Health is the similar to any threat to information security: confidentiality, integrity and availability (C.I.A) [31]. Of equal importance, in the case of e-Health, is the auditability of information. Since the entire infrastructure of e-Health is catered towards patients and the data which they produce, it is of utmost importance that not only is the data protected (via the concept of C.I.A) but also tracked and logged in order to track who has access to the data at any given time.

The concept of C.I.A has been a formally established practice in safeguarding computer systems and auditability of data is being developed at the time of writing this report [32] but the methodology for evaluating e-Health to ensure these safeguards work as expected are still not in place. Hence, this reinforces the importance of the patient simulator proposed in this research since it aims to evaluate e-Health environments under the metrics of efficiency, reliability, security and scalability.

Along with information security issues which the implementation of e-Health environments face, there is also organisational barriers which the healthcare industry, as a whole, must overcome. The next section discusses these three main organisational barriers.

2.2.5 Organisational Concerns of e-Health

Hill and Powell [33] states three main areas which need to be overcome, including legal barriers, operational barriers and cost/benefit barriers. An outline on some of the main issues in regards to these three areas are discussed.

• Legal Barriers: The subject of legal issues faced in the evaluation of e-Health infrastructures has been touched upon earlier in this report. However, in this case, Legal Barriers refers to the legal concerns faced by healthcare organisations rather than the groups of researchers who are proposing e-Health solutions. The fundamental issue faced by healthcare organisations in regards to this barrier is enforcing privacy of data [34, 35]. As healthcare organisations are a patient centric industry, it is of vital importance that, in the application of
e-Health solutions, considerations must be taken to ensure the laws of privacy are followed.

Privacy laws may differ from country to country but an underlying theme is protecting the rights of an individual. The main difficulty faced in enforcing this law is ensuring only authorised medical personnel have access to patient data which is relevant to their day-to-day work [36]. Furthermore, the question of whom grants access rights in the first place is still unclear. It would seem natural for each individual organisation, e.g. hospitals and clinics, to manage their own access rights internally however, as it was identified earlier in this report (Section 2.2.2), e-Health brings about the possibility of cross communication of patient data between different health care facilities thus, methods managing access rights in order to ensure privacy laws are followed is still not formally established.

• **Operational Barriers**: This area of concern relates to the interoperability of systems which e-Health aims to provide [33]. First, as part of this barrier, healthcare organisations must overcome the issue of providing an interface to allow pre-existing computer systems to communicate with new systems which e-Health will introduce. Second, in order to allow for cross communication of medical data, such as patient records, between different healthcare organisations, a standard electronic language must be provided. The formal agreement on what the best method to communicate such data between organisations has yet to be established [37].

• **Cost/Benefit Barriers**: As the name implies, this barrier relates to the question of the cost in implementing e-Health solutions and whether or not the benefits e-Health provide outweigh such cost. From a technical stand point, in terms of speed, reliability and efficient storage of data, the implementation of e-Health solutions is clearly advantageous in comparison with past methods such as the paper based record keeping systems.

However, from a healthcare organisational stand point, these benefits may not outweigh the cost of implementing e-Health solutions. As shown by Mearian [38], it can cost tens of thousands of dollars in the implementation of e-Health solutions and this does not even include the requirement of hiring teams of IT professionals to support and maintain the software throughout its life cycle. Furthermore, as stated by Brailer [34], the first healthcare organisations which chose to implement e-Health may see subtle, if any, benefits since they will need to wait for other organisations to implement similar solutions before advantages in communication of medical data can be taken.

### 2.3 Theory of Simulation

#### 2.3.1 Introduction

As this research is heavily involved in the concept of simulation, this section aims to provides readers with the background to this subject. A broad view of simulation is taken in this section. In particular, it aims to provide the theoretical knowledge in the concepts of simulation. Specific techniques and methodologies are not discussed in this portion of the report. Rather, the aim is to simply provide the fundamental aspects of simulation.
2.3.2 Definition and History

One definition of the term simulation, as given by Oxford Dictionary, is as follows:

"The technique of imitating the behavior of some situation or process (whether economic, military, mechanical, etc.) by means of a suitably analogous situation or apparatus, esp. for the purpose of study or personnel training." [39]

The concept of simulation dates back to many centuries with military establishments using simulation to conduct war games both for training and strategy decision making purposes. Perhaps one of the most well known traditional military simulations, which has evolved into a game still played today, is chess [40]. Similar to its roots, some of the first ever computer based simulation implemented were of a military nature [41] including the Manhattan Project [42] in which the simulation of nuclear detonations were carried out using the Monte Carlo algorithm [43].

Simulation techniques have progressed a significantly from it's early beginnings. The applications of simulation is widespread, with many usages, including aerospace simulation [44], finance simulation [45] and, perhaps most relevant to this research, medical simulation. Each method of simulation may differ based on the algorithms and implementation method used however, the end goal of each simulation will generally fall under one of the following categories: understanding a system, prediction of behavior, training or entertainment. Furthermore, in some cases, simulation may breach more than one category, a prime example being Flight Simulators which can provide both entertainment and education for users.

It should be noted that simulation is not limited to a computer based environment. For example, practicing for a job interview with a friend could be considered a form of simulation. In the scope of this report, the focus is on computer based simulation.

2.3.3 Emulation vs Simulation

Simulation is often confused with emulation. Though similar, emulation is inherently different from simulation. Emulation refers to the goal of imitating the hardware of computer systems including fine details such as emulating microchips and transistors whilst, in comparison, simulation is entirely software based and does not consider the underlying hardware which it is running on [46].

Using a aircraft flight simulator/emulator, as an example, emulation would involve imitating the entire aircraft with 100% identical behavior. This would involve the emulation of every aspect of the aircraft including the engine, wings, flight controls etc. and how they relate to each other. In other words, the physical characteristics of an aircraft and how they affect the flying of the plane will not only be modeled but also implemented in a precise one-to-one manner. In simulation, all components of the airplane will only be "simulated" and although "flying" the plane may be subjectively realistic via good programming, it does not accurately reflect how a plane’s internal components function or work together to induce flight as would be seen in the real world.

To provide an additional example, in research conducted on carrying out Denial of Service (DoS) attack experiments using simulated and emulated tools, it was concluded that simulated environments would "abstract a number of system attributes" [47]
which resulted in a drastic difference of results in comparison with emulated environments. The reason, as justified by the authors of this work, is system attributes, i.e. CPU, memory and device drives, in emulated environments could become bottlenecks during the DoS attack whilst in simulated environments these attributes may be simulated but do not emulate how they work or function in real-life.

2.3.4 Taxonomy of Simulation

In providing a taxonomy for computer based simulation, the work conducted by Sulistio et al.[1] group this subject matter into three main properties: Presence of Time, Basis of Value and Behavior. Figure 2.1 presents the three main components of simulation.

![Figure 2.1: Simulation Core Components Taxonomy [1]](image)

- **Presence of Time** relates to whether or not a simulation will consider the attribute of time. For example, simulating a queuing system in a shop will likely consider time in a dynamic nature (in order to measure how long customers must wait), whilst a simulation of a command line terminal will not require a timing factor (since the simulator only requires input from the user and produce some form of output). The concept behind the Presence of Time property is quite simple: either a simulation uses a time function (incrementing or decrementing), or it does not.

- **Basis of Value** refers to the range of entities, i.e. variables, modeled which the simulation is capable of generating. This can either be discrete entities or continuous entities. Discrete entities are limited in their range, e.g. variables of 1 to 10, whilst continuous entities can have an infinite range, i.e. variables 1, 1.1, 1.2, 1.3 ... etc.

- **Behavior** of simulation refers to whether or not repeated simulation using the exact same parameters and values will result in the same results occurring. In other words, the behavior of a simulation is either deterministic or probabilistic. In deterministic simulation, no random events will occur therefore, repeated simulations will always result in the same outcome. In probabilistic simulation, the opposite is true, and repeated simulations may produce slightly different results.

Along with the core components of simulation, Sulisto et al. also define "User Interface" as part of simulation’s taxonomy. The user interface relates to how a use interacts with the simulator. In simulators which choose to implement a interface, the categories of Visual property including Design, Execution and Integrated Environment
are considered whilst nothing will be presented in the Non-Visual implementations (Figure 2.2).

![Simulation Interface Taxonomy](image)

- **Design** relates to the interface presented to the user using the simulator. This enables ease of use and configuration of simulator in comparison with a non-visual method which would require entry of code which is more time consuming. Forms, buttons and drag-and-drop functionalities help achieve the aim of providing an easy-to-use simulator.

- **Execution** describes what visual feedback is presented to the user whilst the simulator is running. In the presentation of graphs and animations, the current running status of a simulator can be observed in a far simpler manner than simply text and numbers.

- **Integrated Environment** is similar to the concept of Integrated Development Environments (IDE). In other words, the simulator is capable of providing a visual environment for the actual implementation of additional, or new functions, to the simulator without the need to exit the simulator, write code, recompile the code and start it up again.

The key point in the provision of this taxonomy is that each and every simulator, regardless of purpose or choice of programming language, will consist of three main components: presence of time, basis of value, and behavior. However, not all simulators will consist of a User Interface. The choice of whether an interface forms part of the simulator will be dependent on the goal of simulation. For example, in the case of simulating surgical procedures in a hospital for training purposes, visualisation is most likely a necessity whilst simulation of all the numbers in $\pi$ will not likely require any form of graphical visualisation since the output of raw numbers would suffice.

### 2.3.5 Challenges in Successful Simulation

Some of the key challenges in conducting a successful simulation are analysed. The work of Law [48] states that there are seven stages which must be carried out in the successful implementation of a simulator including formulating the problem, gathering of data, validating the model, programming the model, validating the programmed model, conduct and analyse experiments and document and present results. The term "model" which is used in this section refers to the entity which the simulator is attempting to simulate. A description on each of these seven steps, along with additional literature backing, follows:
1. **Formulation of Problem** - This first step involves defining what, exactly, the aim of the simulation is. One could argue that the formulation of a problem is not the challenge itself but rather it is defining how the simulation aims to solve the problem. Hence, this first step is of great importance in providing a successful simulation since it provides the overall objectives of the simulation.

2. **Data Collection** - The second step to simulation is in conducting research to gather data which can be used as a baseline for the simulation. In other words, this involves gathering information which the design and implementation of the simulator will be based upon. It has been stated that data collection is perhaps one of the most time-consuming aspects of conducting a simulation [49] with one source stating that it takes up to 40 percent of a project’s time [50]. Though time-consuming, data collection is fundamental as it is from the data gathered that a simulator will be modeled against. Thus, it is important to ensure the data gathered is accurate since it will influence the simulators results [51].

3. **Validation of Conceptual Model** - From the data gathered, a conceptual, i.e theoretical, model should be presented and validated to ensure it meets the aims of the simulation (as defined in step one). Validation, in this context, refers to ensuring that right model is being built [52]. In other words, it is ensuring that the model presented is a accurate representation of the object(s) or system(s) which the simulator is simulating.

4. **Implementation of Model** - Having validated and presented a accurate model of the simulation, this step involves the actual creation of the simulation. The choice of either using a programming language or existing simulation software is entirely up to the developer’s own preference.

5. **Validation of Implementation** - Similar to step 3, upon successful implementation of the simulator, it is important to validate that the actual implementation is a valid representation of the model the simulator is attempting to simulate.

6. **Experiment and Analysis** - Having presented a valid implementation, experiments can then be conducted and results analysed. The experiments carried out will relate back to what the initial aim of the simulation (as stated in Step 1) is. Having conducted the experiments, the results can be analysed and, at this point, it can be stated whether or not the simulation was successful in solving the problem.

7. **Presentation of Results** - The final and perhaps simplest step, assuming all previous steps have been carried out correctly, the results of simulation can be documented and presented.

It is identified that the steps presented is similar to the iterative process conducted in a system development life cycles, e.g analysis, design, testing, implementation, testing etc., and although carrying out simulation is a challenging task, breaking down the challenge into smaller steps makes the task much more achievable.

### 2.4 Conclusion

The first part of this background has presented a general overview on e-Health. A definition on key terms have been provided along with the history of the health care...
systems used. The many advantages of e-Health have been provided along with justification as to why most implementations of e-Health are beginning to use cloud computing as their main platform of deployment.

Both technical and organisational concerns in the deployment of e-Health has been discussed. From the three organisational barriers which have been presented, one could justify that implementation of e-Health solutions is not only a technical challenge but also challenging in terms of a business perspective. As part of this research, it is proposed that the concept of a patient simulator not only helps developers of e-Health solutions test and validate their work but also allows healthcare organisations to evaluate, at an early stage, whether or not the proposed e-Health solution is capable of overcoming the legal, operational and cost/benefit barriers.

In the second part of this chapter, the theory of simulation has been outlined. It has been shown that several key challenges need to be overcome in conducting successful simulation and furthermore, there are various choices of techniques available in the implementation of a simulator as demonstrated in the taxonomy of this subject. Having provided the background and theory behind these two disciplines, the next chapter presents a review on actual implementations and methodologies applied to this two areas of work.
Chapter 3

Literature Review

3.1 Introduction

A n examination on recent literature in regards to e-Health environments is conducted in this chapter. The review highlights that security in the implementation of e-Health environments is a key concern.

In the second part of this review, the techniques and validation methods applied to computer based simulation is assessed. The purpose of this part of the review is to show the different techniques used in simulation and provide backing for the choice of technique used in the patient simulator framework which is presented in Section 4.

To provide support on this researches view that the implementation of a patient simulator allows evaluation of e-Health environment, mitigating the need for real patient data, the last part of this chapter provides a thorough review of existing simulation methods applied towards a healthcare environment and justification is provided as to how they differ from the research proposed in this report.

3.2 E-Health Review

In the literature review conducted on e-Health infrastructures, a common theme found was that proposed designs and implementations were either catered primarily towards individual users (e.g. patients) of medical systems or catered towards healthcare organisations as a whole. Hence, in order to provide for a more structural approach in the review of e-Health infrastructures, it is proposed in this report that e-Health infrastructures be categorised into two main groups: organisational-centric e-Health and user-centric e-Health.

A review of four of the most recent e-Health infrastructures are provided in this report. Section 3.2.1 and Section 3.2.2 review two main works which are aimed towards organisations of healthcare whilst Section 3.3 and Section 3.4 provide e-Health infrastructures which are more user-centric.

3.2.1 Sensor Based

Rolim et al. [2] propose that, through the use of sensors attached to pre-existing medical equipment (Figure 3.1), medical data of a patient can be automatically exchanged
(ie. stored or retrieved) using cloud services. Furthermore, this proposed solution is not limited to only sensors, as simple workstations and mobile devices can interact with the cloud network via the exchange service. Using standardised protocols, including Hypertext Transfer Protocol (HTTP), Extensible Markup Language (XML) and Wireless Markup Language (WML), means that a high level of interoperability [2] can be achieved since nearly all computer equipment should be capable of understanding these exchange messages.

The main advantage proposed in the approach by the authors of this work is that it eliminates the need for medical personnel to conduct collection and input of patient data. Although it could be argued that EHRs already provide similar functions, the novelty in this work is that data collection is automated and in real-time thus, it helps prevent typing errors and the "manual" aspect of data collection.

![Figure 3.1: Sensor Based e-Health Service [2]](image)

Although the use of sensors allows for a very simple method in the exchange of data between hospitals and the cloud network, there is a fundamental limitation found in the work of Rolim et al. which is the lack of addressing the security aspect on their system. It has been argued that security is the most important attributes when it comes to healthcare environments [53], not only in cloud networks, but within the scope of computer informatics as a whole. Further backing can be provided in the work of Roelfzema, in which thorough investigation into risk assessment shows that healthcare environments still do not have a comprehensive framework for mitigating security breaches in their IT systems [54]. Thus, Rolim et al. present a novel technique in the capture and storage of patient data but concerns remain over the secure storage of such data.

### 3.2.2 DACAR Project

In the work conducted by the Data Capture and Auto Identification Reference (DACAR) project [55], one of the key themes discussed is in the provision of a secure e-Health infrastructure. The aim of DACAR is to develop, implement, validate and dissemi-
generate a novel, secure e-Health platform for capture, storage and consumption of data within a health care domain. The key components of the DACAR platform include:

- **Single Point of Contact (SPoC)** - An authorisation gateway for a health care domain to grant role-based access rights to sensitive medical data and services using a policy syntax.

- **Information Sharing Policy Syntax** - This is a rule-based language syntax inspired by firewall rules. It can be used to define access rights and to express a variety of patient consents.

- **Data Buckets** - The Data Buckets offer long-term persistence of medical data and support the Creation, Reading, Updating and Deletion (CRUD) of attribute values and associated meta data.

Figure 3.2 gives an overview of the DACAR platform. Typically, a user consumes an e-Health service developed on the DACAR platform in five steps:

1. **Authentication**: The user logs on from federated identity providers using a user name and a password, or other unique personal information.

2. **Request for a service**: The user’s client software forwards the security credential obtained in Step 1 to a responsible SPoC, together with a service request.

3. **Instantiate the service**: The SPoC checks the user’s identity, resolves it into a role, and matches the service request to existing security policies. In the case that the service is provided by the local domain, the SPoC is able to tell whether the user is allowed to consume this service, and to locate the service endpoint within the Cloud. However, if the service is provided by a trustworthy foreign domain, the SPoC will route the service request to another SPoC. For example, when a clinician needs to make contact with a patient’s relatives in an emergency, he or she sends a request for a police registry service to the local health care SPoC, which forwards the request to a remote police SPoC.

4. **Authorisation**: If the service request is permitted by corresponding security policies, the SPoC that made the decision creates and signs a Service Ticket.
This contains the user’s pseudonym and role, a reference to the service endpoint, period of validity, and one-off session keys that enable the user’s client software and a service instance to establish a secure session. Otherwise, a message is returned to tell the reason for rejection.

5. **Consume the service**: Finally, the user’s client software initiates a secure session using the information provided in the Service Ticket and starts to consume the service. If the service requires CRUD operations over certain attributes, the service itself becomes a consumer of related Data Bucket services. In this case, the service needs to go through Steps 1 to 4 to obtain necessary Data Tickets from a SPoC using the service’s own identity, or the service consumer’s identity and role. In the latter circumstance the service is “impersonated”, and shall use the Service Ticket received from its consumer as a complementary security credential.

Although this five step process in provides a secure environment for medical data, the limitation in this work is attempting to validate such claims. As described earlier, the DACAR implementation allows for the capture, storage and retrieval of data in a highly secured cloud environment but the solution of evaluating DACAR, and similar e-Health platforms, are still lacking.

### 3.2.3 Microsoft HealthVault

Microsoft HealthVault [3] is a cloud based platform used for the the storage and manipulation of Personal Health Records (PHR). In essence, Microsoft HealthVault allows patients to update and monitor their medical information using the online service which can then be shared with medical professionals and applications [56]. Medical professionals, and applications, can then use the data which the patient has uploaded to provide reminders (such as the next date for an appointment) or general information on the patient’s current health status. Figure 3.3 provides a overview on some of the applications and persons which a patient is capable of sharing their medical data with. The decision over whom to share their medical data with is entirely under the control of the patient.

In addition to allowing patients to update their medical information, the Microsoft HealthVault platform has provided their open source API’s and a Software Development Kit (SDK) to allow for additional applications and features to be integrated with HealthVault. This has allowed for the ability to use real-life medical devices such as heart rate monitors and weight scales to connect directly with HealthVault and upload data dynamically as a patient takes their own measurements [57].

When it comes to storage of patient data, ensuring the laws in regards to data privacy are being followed correctly is always a key challenge (discussed in Section 2.2.5). HealthVault addresses this concern by stating that it is entirely the patient’s own choice in releasing personal data to their service [58]. However, though HealthVault’s claims are valid, concerns are still being raised in regards to the way they store patient data. The remarks from Haas et al. [59] indicate that although HealthVault follows the laws of privacy of data, the authors expresses concerns since all data is stored server-side meaning access controls, i.e. encryption of data, generally remain one sided with patients having no control over how their data is "secured”.

The aspect of patient’s having little control over how their data is "secured” can be further highlighted from recent controversies invoked by the social networking website Facebook [60]. In 2006, Facebook implemented a "News Feed" feature which
broadcasted posts submitted to a user’s personal profile in a widely public manner [61]. Similar to HealthVault, it was the user’s personal choice in releasing their private data to Facebook but the assumption was that in the case of submitting data to their personal profile, this data would have been kept private (and only available to be viewed by others based on the user’s permission) rather than being broadcast publicly.

### 3.2.4 Google Health

Google Health is another implementation of a cloud platform for PHRs. A few comparisons may be drawn between Google Health [4] and Microsoft HealthVault. Google Health, similar to HealthVault, allows for users to upload and manage their medical data using an online cloud platform (Figure 3.4). Furthermore, the capability to share data between patient and other services is featured, with control over the data being given to the patient it belongs to. Finally, like HealthVault, Google has published their API and SDK as open source, allowing for future integration capabilities with other applications [62].

However, though similar to HealthVault, Google explicitly states that Google Health is not simply a PHR platform but "a bit of a different mode" with abilities to "to set personal health and wellness goals and track your progress over time or interact with personalized applications and content" [63]. In other words, whilst Microsoft HealthVault focuses on interoperability between a PHR cloud platform and sharing of data with other users and devices, Google Health has taken the approach of providing a more user-centric experience, one in which is specifically aimed at individual patients.

Unfortunately, though the aims of Google Health differ from HealthVault, they share the same limitation in the aspect of privacy of data. Google Health uses the exact same model for it’s policy [64] on the storage of personal data, it is up to the individual to choose what aspects of their health care records they wish to submit to Google. By storing patient data server-side, users have no specific control over how "secure" their data.
3.2.5 Summary

Four e-Health infrastructures have been presented in this literature review. The key theme throughout this part of the review has been concerns over security. The sensor based patient data collection scheme proposed by Rolim et al. does not address how security is integrated in their work whilst the DACAR project proposes a e-Health platform using policies and gateways for a secured environment as one of its fundamental considerations. Although DACAR’s implementation may be technically sound, it is still a key challenge to validate that their implementation is secure when a huge number of patient data may interact with this platform in a real-life clinical environment.

Another issue, in the provision of organisation-centric e-Health solutions, is highlighted in the work of Pagliari [65]. One of the key challenges, the author states, is in achieving trans-disciplinary work between technical minded developers and healthcare professionals [65]. In particular, Pagliari addresses the issue that e-Health implementations which, though technically sound, may not end up meeting the design requirements of healthcare organisations thus may be rejected. Hence, early evaluation of such e-Health environments is an essential necessity, not only in validating concerns including efficiency, reliability, security and scalability but also demonstrating the functionalities of the service to healthcare organisations so that design requirements can be agreed upon.

Hence, the concept of simulating patient’s for testing and validating e-Health can be reinforced at this point. By providing simulated patients, it is proposed early evaluation of e-Health platforms, such as DACAR, can take place and validation of security can be made. Furthermore, as highlighted by Pagliari, developers of e-Health infrastructures may not always share the same views as healthcare professionals on what the requirements of e-Health should be. The patient simulator can simulate interaction between e-Health infrastructures during preliminary prototyping so that a working, live demo can be presented to both parties and design requirements agreed...
CHAPTER 3. LITERATURE REVIEW

upon, or altered if necessary.

Both PHR platforms proposed by Microsoft HealthVault and Google Health are aimed for use by individual users rather than healthcare organisations. Both share similar goals but, unfortunately, face the same limitation. In both implementations, patient data is stored server-side therefore, a level of trust must be given to these companies by users who choose to use their service. It is acknowledge that the patient simulator proposed in this research work cannot provide validation on whether or not users should trust storing their data on these services however, it is proposed that companies of such services may take advantage of simulated patients in order to evaluate their implementations to ensure that security and privacy laws are respected.

3.3 Simulation Methods

A review of the techniques of simulation is conducted in this part of the report. The taxonomy of simulation (Section 2.3.4) shows that three main components are required in the implementation of a simulator: presence of time, basis of value and behavior. Presence of time is considered a binary choice, either a simulator uses a time keeping function or it does not.

In the case of a presence of time being made available in a simulator, the basis of values are generated using one of two main techniques: discrete-event or continuous-event based simulation. Furthermore, regardless of time being a attribute of simulation, the method employed to simulate entities, e.g. variables, will come under one of two methods: deterministic or probability based behavior.

3.3.1 Discrete-Event vs Continuous-Event Based Simulation

In applying a dynamic presence of time property, implementations of simulations using discrete simulation is commonly refered to as Discrete-Event Based Simulation (DES) whilst continuous simulation is referred to as Continuous-Event Based Simulation (CES) [66].

The "event" refers to a simulation process taking place at a dynamic point in time which can either be specific time intervals (in the case of discrete simulation) or continuous time intervals (in the case of continuous simulation).

For example, consider a simulation which has a total duration of 5 seconds. Employing a DES method, an arbitrary timer interval of 1 second may be specified. Hence, at every 1 second interval, some form of processing, e.g. generation of a variable, in the simulation will be conducted for a total of five times, since the duration of simulation is 5 seconds. In comparison, employing a CES method will result in simulation processes being carried out continuously throughout the duration of simulation. Therefore, at time 1 second a process may take place, and another process may take place at time 1.1 seconds, another process at 1.2 seconds etc. until the total duration of 5 seconds of simulation has been conducted. Figure 3.5 provides a comparison of these two techniques.

Neither method of simulation is better or worse off. Since DES models variables which only change at specific points in time, it is well suited to simulating the events which take place at specific time intervals. CES will model events continuously, therefore this method is better suited to simulating systems which will have constant changes [67, 68].
An example of a DES implementation can be found in the network simulator known as Ns-2 [69]. Ns-2 provides simulations of networks, including TCP, routing and multicast communication for research purposes. Ns-2 is defined as a DES [70] method since it only simulates network communication at specific intervals in time. In comparison, a stock market simulation using CES as the implementation basis is presented by Muchnik et al. [71]. The justification for using CES is that stock prices will change, either going up or down, in a continuous manner with no specific time intervals in which changes are conducted unlike the Ns-2 simulator. In other words, the stock market prices will rise or fall continuously throughout simulation.

Although simulation for a certain goal can be achieved using either DES or CES, the results of the experiments may differ drastically. As demonstrated in one study, both DES and CES methods were applied in simulating vehicle (car) interactions by Jamison and McCartney [72]. In this study, it was found that widely different behaviors in the vehicles were produced. In particular, using DES shows that cars would behave in a chaotic manner, especially with increments in time interval whilst CES resulted in cars behaving in a slightly more fluid manner. Hence, although neither method is better or worse, it can be stated that the choice of either DES or CES must be considered in great detail since the results may be skewed in a favorable or negative
direction based on the method chosen.

### 3.3.2 Deterministic vs Probabilistic Behavior

The calculation and output of results in a simulation can be considered either deterministic or probabilistic. It should be made clear that both DES and CES methods may be employed in producing deterministic or probabilistic calculations.

To provide definition, in deterministic behavior, given one or more input parameters, calculation will result in either one, or a set number, of possible outcomes [73]. As an example, given the inputs $a$ or $b$, input $a$ will also result in output $x$ whilst input $b$ will always result in output $y$ (Figure 3.6). Hence, repeatability and predictability of results [74] is the key attribute in deterministic simulation.

![Figure 3.6: Deterministic Behavior Example](image)

In comparison, the opposite is true of probabilistic behavior. Probabilistic, also referred to as stochastic, behavior introduces the concept of “randomness” in simulation [74]. Using a similar example as deterministic behavior, given the input $a$, probabilistic results may return $x$ with only 40% certainty or it may return result $y$ with 60% certainty (Figure 3.7). Hence, there is a certain element of unpredictability in probabilistic behavior.

![Figure 3.7: Probabilistic Behavior Example](image)

Probabilistic behavior is well suited to simulation where elements of uncertainty are required. As an example, a simulation of a slot machine (found in casinos) will require the use of probabilistic behavior since there must be a degree of randomness.
in order to ensure users do not attempt to cheat the system. In comparison, a sim-
ulation of a factory production line may employ deterministic behavior since, based
on the inputs of the simulation, it can be certain what the output produced by the
production line may be.

In the case of implementing probabilistic behavior, some form of random generation
technique must be applied in order to provide the "randomness" of the simulation.
One of the most well known techniques used is the Monte Carlo Method previously
mentioned in Section 2.3.2. In essence, this technique enables the use of repeated ran-
dom sampling of numbers in order to come to a result [75]. It should be noted that
there is no one singular Monte Carlo algorithm but instead, it is a group of mathe-
matical algorithms.

3.3.3 Summary

The techniques of DES and CES have been presented in this section of the literature
review. Furthermore, simulation behavior including deterministic and probabilistic
behavior has been reviewed. It can be summarised that none of these techniques
have overwhelming advantages over one another. Instead, the choice of technique
used in simulation will depend entirely on the goal of simulation. However, as
demonstrated in the work of [72], careful consideration must still be made in the
choice of simulation techniques used since the results of simulation may be biased
based on the implementation technique used.

3.4 Validation of Simulation

In the background section of this report, an outline of key challenges in conducting
simulation was presented (Section 2.3.5). A important theme which was established
in overcoming the challenges of simulation was the concept of repeated validation
of both the design and implementation of simulators. In this part of the literature
review, a concise study on the techniques used in the validation of simulators is pre-
sented.

Although it has been argued by Sterman that the concept of validating a simulator
is impossible and focus should be placed in model testing to ensure the simulator
is appropriate for its’ purpose [76], this author’s view point is not shared in this
report. Based on the review conducted, many techniques have been proposed for the
validation of simulators including [52, 77, 78].

3.4.1 Qualitative or Quantitative?

Each area of work may be grouped into either quantitative validation, qualitative val-
idation or a combination of both. To provide definition, quantitative research aims to
validate results using absolute data, i.e numeric values, whilst qualitative validation
is generally more subjective whereby the use of words, texts and observations are
used to describe the evaluation process [79]. To provide clarity, the quantitative and
qualitative techniques are used to validate simulation models and implementations
in this section of work reviewed. Simulation is not used to validate quantitative or
qualitative properties.
3.4.2 Qualitative Validation

A prime example on qualitative validation of simulation is in the work conducted by Viet et al. [80] in which they compare stochastic simulation of bovine viral-diarrhoea virus (BVDV) with real life data sets. Their overall goal was in conducting a validation of the spread of the BVDV from previously gathered data in comparison with the stochastic simulation of BVDV which they developed.

Whilst the results produced from this study validate that the simulation method is capable of mimicking real life data on the spread of BVDV, it is acknowledged by the authors that some of the simulated data was not presented because they were “inconsistent with knowledge on BVDV transmission and the simulated demography of dairy herds” and “no observed data were available for comparison” [80]. Hence, it highlights one limitation in using qualitative approach as a primary means of validation. Without sufficient data to conduct a basis of comparison, qualitative validation - by itself - results in studies which may end up objectively bias since comparison of all simulated data with real life data is not always possible.

3.4.3 Quantitative Validation

In comparison with the work of Viet et al., a quantitative validation of user simulation techniques for spoken dialogue systems is presented in the work of Schatzmann et al [73]. In this research, stochastic modeling is once again used, this time to simulate users which are capable of simulating a conversation using a Spoken Dialogue System (SPD). A number of different dialogue simulation techniques are used and in order to validate their success, Schatzmann et al. first used training datasets (of spoken dialogue) to train their simulation system in dialogue. Experiments are then conducted whereby dialogue "interaction" between the simulated user and a dialogue manager (DM) is carried out.

Naturally, one may assume that since this work attempts to validate dialogue produced by simulated users (text and speech being a qualitative property) it would fall under the qualitative validation. However, the novelty in Schatzmann et al. work is that they define a numeric scoring system based on the response produced by the simulate users hence, a quantitative validation can be achieved since there is only a number of set possibilities which the simulated user can respond to dialogue with, e.g. a yes or no answer. Based on the accuracy of response of the simulated user in any given dialogue, a score is assigned and results can be presented in a quantitative manner.

3.4.4 Mixed Validation

Using a mixture of both qualitative and quantitative measurements, the work of Balci [81] proposes that validation of simulators requires using techniques that range from informal, such as subjective reviews and inspections of the simulator (qualitative validation), to formal approaches using statistical techniques such as logical deduction and lambda calculus (quantitative validation). The whole range of techniques are described in the author’s taxonomy and include Informal, Static, Dynamic, Symbolic, Constraint and Formal validation. A description of each technique follows:
• **Informal** - Primarily uses human reasoning and observations rather than formal mathematical approaches (e.g. conducting reviews, inspections and audits of the simulation model).

• **Static** - Involves static analysis of the model and source code of simulation in terms of accuracy (e.g. consistency checking, syntax analysis and structural analysis of code).

• **Dynamic** - Execution of simulation code and validation based on behavior (e.g. Black-Box Testing, Debugging, Stress Testing).

• **Symbolic** - Similar to dynamic validation but use of symbolic input, i.e. mathematical equations, and analysis of output expressions. This technique is used to understand the path taken by the code during execution. (e.g. cause effect graphing, path analysis, partition analysis).

• **Constraint** - Used to check the simulations correctness by investigating the boundaries of the simulation (e.g. boundary checking, assertion checking).

• **Formal** - Conduct qualitative validation using mathematical techniques to proof that the simulation is correct (e.g. predicate calculus, proof of correctness).

### 3.4.5 Summary

This section of the literature review presents qualitative, quantitative and a mixed approach in the validation of simulators. It may be concluded from this portion of the review that there is no one correct method in the validation of simulators. It is perhaps best stated by Balci that there is no singular metric for the validation of simulators; only repeated testing and verification throughout the development of the simulator, using all approaches deemed necessary, results in a successful validation [78]. Backing is provided by Martis in which it is stated that "no single procedure can suit the [validation of] of all models" [82] whilst further support is provided by Schatzmann et al. in which, from their work, they conclude that "the nature of the simulation problem is such that no single measure of goodness exists"[73].

### 3.5 Healthcare Simulators

A vast amount of simulators are currently available, both commercially and for purposes of research. It is not possible to cover every simulator in existence, hence, in the scope of this report, a focus on simulators which are related to healthcare is reviewed. This section, in particular, aims to present approaches simulators have been used to enhance healthcare environments and also justification that, though the techniques employed are novel, they differ from the work of this research which proposes a patient simulator for evaluating e-Health environments.

#### 3.5.1 Human Patient Simulator

Laerdal presents SimMan®, a full-scale robotic mannequin [5], otherwise known as a Human Patient Simulator (HPS), which is capable of simulating the physical attributes of a patient. Dedicated software, which can either be run on a personal
computer or a replicated patient monitor, allows for the simulation of vital physiological signs of the mannequin. Figure 3.8 shows demonstrates the mannequin and replicated patient monitor used in the SimMan implementation.

Figure 3.8: SimMan®Robotic Mannequin [5]

In evaluations conducted on SimMan®, Hesselfeldt et al. [83] concluded that the simulated airway of the mannequin, though acceptable, was lacking in realism due to the mannequin having anatomic insufficiencies in comparison with a real life human being. In regards to the generation of vital signs, the studies carried out by Wyatt et al. [84] found that 36 out of 54 features of SimMan were rated at least average physiologically accurate by health professionals. Thus, it can be stated that SimMan® has generally faced favourable reviews in the simulation of a patient.

In related works, Hwang et al. [85] proposes the integration of both HPS and Virtual Patients (VP) to provide a physical simulation of a patient and a virtual simulation of the clinical environment. The HPS component of their work is a mannequin which models the clinical signs, e.g. heart rate, blood pressure, body temperature, of a patient using scripts whilst the VP component acts as an interactive clinical environment allowing users to "control" the mannequin with predefined commands. One novel feature of this work is the ability of the HPS to react to speech commands via the VP system. This feature is achieved using what the authors describe as the *HPS Internal Data Exchange Protocol* (IHDEP) (Figure 3.9), a Java based communication protocol, which enables two way communication between the virtual aspect and the physical (the mannequin) aspect of the simulated patient.

![Figure 3.9: HPS and VP Integration using HIDEP exchange](image)

The work of both Laerdal and Hwang et al. present novel features in the implementation of simulated patients. However, both works conducted relate to the implementation of HPSs whereby the main aim of such simulation is for the purpose of educating medical personnel for training purposes by providing a physical representation of a human body [86]. The work of Hwang et al. propose the use of VPs but the purpose of this component is for providing a script based approach in the "control" of
CHAPTER 3. LITERATURE REVIEW

their mannequin. The research proposed in this report differs from the aims of both works. Rather than provide a HPS catered towards the educational environment, this research provides a completely virtual (non-physical) representation of patient and the data they generate aimed towards evaluating e-Health environments.

3.5.2 Traffic Flow Simulator

Using the technique of discrete-event based simulation (Section 2.3.4) Meng et al. [87] propose the modeling of emergency hospital environments, i.e. Accident & Emergency, in order to understand traffic flow of patients and provide solutions to overcome overcrowding issues. Using a number of predefined scenarios, e.g. increment or decrement waiting times to see a consultant, and modeling patient arrival times using predefined schedules along with stochastic (random) arrivals, their simulation is capable of determining the time periods in which increase in waiting times occurred and factors which contributed to this trend.

In order to validate their simulation, Meng et al. compared the simulated patient’s, both scheduled and stochastic, arrival times with the data gathered from past medical records. As no formal metrics is established for accuracy, seen in works such as [73], the results presented are mainly observational based hence a qualitative validation.

In similar work, Kolb et al. [88] propose five patient buffering concepts in order to reduce the traffic flow and overcrowding in emergency departments. Once again discrete-event based simulation was the technique chosen to test their five buffering techniques. The results of this work state that all five buffering techniques improve upon the traffic flow of emergency departments to some degree. In terms of validation, the authors claim to follow the validation principles of work discussed previously including [66, 78, 52] but, unfortunately, no specific methodologies were discussed.

It can be stated that discrete-event based simulation is widely applicable in healthcare environments. The two areas of work covered in this section both emphasise the importance of validating their simulators. Although the patient simulator proposed in this research differs from the aims of these two pieces of work (since this research aims to simulate virtual patients and the data they generate rather than the flow of traffic) a key point which has been noted is the importance of providing the methodology behind the validation process of simulators.

3.5.3 Virtual World Simulator

The virtual reality world Second Life [6] has been a domain in which healthcare simulation systems have been popular. Second Life is a 3D on-line virtual world which allows users to interact with each other using avatars (Figure 3.10). In essence, the avatar is simply a virtual representation of the user within the virtual world of Second Life. Various health care simulation projects have taken place in this environment.

The work of Beard et al. [89] identify five main categories of health care simulations found in this virtual world including education and awareness, support, training, marketing and promotion of health services and research. Table 3.1 gives an overview on these five main categories.

As Second Life is, in essence, a user-driven interactive world with the main goal of providing entertainment, it was found that most health care simulation was simply
to provide users with information rather than employing any techniques to simulate actual patients. Hence, though work conducted in Second Life presents a novel method in engaging people in the field of health care, the aims and objectives of virtual healthcare simulation in Second Life differs completely from the concept of attempting to simulate virtual patients for a e-Health environment.

3.5.4 Vital Physiological Sign Simulator

With focus on work towards vital physiological sign simulation, Agar et al. [7] presents a simulation system which is capable of simulating blood glucose and insulin levels of both healthy patients and patients with Type 1 diabetes for educational and training purposes. The model they present is focused on the physiological compartments which relate specifically to blood glucose and insulin levels including the heart, brain, liver and kidney as examples.
The key novelty in their work is that the simulator allows for the input of parameters (Figure 3.11) including time of meals, carbohydrate intake, insulin dosage, patient’s weight etc. and by running the simulation, correlation between input parameters and output of glucose and insulin concentrations in blood and liver can be observed.

![Figure 3.11: Input Parameters for Type 1 Diabetes Simulation](image)

Whilst Agar et al. focus on simulation of one specific vital physiological disease, the research proposed in this research aims to simulate not only the vital physiological signs of a patient, but also other key attributes of a patient which would be normally be stored by healthcare environments. Furthermore, the work of this research differs from the goals of Agar et al. since the aim of the patient simulator is in the validation and testing of e-Health infrastructures rather than provide education or training. However, Agar et al. work is able to complement this field of research since it outlines the correlation between vital physiological signs which is one aspect which the patient simulator proposed will need to take into consideration further on in the research and development process.

### 3.5.5 Summary

A key differentiation between HPS and VP has been made. A HPS is a physical entity, i.e. mannequin, whilst VP is a completely virtual representation of a patient. In reviewing the five different categories of simulation conducted in healthcare environments, it has been highlighted that currently, no form of simulation exists which is catered specifically towards evaluating e-Health environments. Furthermore, the goals of the simulators reviewed is almost always catered towards a training and educational environment.
3.6 Conclusion

This chapter begins by conducting a review of four e-Health infrastructures. It has been highlighted that the key challenge which must be overcome is ensuring security of medical data, especially in regards to privacy of patient data [34, 35]. A secondary challenge in the provision of e-Health environments, including both platforms and services, is ensuring the design and implementation meet the needs and requirements of healthcare organisations [65]. It has been proposed that in conducting research and development into patient simulation, such a technique will enable early testing and validation of the security concerns of these infrastructures. Furthermore, it enables live demonstration of e-Health environments to show case to health organisations and ensure the design and implementation meets their expectations.

Having shown the key issues surround e-Health environments, and how this researches aim of providing a patient simulator helps alleviate these issues, the second part of the literature review conducts a concise analysis on the key techniques towards simulation. The core theme established is that regardless of the simulation technique chosen, e.g. DES or CES with Deterministic or Probability behavior, the most important aspect of providing a good simulation is in validating both the design and implementation in an empirical manner. By reviewing both quantitative and qualitative validation techniques, further understanding can be gained in the methodologies and approaches which have been taken to validate simulators. As stated by [78], there is simple method in validating a simulator but only through concise and thorough evaluation using both quantitative and qualitative techniques can this aim be achieved.

In the last part of this chapter, a review was conducted on healthcare simulators. It can be concluded from this review that existing work, in regards to healthcare simulation, is mainly driven towards the goal of providing education and training. This is especially demonstrated in works such as SimMan®, and the Type 1 Diabetes Simulator provided by Agar et al. Although many novel ideas are established in the healthcare simulators, not one single simulator was found which is catered towards evaluating e-Health infrastructures thus, the novelty is the research which is proposed in this report can be reinforced at this point.

In reviewing the current trends of e-Health environments, the techniques employed in implementing and validating simulators and learning some of the novel approaches taken in carrying out simulation in a healthcare environment, the framework of the proposed patient simulator, based on the knowledge gained in this literature review, can be presented. The chapter which follows outlines the patient simulator framework.
Chapter 4

Proposed Patient Simulator Framework

4.1 Introduction

This chapter presents the framework which is capable of simulating patient and their data for the evaluation of e-Health environments. It is proposed that this framework is capable of evaluating e-Health environments under the metrics of efficiency, reliability, security and scalability. A four tiered approach has been taken in defining the patient simulator framework. The four components which make up the simulator include: user interface, patient models, simulation engine and communication engine. An abstract view of the framework is presented in Figure 4.1. A discussion on each of the four components of this framework is presented in this chapter.

4.1.1 User Interface

As highlighted by [1], the user interface of a simulation provides a far easier means of running a simulator in comparison with a command line driven interface. It is acknowledged that this part of the framework is not an essential necessity in conducting research. However, as demonstrated in previous research [90], a user interface can bring about ease-of-use, automation and simplicity in conducting testing and validation tasks. In other words, it makes running the application a lot simpler.

4.1.2 Patient Models

The secondary component of the patient simulator is the abstract models of a patient. Taking a object-orient programming approach, it is proposed that the patient simulator is capable of simulating patients and their data based on modeled templates of a patient. Such modeled templates can be "extracted" during the running of the simulator and new instances of patients can be created.

The current ontology of a patient consists of two main categories: non-medical attributes and medical attributes. Within the subclass of medical attributes, these may be further split into dynamic medical attributes and static medical attributes. By defining a patient under two main categories, this model enables a "plug-and-play"
architecture which allows for the ability to easily add further attributes so long as they fall under either non-medical attribute or medical attribute. Figure 4.2 shows the current model of a patient.

A description on the categories and attributes is provided as follows:

**Non-medical attributes** refer to attributes which, though important, do not have significance when applied to a health care environment. In other words, medical staff will not take these attributes into consideration when it comes to diagnosing a patient’s health. Examples of non-medical attributes are:

- **Patient Identification** - A unique ID which is given to each patient
- **Forename(s)** - The first given name of a patient along with any middle names
- **Surname** - The last given name of a patient
- **Home Address** - Patient’s address of residence

**Medical attributes** are split into two subcategories: static and dynamic attributes. Static attributes are defined as such since they will not generally change regardless of a patient’s status. Examples of the static medical attributes are as follows:

- **Blood Type** - Blood type of patient (i.e. A, B, AB or O)
Finally, dynamic medical attributes relate to a patient’s vital physiological signs, which have the characteristics of discrete change throughout a patient’s stay in a health care facility. Examples of the dynamic medical attributes include:

- **HR** - The heart rate of a patient, measured in the unit of beats per minute (BPM)
- **BP** - The blood pressure of a patient, measured in mmHg (millimeters of mercury)
- **Temp** - Temperature of a patient’s body, measured in degree celcius (°C)
- **SpO2** - Oxygen level of a patient, measured in percentage (%)
- **RR** - Respiratory rate of patient, measured in Breathing Frequency (BF)

The dynamic medical attributes which have been chosen to simulate in a patient is based upon the concept of the Early Warning Score (EWS) system currently used in hospitals. To provide a brief background, the EWS was originally developed by Morgan et al. [91] and is a risk based scoring system used by nurses and other healthcare staff in rating a patient’s health status (e.g. in Accident and Emergency departments). In essence, a higher score suggests a greater risk to a patient’s life. Traditionally, the EWS system has been applied as a paper-based observation chart whereby the “risk” is calculated manually by a member of hospital staff.

Several attributes are taken into consideration when calculating the risk of a patient. Generally, vital physiological sign parameters including heart rate, blood pressure, body temperature, respiratory rate and oxygen levels are observed [92]. Medical staff who carry out observation of the patient will assign a numeric value for each parameter and all values calculated together gives the patient’s risk score.

By modeling the dynamic medical attributes on the vital physiological signs which are observed in the EWS, this allows for the patient simulator to cover one of the
key groups of attributes of patient data. Furthermore, early work has already been proposed to migrate the EWS system to an computer based e-Health service by [17] (Section 5.2.1 details the work conducted on this topic in greater detail). Hence, in implementing the patient simulator which is capable of simulating vital physiological signs, it is proposed testing and validation of early prototypes of the e-Health based EWS system can be conducted to ensure the systems accuracy in providing a risk score to patients.

4.1.3 Simulation Engine

The simulation engine forms the core aspect of the patient simulator framework. It is within this component that the actual simulation process takes place, including simulation of both medical and non-medical attributes of a patient. In this framework, it is proposed that the use of Discrete-Event Based simulation along with mixture of both probabilistic and deterministic behavior. The simulation engine will manage the time intervals during simulation along with producing output values during simulation.

As the literature review demonstrated, many existing healthcare simulations already use DES as part of their simulation [87, 88]. To justify the choice of using DES for this framework, it can be stated that interactions in a real life healthcare environment do not occur in a continuous basis. For example, in the case of a nurse taking a temperature of a patient, this task will only ever be conducted at certain time intervals. Similar to manual methods, in using e-Health infrastructures, the temperatures of a real-life patient will only be read at certain time intervals before being uploaded to a storage system. Hence, using DES method allows for the simulation of such interactions as would be seen in real life.

As part of the simulation engine, a choice of both probabilistic and deterministic behavior outputs has been chosen. In the simulation of static and non-medical attributes, deterministic behavior of output is simulated. However, in the case of dynamic medical attributes (vital physiological signs), probabilistic behavior is chosen. Vital physiological signs are, naturally, dynamic hence there is a degree of variability in a patients vital signs at any time interval. Therefore, using probabilistic behavior, in the simulation of vital signs, enables more realistic simulation of these attributes. Section 5.4.1 discusses the work conducted in regards to generating vital physiological signs in greater depth.

4.1.4 Communication Interface

The final component of the propose patient simulator framework is the communication interface. It is proposed this part of the framework enables communication with e-Health infrastructures using standardised protocols. Examples of such protocols include XML, HTTP and WML as used in the e-Health implementation by [2] and SOAP based protocol as used by [55].

The choice of protocol used will be entirely dependent on the supported communication method of the e-Health infrastructure under testing. By defining the communication interface as a separate component, future implementations of the patient simulator can be easily upgraded and extended without other components, i.e. simulation engine and patient models, being modified.
4.2 Conclusion

This chapter presents the patient simulator framework for the evaluation of e-Health infrastructures. The framework consists of four main components including: user interface, patient models, core engine, and communication interface. By separating the patient simulator into four main entities, modifications and updates to the framework can be achieved with a greater amount of flexibility and simplicity. Furthermore, changes to one component of the simulator will not affect the code base of the other components. In presenting this framework, it is proposed that a novel evaluation of e-Health environments, under the metrics of efficiency, reliability, security, and scalability can be conducted.
Chapter 5

Work to Date

5.1 Introduction

This chapter outlines the work conducted in the last year of this research. It presents the prototypes of the patient simulator which have been implemented. Very early testing work conducted using the patient simulator with a e-Health service called CareMagic which is developed by Kodit [17] is described. An outline on papers written, one as primary author and two as co-authors, is provided. Finally, a demonstration of the patient simulator, implemented as a WCF service to show the proposed frameworks flexibility is outlined.

5.2 Prototyping of Patient Simulator

The simulator has gone through several phases of prototyping throughout this years research. This report presents the first every primarily prototype of the patient simulator and the latest implementation to provide basis for comparison in terms of improvements made. The first implementation, using the .NET Framework and C# programming language is shown in Figure 5.1. The key limitation in this implementation is that the graphs generated for the vital physiological signs are exceptionally "spiky" and no model of a patient is actually implemented. Therefore, upon carrying out multiple iterations, the latest implementation of the patient is presented in Figure 5.2 along with a brief description on each component. A concise technical analysis on the techniques used for the generation of vital physiological signs in the current implementation of the patient simulator is provided in Section 5.4.

5.2.1 Early Testing of CareMagic

CareMagic is a e-Health web service currently under development. The main aim of this project is to provide a web based service which allows for medical staff to manage the input of patient data within hospital environments. Although similar to EHRs, the novelty in this work is that development is focused on providing a service in which medical staff can manage patient data via portable hand held devices including PDA’s and mobile phones.

Furthermore, as part of this implementation, development is under way to migrate the paper-based EWS system to a automated electronic system. Via the use of portable
CHAPTER 5. WORK TO DATE

Figure 5.1: Patient Simulator First Prototype

Figure 5.2: Patient Simulator Current Prototype
hand held devices, patient’s vital physiological signs can be uploaded periodically to the e-Health service or automatically captured using medical equipment. The EWS application will then calculate a risk score based on vital sign inputs and produce an alert if the patient’s health is abnormal.

In order to test the functionality of this e-Health service without the need to use real patients, the patient simulator - as proposed in this research - was used to simulate patient’s vital physiological signs and upload the data to CareMagic. Since CareMagic is a web based service, the vital physiological signs generated were uploaded using JSON format for ease of interoperability. As examples, Figure 5.3 demonstrates the blood pressure chart whilst Figure 5.4 shows the EWS chart, both generated by CareMagic based on vital sign simulation of patient “John A. Doe” using the patient simulator.

Since the testing work conducted here is preliminary, comprehensive evaluation results are not available. However, in carrying out this early testing, it was discovered
that there were flaws in the reliability of CareMagic when uploading vital physiological data of a patient. Using the default timer interval of 1 second in the simulation of vital signs, the e-Health service became unresponsive due to its inability to process data fast enough (similar to DoS attack). Based on this discovery, modifications were made to CareMagic’s code to improve upon reliability. Hence, this piece of work has demonstrated the patient simulators ability in evaluating e-Health services functionality at a very early stage. Further testing of CareMagic, and other e-Health services, will be carried out along with more comprehensive evaluations in the future as part this research.

\section*{5.3 Co-Authored Papers}

Two papers have been co-authored in the past year of research. This section provides a brief description on the two papers.

\subsection*{5.3.1 IWEI 2011 Paper}

This paper was co-authored and submitted to the International IFIP Working Conference on Enterprise Interoperability to be held in Sweden. The paper is titled: ”Towards a European E-Health Cloud. Architectural approaches and solutions for the health care and ambient assisted living industry”. Unfortunately, the paper has not been accepted however, in contributing the paper, a good deal of background has been learned in regards to organisational issues in regards to cloud based e-Health platforms.

The abstract to this paper is provided as follows:

"Societal changes, financial constraints and changes in the way how care will be delivered in the future have triggered an evolution of novel technologies in the health care and ambient assisted living industry. Novel smart devices, tracking technologies and mobile devices will enable access to medical information at any time from anywhere and anyhow. Cloud computing will play an important role in the upcoming patient centric distributed care models. This paper lines out the technological problems such as inter-cloud service federation, federated identity and trustworthiness (safety, security, resiliency) which are currently being addressed in the UK based DACAR project and the MUNICH platform which is under development in Germany."

\subsection*{5.3.2 IEEE CLOUD 2011}

The secondary co-authored paper is titled "DACAR Platform for eHealth Services Cloud" and deals with the subject of introducing the DACAR platform in concise technical technique. The concept of the patient simulator is also touched upon briefly in this paper. The paper has been submitted to the IEEE Cloud 2011 and notification of acceptance 1 April 2011. The abstract of this paper is provided as follows:

"The use of digital technologies in providing health care services is collectively known as eHealth. Considerable progress has been made in the development of eHealth services, but concerns over service integration, large scale deployment,
and security, integrity and confidentiality of sensitive medical data still need to be addressed. This paper presents a solution proposed by the Data Capture and Auto Identification Reference (DACAR) project to overcoming these challenges. The DACAR platform uses a Single Point of Contact, a rule based information sharing policy syntax and data buckets hosted by a scalable and cost-effective Cloud infrastructure, to allow the secure capture, storage and consumption of sensitive health care data.

A prototype of the DACAR platform and a demonstration application have been implemented in order to assess the viability of this approach. Simulated experimental results show that the end-to-end communication latency of 97.8% of application messages were below 100ms. Hence, the DACAR platform is efficient enough to support time critical eHealth services. In future work we plan to carry out a more comprehensive evaluation of the DACAR platform in a real life medical environment as well as to roll out a number of commercial eHealth services.

5.4 Primary Author Paper

The paper, titled "Patient Simulation: Towards Testing and Validation of e-Health Infrastructures", has been submitted to the ICST Pervasive Health 2011 conference which is to be held in Dublin. The paper details work which is similar to the overall topic of this report however it focuses on one specific novel aspect of the patient simulator: vital physiological sign simulation. Furthermore, as a secondary contribution, the paper also analyses the limitations found in the Early Warning Score (EWS) system used in healthcare environments along with presenting a proposed solution which is being developed by the DACAR project.

As demonstrated in Section 4, the patient simulator framework consists of four main components. In the paper submitted to the ICST, a focus on the core engine prototype is provided. In particular, a focus on the implementation, experiments and evaluation of the generation of vital physiological signs including heart rate, blood pressure, oxygen levels and respiration rate was conducted. The details of this work is provided in the sections which follow.

5.4.1 Generation of Vital Physiological Signs

In order to generate vital physiological signs for the patient simulator, the choice using DES and probability based behavior is implemented (see Section 4.1.3 for justification). The probability statistics concept of Normal Distribution has been applied for the generation of vital physiological sign values. Alternatively known as Gaussian Distribution [93], normal distribution is the theory that by generating a set of random values, and applying the mean (average) and standard deviation (variance), the values will tend to cluster around the mean [94, 95].

To provide an example, consider the vital physiological sign of a patient’s heart rate. The work of Fox et al. [96] state that the average heart rate of a normal healthy adult person will fall under the range of 75 to 85 beats per minute (BPM). Given this fact, for the purpose of generating a heart rate using normal distribution, it can be stated that the mean value is equal to 80. In applying the arbitrary value of 1.5 for standard deviation, a random number of x samples will provide variables which will
resemble a bell-curved chart as shown in Figure 5.5. In using this technique for the generation of vital physiological signs at each time interval (the discrete event), the concept of "controlled randomness" is produced, yet the simulation retains realism in the variables due to the fact that the values will always be catered towards a mean. In other words, the vital physiological signs which simulated will never deviate too far from what is considered abnormal, e.g. a heart rate of 300 BPM at time interval 1 and then a heart rate of 10 BPM at time interval 2 would never occur.

![Figure 5.5: Normal Distribution of Heart Rate](image)

Defining a mean value for each vital physiological sign is outlined later on in this section but, unfortunately, determining the standard deviation proves to be a lot more difficult. In determining the standard deviation which would be applied to heart rate, as an example, one must consider the heart rate variance (HRV) of a patient. The HRV is the change in a person’s heart rate from each time interval. A number of factors contribute to the HRV of a patient, including the gender and age of a patient, the lifestyle choices they make, e.g. smoking and alcohol consumption [97] and any diseases they may have such as diabetes [98].

Due to such complexities in defining a variation for heart rate (and the other four vital physiological signs), by default, the patient simulator applies a arbitrary standard deviation for each of the vital signs. The default standard deviation value used for each vital sign is relatively small, with heart rate (as an example) using the value of 1.5. Far from being a limitation, since both the mean and standard deviation values can be dynamically changed prior to, or during, simulation, it is possible to adjust the variance of a simulated patient’s vital signs during the course of simulation depending on what the purpose of the simulation is. For example, in the case of testing and validating the EWS application, one may first define a standard deviation of 1.5 for heart rate, validate that the EWS considers the generated heart rate as "normal" before applying a widely inaccurate standard deviation such as 150 to test how the system will cope and react to this anomaly.

Table 5.1 presents the default mean and arbitrary standard deviation applied to all five vital physiological signs which are simulated in a patient by default. The mean value of blood pressure is based on the work of Pesola et al. [99] in which they state that a normal systolic blood pressure is found to be 112 mmHg. From studies carried out by both Mackowiak et al. and Shoemaker the result of 36.8 °C [100, 101] is applied for the mean body temperature. O’Driscoll et al. [102] defines normal Spo2 as 96-98%, hence the average value of 97% is used. Finally, both Sherwood and Tortora et al. agree that the mean respiration rate is found to be 12 breaths per minute [103, 104].
5.4.2 Experimentation

Two experiments were presented in the ICST paper. The default timer interval, i.e. interval in which a vital physiological value is generated, of one second was used in both experiments. The first experiment simulates a normal healthy patient by applying the default mean and standard deviation to each vital physiological sign as established in the previous section. The goal in this experiment is not to prove the "realism" of the simulator but simply to demonstrate that the implementation works and is capable of generating values which cater towards a mean. The Patient Simulator was ran over the course of one hour and generates each of the five vital physiological signs at one second intervals. Figure 5.6 presents simulation results using the graph generated automatically by the Patient Simulator’s excel report generation function.

The second experiment aimed to compare the approach taken by the Patient Simulator in generating heart rate with values seen in a real life patient. In order to provide

---

**Table 5.1: Mean and Standard Deviation of Vital Signs**

<table>
<thead>
<tr>
<th>Vital Sign</th>
<th>Mean</th>
<th>Standard Deviation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Heart Rate</td>
<td>80 BPM</td>
<td>1.5</td>
</tr>
<tr>
<td>Blood Pressure</td>
<td>112 mmHg</td>
<td>2</td>
</tr>
<tr>
<td>Body Temperature</td>
<td>36.8°C</td>
<td>0.2</td>
</tr>
<tr>
<td>SpO2</td>
<td>97%</td>
<td>0.5</td>
</tr>
<tr>
<td>Respiration Rate</td>
<td>12 breaths per min</td>
<td>1</td>
</tr>
</tbody>
</table>

---

Figure 5.6: Simulation of Normal Patient
a basis of comparison, real life heart rate samples of a healthy patient provided by PhysioNet \[105\] was used. A sixty minute heart rate sample was taken from the data file \texttt{a3rr.txt} available at \[106\]. The heart rate samples provided by PhysioNet were measured in RR-intervals, therefore the conversion to beats per minute was carried out using the equation:

\[
\text{HeartRate (BPM)} = \frac{60}{\text{RRInterval}}
\]

The mean heart rate, from the PhysioNet data provided, was 68 BPM whilst calculations produced a standard deviation of 13.58. It is acknowledged that these two values differ from the predefined values in the Patient Simulator. Therefore, in order to provide a more leveled basis of comparison, the mean and standard deviation of the simulated heart rate were adjusted to the exact same values as the samples found in PhysioNet. Table 5.2 provide the minimum and maximum simulated heart rate values found after running the experiment in comparison with the real-life data whilst Figure 5.7 provides a overlay graph of the simulated heart rate against the real life heart rate.

<table>
<thead>
<tr>
<th>Data</th>
<th>Minimum Value</th>
<th>Maximum Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>PhysioNet Heart Rate Sample</td>
<td>35 BPM</td>
<td>240 BPM</td>
</tr>
<tr>
<td>Simulated Heart Rate</td>
<td>8 BPM</td>
<td>114 BPM</td>
</tr>
</tbody>
</table>

### 5.4.3 Summary and Evaluation

The results of the first experiment show that by using normal distribution and discrete event based simulation, the Patient Simulator is capable of simulating each of the five vital physiological signs of a patient with a control over the mean values but a degree of randomness related to normal. As displayed in Figure 5.6, the vital sign values generated remain very close to the mean value which have been specified. However, it can also be seen that, in some instances, the values will be slightly greater or lesser than the mean value. This allows us to simulate the variance of a patient’s vital signs yet remain within values which would be seen in patients monitored in real life scenarios.

By applying the same mean and standard deviation as the data provided by PhysioNet, the results from the second experiment show that the Patient Simulator is capable of simulating heart rate values which resemble a real-life patient as the overlay graph demonstrates (Figure 5.7). However, since a standard deviation of 13.58 was applied to the simulated heart rate to match the real life data (in order to provide a better basis of comparison), it was found that the simulated heart rate would reach very low values, to a minimum of 8 BPM (as demonstrated in Table 5.2). This would be considered abnormal since the heart rate samples provided by PhysioNet only reaches a minimum of 35 BPM. In future, thresholds need to be implemented in the Patient Simulator to prevent such low values from occurring.

On the opposite spectrum, the maximum heart rate which was simulated was 114 BPM which, though quite high, is still a realistic representation of a heart rate. However, the maximum heart rate sample provided by PhysioNet was 240 BPM, far beyond what is shown in Figure 5.7. On more than one occasion, these exceptionally high peaks in the heart rate is found in the PhysioNet data provided. Upon investigation, it was found that these peaks are considered "outliers" meaning that they are
Figure 5.7: Comparison of Simulated Heart Rate with Real-Life Data
produced due to machine faults such as signal loss or artifacts mis-detected as a heart beat. Rather than attempting to filter out these outliers, it was chosen to include them in the secondary experiment since it demonstrates that real-life medical monitoring equipment may not always produce realistic or accurate results.

5.5 Additional Experimental Work: WCF Service Implementation

An additional experiment was conducted which involved the implementation of the patient simulator as a Windows Communication Foundation (WCF) Service. To provide a brief definition, the WCF service (provided by .NET Framework) is a method used for producing applications which can communicate across the web and local networks. The main advantage of this implementation method is that it provides a very high level of interoperability with existing applications since it uses the .NET Framework (therefore supporting standardised protocols including XML, HTTP, SOAP and JavaScript Object Notation). Clients wishing to access the WCF service simply invoke the methods which are presented by the service (known as Operation Contracts).

The main purpose in conducting this implementation was to validate whether or not the patient simulator would function better as a background service rather than an executable application. Since the design of the patient simulator framework is four-tiered, most of the code base could be reused (such as the simulation engine and patient models). In the case of migrating the patient simulator to WCF service, the Graphical Interface component was simply removed and modifications made to the Communication Interface to support WCF functionality.

Serialisation of data was conducted using JavaScript Object Notation (JSON) in order to provide the highest level of interoperability with client machines. The operation contracts including Start() and Stop() methods were implemented so that a client could invoke simulation of a patient remotely. Once a client machine invoked the Start() method, the simulator would conduct the simulation of a patient’s vital physiological signs in the exact same manner as described in the previous sections of this report. Since WCF services (by itself) produces no output, it was chosen to output the simulated data to CareMagic in order to demonstrate the functionality of this implementation. Figure 5.8 demonstrates the abstract model in this implementation. In the case of stopping the simulator, the remote client simply invokes the Stop() method.

5.5.1 Advantages

In preliminary testing of this implementation, it was found the simulation of a patient was conducted in the same manner as was described in Section 5.2.1. As expected, the simulated data would simply be uploaded to the CareMagic website in the exact same manner as in the stand alone application. The main advantages this implementation demonstrates is that the proposed patient simulator framework is highly modifiable. Since the core components of the framework are implemented independent of the user interface, migration of code can be conducted with relative ease.
5.5.2 Drawbacks

Although the WCF service implementation allows for greater ease in interoperability with web services, technical issues, especially in regards to network communication, needs to be addressed. It was discovered in preliminary testing that, in the case of a remote client disconnecting without invoking the Stop() method, the patient simulator would run indefinitely. Reconnecting the client to the WCF service, and invoking the Stop() method would not work, since the service would consider this client as a new connection. Hence, the main limitation in the WCF implementation, at this point, is that further development of code is required (such as exception handling) to ensure such bugs do not occur.
Chapter 6

Proposed Novelty and Future Work

6.1 Novelty of Research

ALTHOUGH techniques in simulating patients is not entirely novel, the purpose in simulating patients - in the scope of this research - is. This research proposes the simulation of patients catered towards evaluating e-Health environments including both infrastructures and services. The current trend of healthcare simulation is in the provision of education and training [5, 85, 6, 7] of medical personnel whilst work including [87, 88] aim to improve the efficiency of hospital productivity by simulating traffic flow in hospitals. It can be safely stated that simulation of patients has yet to be used in the evaluation of e-Health environments, hence the proving the novelty of this research.

Furthermore although the challenges of evaluating e-Health environments have been highlighted, including works of [53, 65] as discussed in the literature review, and evaluation frameworks proposed [107], no comprehensive methodology for evaluating e-Health has yet to be agreed upon [54].

In the implementation of a patient simulator, it is proposed that the novel aspects of this work is in enabling the evaluation of e-Health environments under the metrics of efficiency, reliability, security and scalability. A discussion on the meaning of these metrics, in the scope of evaluating e-Health, is provided along with justification as to how the patient simulator is capable of assessing these metrics:

• Efficiency - the metric of efficiency is the measurement of time taken for an action to be conducted. In relation to e-Health environment, efficiency relates to how quickly, or slowly, the processing of a task will take. In some instances, such as updating a patient’s EHR, there is not a need for a great deal of efficiency so long as the data is processed eventually. However, in e-Health services such as the EWS application, efficiency is of absolute importance.

To justify, the EWS system is used as a preliminary method to alert medical staff on a patient’s health status. In the case of a patient having a sudden illness, one should expect the EWS application to produce an alert as quickly as possible. Since the proposed patient simulator is capable of simulating both non-medical and medical attributes, testing and validation can be conducted on e-Health services (such as the EWS) to ensure the efficiency of these applications is suitable for their purpose.
• **Reliability** - this metric relates to how dependable an e-Health environment is. Ideally, a e-Health environment should be 100% reliable, meaning no system downtime should ever occur. Of course, in real-life, this is far from likely as preliminary testing on CareMagic showed (Section 5.2.1). Therefore, the novelty of the patient simulator is that it is capable of conducting extended tests, e.g. regression testing, on e-Health environment - prior to live deployment - to assess how dependable the system is. Since simulated patients are used, reliability testing can be conducted for periods of days if not weeks and months without user intervention so long as the simulator is configured correctly.

• **Security** - security issues have been widely discussed throughout the scope of this report. Both e-Health environments and health care providers, i.e. hospitals, must ensure the security data. In particular, the laws regarding privacy of data must be followed. The patient simulator allows testing of e-Health environments at an early stage without the need to worry about this security concern since all data is simulated. Furthermore, as demonstrated in Section 3.3 & 3.4, developers of e-Health environments must also ensure their systems do not have security flaws which accidentally leak private data. Hence, the patient simulator can provide an early method in testing these environments to ensure such risks do not occur, and - if they do - then they can be fixed prior to live deployment.

• **Scalability** - this metric relates to how well an e-Health environment is capable of handling growth. With an increase in the number of patients, the system itself will experience greater and greater loads. Currently, a vast amount of patients is logged in a day-to-day basis in hospitals. In statistics provided by [108], 16.6 million finished consultant episodes were logged between the months of March 2009 and February 2010 in NHS hospitals in England. Ordinary episodes, day case episodes, finished admission episodes and emergency admissions were also logged. Overall, more than 52 million episodes are logged within the space of 12 months.

In evaluating e-Health environments, developers must ensure their system is capable of handling such high volumes of patient data. The novelty of the patient simulator is that it is capable of deploying huge numbers of simulated patients which imitate the data seen in a real-life hospital. This enables e-Health environments to be validated in a testing environment to ensure it is capable of handling scalability. By providing early validation of scalability, e-Health environments can then be deployed in live clinical-trials with certain confidence that they are capable of handling high volumes of data.

### 6.2 Time Plan and Milestones

The time plan for the next two years of this research along with some of the expected milestones is outlined in this section.

#### 6.2.1 Time Plan

Two time plans are presented. The first time plan (Figure 6.1) defines a detailed outline of the work to be conducted in Year 2 whilst the second time plan gives overview of the work to be conducted in Year 3 (Figure 6.2).
6.2.2 Abstraction of Patient Model

The outline of a patient model has been proposed as part of the patient simulator framework. Although the dynamic attributes of a patient, i.e. vital physiological signs, has been well established in this report, it is felt further work must be conducted in the static and non-medical attributes of a patient. In particular, literature and expert backing needs to be achieved in defining what the static and non-medical attributes of a patient, as seen in health records are. It is only through concise abstraction of the model that realistic simulation of a patient can be conducted.

6.2.3 Refinement of Patient Simulator Framework

Two specific limitations are noted with the patient simulator framework presented:

1. **Correlation of Vital Physiological Signs** - the simulation engine does not take into consideration the correlation between each of the five vital physiological signs. As a simple example, in the case of simulating a high heart rate, the expectation is that respiration rate would also increase, but such a feature has not yet been implemented. A wide range of literature is available on how vital physiological signs relates to each other [109] however it is also important to include professional opinions from doctors and nurse. Hence, the meetings and workshop to be conducted in the year which follows will also allow for this area of weakness in the patient simulator to be addressed.

2. **Deployment Method** - currently the patient simulator is implemented as a standalone application. Limitations are identified with this method of implementation especially in regards to scalability. Experimentations with WCF service have been conducted (Section 5.8) but it would be more ideal to implement this application as an agent based software (similar to the work of Stainsby et al. in which they propose the use of simulated agents to optimise and model traffic in hospital emergency departments [110]) in order to allow for simulated patients to be distributed throughout a large e-Health network rather than a single point of deployment.

These two areas of limitation will be addressed in future work along with overall improvements to the method applied in the simulation of patients.

6.2.4 Validation of Patient Simulator Model and Implementation

Quantitative and qualitative validation techniques, as reviewed in the literature, need to be established and conducted on the patient simulator to verify that the simulation is correct. This milestone needs to be re-iterated multiple times throughout the development process to ensure the simulator is up to a sufficient accurate standard.
6.2.5 Evaluation Metrics Formalisation

In this report, it is proposed that the patient simulator is capable of evaluating e-Health environments under the metrics of efficiency, reliability, security and scalability. Formalisation of these metrics need to be conducted so that empirical evaluation can take place. In particular, similar to validating the patient simulator framework itself, the concepts of quantitative and qualitative evaluation of e-Health environments need to be investigated. Works conducted on e-Health

6.2.6 Knowledge Transfer with Health Professional

A meeting with Professor Catriona Kennedy has been arranged on 4 March 2011. The key aim of this meeting is in knowledge transfer of patient illnesses. In particular, it is hoped knowledge can be acquired in regards to how certain illnesses, e.g. heart attacks, will affect the vital physiological signs of a patient. In conducting this knowledge transfer, more data can be gathered to provide for a more realistic simulation of a patient’s vital signs. Furthermore, it is hoped that future collaborations with nursing students can be established in this meeting so that expert knowledge can be obtained to enhance the patient model for simulation purposes.

6.2.7 Evaluation Workshop with Chelsea and Westminster Hospital

Initially, a workshop was arranged with Chelsea and Westminster Hospital on 9 November 2010. One of the agendas in this workshop is the presentation of the patient simulator. In particular, a subjective based evaluation on the method employed in the generation of vital physiological signs was to be conducted. The attendees of this workshop consisted of medical staff including nurses and medical directors. A survey was produced along with a presentation of the patient simulator. Unfortunately, due to the adverse weather conditions, the workshop was canceled. However, at the time of writing this report, it is planned that the workshop will be re-scheduled for Summer of this year. In conducting this workshop, key validation of the patient simulator model can be provided from the expert opinions of medical staff.

6.2.8 Experimentation with DACAR project

The purpose of the patient simulator is for the evaluation of e-Health environments. Therefore, having reach all milestones up to this point, the last milestone of Year 2 will be to conduct a evaluation of the DACAR platform using the patient simulator and presentation of results.

6.2.9 Six and Twelve Months Progress Review

In the six month progress review, as shown in the Year 2 time plan, it is hoped the milestone of producing a formalised abstract model of a patient can be presented. Since the last year of research will involve a high volume of experimentation and analysis and presentation of results, the milestone in the Twelve Month Progress Review is to present both the formalised simulation framework and evaluation metrics. Based on the time plan, further time has been given for further formalisation of this researches work if necessary however, it is expected that the last year of research will
be mainly experimentation based along with the presentation of results in the form of the thesis.

6.2.10 Journal and Conference Selection

After improving upon the overall patient simulator framework, along with enhancing the method for generation of vital physiological signs, multiple journals and conferences will be targeted. One of the key conferences which this research aims to provide publication in is the **Winter Simulation Conference 2011**. The conference, as the title implies, deals with all aspects of computer based simulation hence, the acceptance of a paper presenting the work of the patient simulator is highly sought after.

In order to provide a high level of peer-review in this research, international level papers will also be targeted. An IEEE journal or equivalent would be the most ideal publication. It is proposed that after the evaluation of DACAR is complete, the results produced should be of high enough standard to submit a paper to an international journal.
Bibliography


[60] Facebook, 2011.


