Using analytical and empirical techniques for improving medical device number entry systems design.

Cauchi, Abigail

Prifysgol Abertawe
Swansea University

Using analytical and empirical techniques for improving medical device number entry systems design

Abigail Cauchi

Doctor of Philosophy

2014

Abstract

User interfaces that employ the same display and buttons may look the same but can work very differently depending on how they are implemented. In healthcare, it is critical that interfaces that look the same are the same. Hospitals typically have many types of visually similar infusion pumps, but with different software versions and variation between pump behaviour, and this may lead to unexpected adverse events. For example, when entering drug doses into two similar infusion pumps, different results may arise when pushing identical sequences of buttons. These differences arise as a result of subtle implementation differences and may lead to large errors that users do not notice.

"Differential formal analysis" is a new user interface analytic evaluation method based on stochastic user simulation. The method is particularly valuable for helping evaluate safety critical user interfaces, which often have subtle programming issues. This new approach starts with the identification of operational design features that define the design space to be explored. All combinations of design features are analysed by simulating keystroke sequences containing keying slip errors. Finally, each simulation produces numerical values that rank the design combinations on the basis of their sensitivity to key slip errors.

Differential formal analysis is demonstrated through case studies of number entry systems, many of which represent a common safety-critical user interface styles found in medical infusion pumps and elsewhere. The results uncover critical design issues, and are an important contribution of this thesis since the results provide device manufacturers guidelines to improve their device firmware.

The analysis is complemented with models of usage based on 1,362 days of use of number entry systems from 19 infusion pumps over a 3 year period in a UK hospital. This thesis also suggests some improvements to medical device logging, which will help further evidence-based improvement to medical device safety.

Previously, empirical methods and analytic methods have been used independently to analyse and improve number entry system designs. This thesis identifies key contrasts in exploring number entry errors using laboratory studies and analytic methods. The implications of combining methods to more thoroughly analyse safety critical design are discussed.

Contents

Acknowledgements

I would like to thank my supervisor, Harold Thimbleby, for his guidance and support throughout this degree and for making this PhD a pleasant and enjoyable experience. Frequent meetings and discussions with Harold have been a major inspiration for this work. Valuable lessons learnt from Harold go beyond this thesis, and for that I will always be grateful.

I would like to thank Michael Harrison for his excellent supporting and complementary views about this thesis. Meetings with Michael have inspired some of this work. I would like to thank Michael for thoroughly reading this thesis and giving me valuable feedback for improvements.

My thanks extends to my friends and colleagues Andy Gimblett, Paolo Masci, Gerrit Niezen, Patrick Oladimeji and Gordon Pace for their interest and active role in discussing and challenging the ideas in this work; also Gregory Abowd, Ann Blandford, Duncan Brumby, Anna Cox, Paul Curzon, Parisa Eslambolchilar, Paul Lee, Todd Johnson, Rimvydas Rukšėnas and Sarah Wiseman for insightful meetings about this work, and of course all the other members of the CHI+MED team for their input.

I would like to thank my friend and colleague, Victoria Hurst, for her excellent administrative support throughout my PhD. Her support made the logistics of these past years possible while allowing me to use my time on this work.

I would like to thank researchers I worked with at Microsoft Research India, especially Kentaro Toyama and Saurabh Panjwani, for giving me the opportunity to see the beauty of doing HCI research and for encouraging me to embark on a PhD program.

Special thanks goes to my close friends and family who supported me throughout this degree.

Thanks goes to Nicholas Komaromy from the Royal Free London NHS Foundation Trust for making data collection for this thesis possible.

This PhD was funded by the grants, CHI+MED: Multidisciplinary Computer-Human Interaction research for the design and safe use of interactive medical devices project, EPSRC Grant Number [EP/G059063/1] and The Global Hub in Medical Technologies and NanoHealth at Swansea University, EPSRC Grant Number [EP/K004549/1].

Preface

This thesis has led to publications that are listed below. Published work where I am listed as first author is work that is directly derived from this thesis. Other publications are related work that I have been involved in during my PhD.

Cauchi, A.; Harrison, M.; Thimbleby, H.; Oladimeji, P.; Using medical device logs for improving medical device design, *ICHI'13 Proceedings of the IEEE International Conference on Healthcare Informatics*, Pages 56-65, IEEE, New York, NY, USA, 2013.

Cauchi, A.; Using Differential Formal Analysis for dependable number entry, *EICS'13 Proceedings of the 5th ACM SIGCHI symposium on Engineering interactive computing systems*, Pages 155-158, ACM New York, NY, USA, 2013.

Cauchi, A.; Dependable number entry using differential formal analysis, Institute of Ergonomics and Human Factors, *Early Careers Research Symposium Abstracts*, page 23, UK, 2013.

Li, Y.; Ding, X.; Dong, Z.; Qin, L.; Masci, P.; Vincent, C.; Thimbleby, H.; **Cauchi, A.**; Lewis, A.; Xing, B.; Sun, S.; Liu, E.; Di, J.; Wang, J.; Welch-Brady, M.; MediCHI: safer interaction in medical devices, *CHI '13 Extended Abstracts on Human Factors in Computing Systems*, pages 3267-3270, ACM New York, NY, USA, 2013.

Masci, P.; Ruksenas, R.; Oladimeji, P.; **Cauchi, A.**; Gimblett, A.; Li, Y.; Curzon, P.; Thimbleby, H; The benefits of formalising design guidelines: A case study on the predictability of drug infusion pumps, In *Innovations in Systems and Software Engineering*, Pages 1-21, Springer-Verlag London, 2013.

Cauchi, A.; Gimblett, A.; Thimbleby, H.; Curzon, P.; Masci, P.; Safer 5-key number entry user interfaces using Differential Formal Analysis, *BCS-HCI '12 Proceedings of the 26th Annual BCS Interaction Specialist Group Conference on People and Computers*, Pages 29-38, British Computer Society Swinton, UK, 2012.

Cauchi, A.; Differential Formal Analysis: Evaluating safer 5-key number entry user interface designs, *EICS'12 Proceedings of the 4th ACM SIGCHI symposium on Engineering interactive computing systems*, Pages 317-320, ACM New York, NY, USA, 2012.

Cauchi, A.; Thimbleby, H.; Gimblett, A.; Simulation to evaluate alternative approaches to blocking use errors, *Design for Medical Devices*, 2012, Minnesota, US. 2011.

Cauchi, A.; Gimblett, A.; Thimbleby, H.; Goal-based design improves interaction dependability, *Digital Engagement 2011*, Newcastle.

Li, Y.; Oladimeji, P.; Monroy, C.; **Cauchi, A.**; Thimbleby, H.; Furniss, D.; Vincent, C.; Blandford, A.; Design of Interactive Medical Devices: Feedback and Its Improvement, 2011 *International Symposium on IT in Medicine and Education (ITME)*, vol. 2, pp.204-208, Guangzhou, China.

Oladimeji, P.; Li, Y.; **Cauchi, A.**; Eslambolchilar, P.; Gimblett, A.; Lee, P.; Thimbleby, H., Visualising Medical Device Logs, *1st BCS Health Wales Workshop*, Wrexham, Wales, 2011.

Masci, P.; Ruksenas, R.; Oladimeji, P.; **Cauchi, A.**; Li, Y.; Curzon, P.; Thimbleby, H., On formalising interactive number entry on infusion pumps, *4th Inter-*

national Workshop on Formal Methods for Interactive Systems, Limerick, Ireland, 2011.

Cauchi, A.; Curzon, P.; Eslambolchilar, P.; Gimblett, A.; Harrison, M.; Huang, H.; Lee, P.; Li, Y.; Masci, P.; Oladimeji, P.; Rukšėnas R.; Thimbleby, H., Towards Dependable Number Entry for Medical Devices, *1st International Workshop on Engineering Interactive Computing Systems for Medicine and Health Care*, Pisa, Italy, 2011.

Blandford, A.; **Cauchi, A.**; Curzon, P.; Eslambolchilar, P.; Furniss, D.; Gimblett, A.; Harrison, M.; Huang, H.; Lee, P.; Li, Y.; Masci, P.; Oladimeji, P.; Rajkomar, A.; Ruksenas, R.; Thimbleby, H., Comparing Actual Practice and User Manuals: A Case Study Based on Programmable Infusion Pumps, *1st International Workshop on Engineering Interactive Computing Systems for Medicine and Health Care*, Pisa, Italy, 2011.

Thimbleby, H.; Gimblett, A.; **Cauchi, A.**, Buffer Automata: A discrete UI software architecture based on user models, *EICS '11 Proceedings of the 3rd ACM SIGCHI symposium on Engineering interactive computing systems*, Pages 73-78, ACM New York, NY USA, 2011.

List of Figures

List of Tables

Chapter 1

Introduction

The aim of medical care is to improve people's health, and this is pursued with great success today. Unfortunately, however, medical care also entails injuries to certain patients, which in some cases have fatal results. Some of these injuries are expected side-effects of medicines and treatment. They are usually accepted since the benefits of medicines and treatments are more substantial than the injuries they cause to patients.

Incorrect drug doses and incorrect drug dose calculations are a significant contributory factor in unnecessary fatalities in healthcare. There are many papers on the prevalence of prescribing errors (Dean et al., 2002), but very few on user interaction errors, since interaction errors are harder to measure as they generally do not leave a paper record that can be easily analysed. Vicente et al. (2003) estimate the probability of fatal number-entry errors on Patient Controlled Analgesia (PCA) pumps (ones controlling pain, typically delivering opiates) as between 1 in 33,000 to 1 in 338,800 (the large uncertainty is due to estimating reporting rates — many errors are not reported); or in absolute terms approximately 65–667 per year in the US or (scaling by population) 155–1,587 per year in Europe. Vicente et al. (2003) warn that these are low estimates as they are based on fatalities in the US but the PCA pump is used worldwide, and hence the denominator used, the number of pumps sold, would have been too high. By way of comparison, the probability of death from general anæsthesia is approximately 1 in 200,000–300,000.

This thesis argues that devices are at least partly to blame for the 9,800 deaths per year caused by calculation errors. So far, little research has been carried out into how to design medical devices to reduce this death rate.

1.1 A case for medical HCI

The Institute of Safe Medication Practices (ISMP) Canada has issued a report of an incident that lead to the death of a 43 year old cancer patient (Institute of Safe Medication Practices Canada, 2007). The ISMP Canada reports that the determined cause of death was “sequelae of fluorouracil toxicity”. The patient received a dose of flourouracil that was intended to be administered in four days over four hours. The medication incident was recognised an hour after the infusion was completed but injuries caused by the overdose lead to multiple organ failure and the patient’s death.

Following this incident, the ISMP Canada carried out a root cause analysis that is reported in (Institute of Safe Medication Practices Canada, 2007). As part of the root cause analysis, a study was carried out with five nurses from the same cancer care centre where the incident occurred. The five nurses were asked to perform the same task that caused the medication incident that lead to the death of the patient. This short study with five nurses found that:

- Three out of five nurses entered incorrect data
- All five nurses were confused by the setup, or the selection of mL/hr
- Two out of five nurses were confused by programming the device to administer the drug dose
- Three out of five nurses were confused by how the decimal point works on the device

The results from this study show that running a short study with 5 nurses is useful for finding out problems with medical device design. This thesis argues that usability concerns related to medical device use should be addressed before medical devices are put on the market. Better pre-market medical device evaluation can prevent incidents such as the one described in this section.

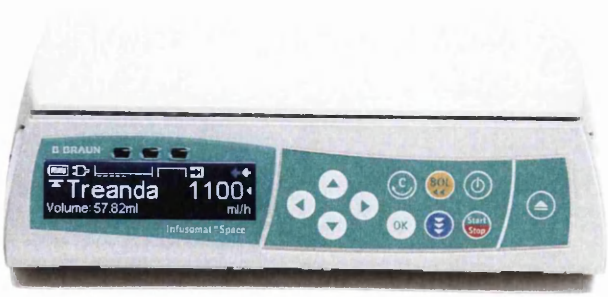


Figure 1.1: A BBraun Infusomat Space pump; a widely used drug infusion pump

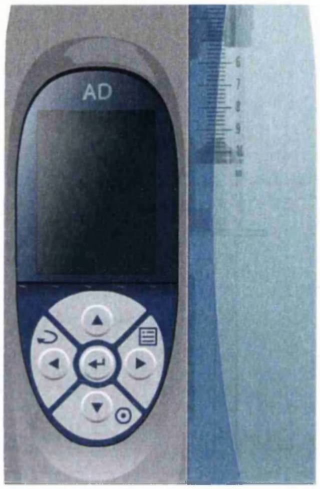


Figure 1.2: A Zimed AD pump

1.2 A typical medical device problem

Two infusion pumps from two international, world leading manufacturers were analysed. A BBraun Infusomat Space seen in figure 1.1 and a Zimed AD seen in figure 1.2. Infusion pumps are ubiquitous devices in hospitals that are used to infuse drugs, nutrition and fluids intravenously. Infusing too much of a substance is likely to cause cause harm or death, while infusing too little can leave patients untreated, which may also be harmful.

Both the BBraun Infusomat Space and Zimed AD allow clinicians to enter doses using a 5-key number entering system as seen in figure 1.3. A 5-Key interface has a display and 5 number entry buttons. On the display there is a cursor that highlights a digit and the ▲ ▼ buttons change the highlighted digit. The ◄ ► buttons move the cursor between digits and the OK button confirms the input number.

The work for this thesis was motivated by an observation of entering the dose

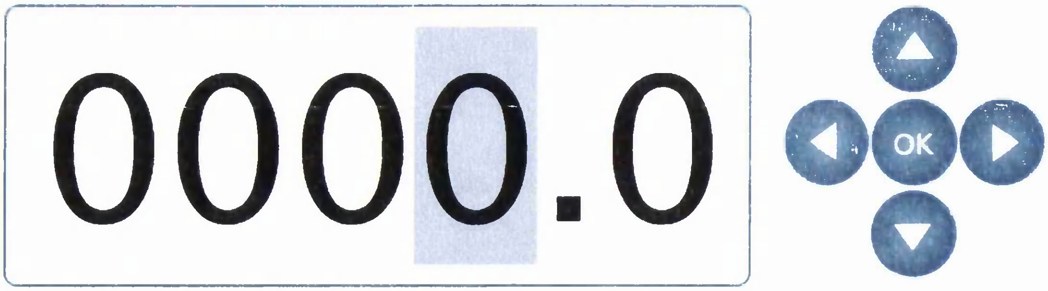


Figure 1.3: A 5-Key interface with a cursor highlighting a digit. In this system, ▲ and ▼ buttons manipulate the highlighted digit, ◄ ► buttons move the cursor between digits, and the OK button confirms the number.

Key Press	Zimed AD	BBraun Infusomat Space
	0	0
◄	00	00
▲	10	10
▲	20	20
▲	30	30
▲	40	40
▲	50	50
◄	050	050
▼	950	000.1

Figure 1.4: A key sequence being input into Zimed AD and BBraun Infusomat Space from the same key sequence. This table shows the change in displays after pressing the key in the "Key Press" column. In the first row there is no key press to show the starting displays of the two devices.

of 950 mL into the BBraun Infusomat Space and Zimed AD devices starting from the same screen and using the same key presses on both devices. From a display showing 0 on both devices (where the underline indicates the digit highlighted by the cursor), pressing the plausible key sequence ◄ ▲^{5} ◄ ▼ on the BBraun Infusomat Space and Zimed AD simultaneously resulted in the Zimed AD display showing 950 at the end of the key sequence and the BBraun Infusomat Space showing 000.1. This behaviour is illustrated in figure 1.4 what shows how the displays of both devices change when each of the keys are pressed.

Starting from 0 on both devices, both devices work in the same way until the final key press of ▼. This is where the two number entry systems are implemented to do different things that result in the different values. There is a significant difference in results between the two devices and implemented in medical infusion pumps, this difference could be harmful, or even fatal.

Chapter 4 details why the two 5-key number entry systems on the BBraun Infusomat Space and the Zimed AD work differently. There are interaction design decisions about how 5-key number entry systems are implemented in program code. Chapter 4 shows that there are at least 28 different ways of implementing 5-key number entry systems. In medical devices, it is important that all devices are standardised to implement *one* of these possible interaction designs.

The problem is larger than 5-key implementations working differently between different devices by different manufactures. The 5-key implementation of the device is implemented in the device firmware and the manufacturers upgrade firmware. Analysing the 5-key number entry system of different firmware versions of one of the commercial infusion pumps, resulted that the firmware upgrade changed the 5-key number entry system implementation. This results in devices that look identical and work differently. In a hospital, since devices are in use by patients, the device upgrade process is gradual. It happens that devices that look identical, work differently because of the firmware version installed. Nurses are overworked, and hospital wards are very busy (R. J. Koppel & Gordon, 2012), having different implementations of 5-key number entry systems can lead to unnecessary incidents that can be fatal.

Under the UK Health & Safety At Work Act (1974) and under similar legislation in other countries, devices should be designed to reduce risk to be As Low As Reasonably Practical, ALARP. This work shows that the details in medical device design, such as how to implement the safety critical number entry system to enter doses on a device, are often overlooked and they are not implemented to reduce risk to the legal requirements of the ALARP principle. This thesis shows that choosing the best design over the worst design out of the 28 possible implementations of 5-key number entry systems reduces the harm caused by human error eight-fold. It is important that the best design is implemented, rather than the worst.

1.3 Contributions

The main contribution of this thesis is a new analytical analysis method: Differential Formal Analysis. Choosing the best implementation of a 5-key interface is difficult to do using lab studies, specifically because number entry error rates are not high enough to obtain significant results of which of the designs (with subtle differences) is best. In a study of 20 participants carried out by Oladimeji et al. (2013) to find out what speed/accuracy tradeoffs there are between interfaces of different button layouts, no participants confirmed erroneous numbers on the 5-key interface. This shows that it would be very time consuming to carry out human experiments to study subtle programming differences in one button layout and obtain significant results.

Differential Formal Analysis is a stochastic simulation based method that is capable of generating millions of number entry tasks and results are significant. One of the strongest benefits of Differential Formal Analysis is that it raises important discussions and very good empirical questions that would not arise without going through the Differential Formal Analysis process. For some empirical questions raised, the Differential Formal Analysis method can be used to determine whether the results of the empirical question are important or not *i.e.* whether they will change the results from the Differential Formal Analysis. If the results of the empirical experiment matter, investing time and money in the experiment is worthwhile depending on whether the question is of value to research or marketing a device.

This thesis shows how empirical and analytic techniques can be used together to improve medical device design. Logs from medical infusion pumps are analysed to answer questions about what digit distributions are used in medical devices; how users input numbers; and more.

Finally, this thesis compares and contrasts between using a lab study approach and the analytic approach introduced in this thesis. It discusses the importance of triangulating the two approaches for designing safety critical number entry systems.

1.4 Thesis overview

An overview of the rest of this thesis is described in this section, highlighting the contributions of each chapter.

Chapter 2: Background – This chapter provides a background to this thesis. It gives a background into human error; an account of the current European and American regulations, standards and guidance on medical device design and highlights the HCI techniques used in industry. A description of current HCI techniques used to manufacture medical device design is detailed, and a description of how this thesis contributes to the current manufacturing process is given. Finally, the current state of the art of number entry research is described.

Chapter 3: The Differential Formal Analysis Method – This chapter introduces a novel analysis method for analysing safety critical user interface implementations. The Differential Formal Analysis process is described, and the core method used throughout this thesis, Stochastic Key Slip Simulation (SKSS) is detailed.

Chapter 4: Differential Formal Analysis Case Study – 5-Key Number Entry – This chapter makes a case for safety critical 5-key number entry design. 5-Key number entry is analysed as a case study of Differential Formal Analysis using SKSS. Implementation details of the SKSS method are described and finally, results for 5-key number entry are presented.

Chapter 5: Empirically informed SKSS for medical device design – This thesis goes on to describe how the SKSS method can be empirically informed to tailor an analysis to a domain, in this case, medical devices. Medical device logs from 19 infusion pumps used in a UK hospital over the period of three years are analysed to retrieve empirical parameters to be used in the SKSS method. An empirically informed SKSS analysis is carried out on 5-key number entry systems and results are presented and compared to the generic results that were previously presented in Chapter 4.

Chapter 6: Triangulating Stochastic Key Slip Simulation and Empirical Techniques – This chapter extends the previous implementation of SKSS to a tool that accepts a description of a number entry system in JavaScript Object Notation. Three number entry systems that were previously studied by Oladimeji et al. (2013) in a lab study are analysed using SKSS. The results from the SKSS analysis are discussed in relation to the results in (Oladimeji et al., 2013) and an argument for triangulating analytic and empirical techniques is made for safety critical number entry design.

Chapter 7: Conclusions & further work – This chapter summarises this thesis, outlines further work and makes concluding remarks.

Chapter 2

Background

This thesis is about providing evidence — for number entry systems in medical devices — to show that they are designed to cause the least possible harm in the presence of human error. This chapter discusses a background into: human error; the regulations and standards enforced for the manufacturing and marketing of medical infusion pumps; a broad view into usability and HCI and a description of how sufficient safety evidence is lacking in currently employed methods; a description of relevant empirical and theoretical work about number entry system design.

2.1 Human error

Incorrect actions by the user, i.e. human errors, are important to deal with in improving safety. Much work has been devoted to defining and classifying human error. A classic definition is given by Reason (1990): human error is *a generic term to encompass all those occasions in which a planned sequence of mental or physical activities fails to achieve its intended outcome, and when these failures cannot be attributed to some chance agency.* Reason (1990) subdivides human error with the aid of the Generic Error Modelling System (GEMS) and according to this subdivision, an incorrect action is one of the following:

- Slip

- Lapse

- Rule-based mistake

- Knowledge mistake

- Violation

Reason (1990) makes a distinction between active failure (active error) and latent condition (latent error). Active failures are the direct errors committed by people in a system, while latent conditions are existing defects in the system such as poor design and deficient training that, when combined with local circumstances, can result in exposure to hazard. These defects may lie latent for a long time without any harm arising; hence the term. Latent conditions can also increase the probability of active failure by creating local factors that promote error. The human errors that have been considered in this thesis are of the active failure type.

Reason's Swiss Cheese model (Reason, 2000), is widely accepted in healthcare. It illustrates systems as stacked slices of swiss cheese where each slice of cheese is a layer of defence against an incident occurring. Figure 2.1 shows that each layer of defence is not perfect and it has holes that could let an incident go through that layer. If an incident happens to go through the holes of each layer of defence, the incident happens. The model clearly illustrates that it is only when the holes in the swiss cheese model line up that an incident occurs.

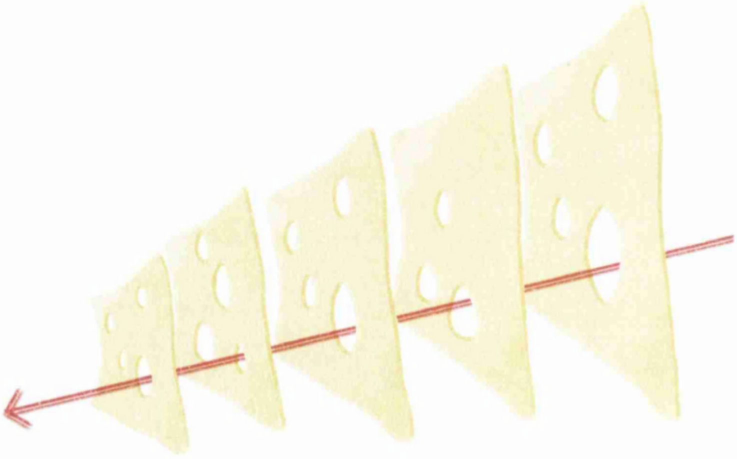


Figure 2.1: The swiss cheese model illustrates that when the holes in the levels of defence line up, accidents go through.

In healthcare, clinicians are largely considered as the last layer of defence. Incident investigations are focused on nurses rather than on the larger system, such as in the widely reported cases the nurse Kimberly Hyatt who was suspended for a dosing error (Ostrom, 2011) and Denise Melanson, who was a cancer patient who died because of the lack of usability studies on the infusion pump used in her therapy (Institute of Safe Medication Practices Canada, 2007).

This thesis argues that in safety critical drug dose input, the last layer of defence is not the nurse — it is the medical device. In the case of the cancer patient fatality described in chapter 1, the nurse who administered the drug is not the last layer of defence. The device could have been designed better to prevent the fatal error from occurring. This thesis shows that like all layers of defence, the medical device also has holes in it. Improving number entry design will not lead to no number entry errors. The aim, using the swiss cheese analogy, is to make some holes in the swiss cheese slice *smaller*.

Two techniques that help in designing devices to prevent human error are Systematic Human-Error Reduction and Prediction Approach (SHERPA) (Stanton, 2003) and Task Analysis For Error Identification (TAFEI) (Stanton & Baber, 2005). Both SHERPA and TAFEI aim to improve device design by preventing errors from occurring in the design of the system and they have been used to reduce errors when withdrawing cash from automatic teller machines (Burford, 1993), recalling a phone number on mobile phones (Baber & Stanton, 2001), buying a ticket on the ticket machines on the London Underground (Baber & Stanton, 1996), operating high-voltage switchgear in substations (Glendon, Clarke, & McKenna, 2006), medical applications (Yamaoka & Baber, 2000) and more.

SHERPA (Stanton, 2003) is a method used to determine possible error modes in the device. A Hierarchical Task Analysis (HTA) (Annett, 2003) is used to represent task performance in terms of goals, operations and plans. To apply SHERPA, each task step is classified as either: action; retrieval; checking; information communication; or selection. After this classification has been carried out, the analyst considers credible error modes associated with that activity. The errors are described in terms of the consequence in order to determine how critical they are. One of the problems with the SHERPA method, identified in (Stanton, 2003), is that novice users over predict errors. TAFEI aims at predicting errors that can occur by comparing normative behaviour with possible transitions of the state of a device, this makes it less likely to generate false alarms than SHERPA.

TAFEI (Stanton & Baber, 2005) attempts to predict errors with device use by modelling the interaction between user and device. In the TAFEI method, a hierarchical task analysis (HTA) is performed to model the human side of the interaction, then a state-space diagram (SSD) is drawn up to describe the device model. Plans from the HTA are mapped onto the SSD and finally, a transition matrix is drawn up to display state transitions when the device is used. The aim of TAFEI is to highlight state transitions that are possible but undesirable. TAFEI's use in design is to help design devices that make the possible undesirable actions impossible in the device design.

SHERPA and TAFEI improve device design by redesigning systems such that possible human interactions do not lead to erroneous states. When considering number entry in medical devices, it is difficult to predict which number entry states are erroneous or not. In a device where the possible inputs are 0 – 99999 any input can be valid for specific goals therefore preventing a particular state from occurring (in this case, a particular number) is not the way to improve number entry design. This thesis presents an approach to reduce the severity of number entry errors by simulating number entry tasks, simulating possible errors and measuring the severity of errors. A number entry design is chosen based on which design is predicted to lead to the least number of severe errors.

2.1.1 "Out by r" error definition

In healthcare, giving a patient ten times too much (or too little) of a drug is almost always a critical error regardless of the drug involved. This is often called an out by ten error. More generally, if the number a user enters e is *at least* r ($r > 1$) times higher or lower than the intended number, this is called an "out by r" error. For example, the user entering (at least) 100 when 10 was expected or (at most) 10 when 100 was expected has made an out by 10 error. Of course, entering 100 or more in error when intending 10 may have made an out by 11, out by 20, etc, error, which would be worse than "just" an out by 10 error. In general any out by x error is also an out by y error if $x \geq y$. In this thesis, out by r error is commonly referred to in order to describe error magnitudes.

2.2 Regulation and standards

Medical device manufacturers go through regulation processes to legally sell their devices in various countries. This section describes the processes that medical device manufacturers go through to be able to legally market and sell their devices in the Europe Union (EU) and the United States (US).

2.2.1 EU legislation

To market and sell a product in the European Economic Area (EEA), the European Commission requires that manufacturers get CE Marking (European Commission, 2013) for their device. The CE Mark is a declaration by the manufacturer that the device conforms to EU legislation and that the manufacturer assumes full responsibility for that conformity. The CE Marking process involves:

1. Deciding which EU Directives are applicable for the product
2. Ensuring that the product is applicable with the relevant Directives by testing and applying the relevant conformity assessment procedures
3. Compiling and retaining a technical file that satisfies the requirements of the Directives
4. Writing and signing the Declaration of Conformity and keeping the original with the technical file
5. Applying CE Marking to the equipment in accordance with the requirements of the Directives

EU Directives require that manufacturers certify that they have followed standards to manufacture their product. In regards to medical infusion pumps and interaction design, manufacturers are required to follow the usability standard EN ISO 62366:2008 Medical devices - Application of usability engineering to medical devices (BS EN, 2008).

As discussed in chapter 1, under the UK Health & Safety At Work Act (1974), it is required that safety critical and dependable applications should abide by the ALARP principle. This requires that medical device manufacturers use methods and tools to provide evidence that their devices abide by this principle.

2.2.2 US legislation

The United States Food and Drug Administration (FDA) regulates medical devices that are to be sold in the United States. Medical devices that are similar to others that are FDA approved, are regulated through the FDA's 510k premarket notification process (US FDA, 2013). Infusion pumps manufacturers use the 510k premarket notification process since the the type of device (infusion pump) has already been approved.

In a letter to infusion pump manufacturers (Shuren, 2010), the FDA writes:

> FDA has seen an increase in the number and severity of infusion pump recalls. Analyses of Medical Device Reports (MDRs) have revealed device problems that appear to be a result of faulty design. Between January 1, 2005 and December 31, 2009, FDA received over 56,000 MDRs associated with the use of infusion pumps. Of these reports, approximately 1% were reported as deaths, 34% were reported as serious injuries, and 62% were reported as malfunctions.
>
> The most frequently reported infusion pump device problems are: software error messages, human factors (which include, but are not limited to, use error), broken components, battery failure, alarm failure, over infusion and under infusion. In some reports, the manufacturer was unable to determine or identify the problem and reported the problem as "unknown." Subse-

> quent root cause analyses revealed that many of these design problems were foreseeable and, therefore, preventable.

The 510k process involves submitting the required documentation for device review. The documentation includes proof that the device is substantially similar to a device that is legally marketed in the US, and that the manufacturing process followed the relevant FDA guidance documents and standards.

After the forms are submitted by a company, the documentation is reviewed by the relevant FDA department and manufacturers are notified in approximately 90 days. Manufacturers can question the guidance documents and relevant standards and use a process that is fit for the device they are manufacturing, however, this would considerably delay the approval process.

In the case of infusion pumps, manufacturers are required to demonstrate (through documentation) that they have followed medical devices design standards such as the standard, ISO 62366:2008 Medical devices - Application of usability engineering to medical devices (BS EN, 2008).

The ISO 62366:2008 standard includes usability engineering processes from Nielsen (1994) applied to medical devices, risk management, and usability validation. The processes described in the standard are processes that have been used in desktop application development since Nielsen (1994) and Norman (2002) made them popular for the success of desktop application software companies.

These processes outlined in the relevant standards were not developed with the intention of safety critical design in hospital scenarios but rather, for the development of desktop application software used in homes and offices. The HCI standards used in the medical device design processes are therefore not design with safety in mind.

This thesis aims at developing methods to provide evidence that should be used in process of medical device design and regulation. The method for providing evidence in this thesis is designed for safety critical medical device interaction design, making it important to include in the current manufacturing processes and regulation.

In relation to infusion pumps, the FDA has a Generic Infusion Pump project (Arney, Jetley, & Jones, 2007) where the aim is to develop a set of safety models that manufacturers can employ in their infusion pump manufacturing process. These models can be used as a safety reference standard to verify safety properties in various classes of infusion pumps. The Generic Infusion pump was designed following a Hazard and Operability (HAZOP) analysis carried out by Y. Zhang, Jones, and Jetley (2010).

A risk matrix (NHS National Patient Safety Agency, 2008) is commonly adopted in risk analysis in healthcare and aviation. It defines safety as the product of the *likelihood of occurrence* of the risk and the *severity of the consequence* if that risk occurs. In relation to the risk matrix, in this thesis, the risk is human error occurring while a practitioner enters a drug does in an infusion pump. The aim of this thesis is to reduce the severity of the consequence caused by human error, rather than reducing human error itself.

The method presented in this thesis provides evidence that regulatory bodies should require for pre-market approval.

2.3 Usability and HCI

The medical device usability standard, ISO 62366:2008 refers to the most commonly employed usability standard, ISO 9242-11:1998 for guidance on the usability engineering process. ISO 9242-11:1998 defines usability as "The effectiveness, efficiency, and satisfaction with which specified users achieve specified goals in particular environments." The components are explained as follows:

- Effectiveness – The accuracy and completeness with which specified users can achieve specified goals in particular environments.
- Efficiency – The resources expended in relation to the accuracy and completeness of goals achieved.
- Satisfaction – The comfort and the acceptability of the work system to its users and other people affected by its use.

The HCI methods advised in usability standards are based on well established literature in HCI, among which is Nielsen's *Usability Engineering* (Nielsen, 1994), Norman's *The Design of Everyday Things* (Norman, 2002) and Shneiderman's *Designing the User Interface* (Shneiderman, Plaisant, Cohen, & Jacobs, 2013).

Nielsen, Norman and Shneiderman give user interface design principles for designing applications. While these principles are very useful for designing commercial systems (like word processors), arguably, they were not conceived or written for designing safety critical user interfaces.

One of Nielsen's usability heuristics is about "error prevention". Nielsen (1994) suggests that when designing for error prevention, clear, understandable error messages should be displayed to the user when an error occurs. Nielsen goes on to discuss that even better than error prevention, designers should either eliminate error prone conditions or check for then and present users with a confirmation option before committing to the action.

In medical scenarios, eliminating error is not always possible. Consider infusion pumps as an example, the value of a drug being entered can have a wide range of possible valid values. Nurses care about patients and their intentions are to help and care for patients, however, nurses are overworked and often sleep deprived, and often find themselves doing safety critical tasks in less than ideal conditions (R. J. Koppel & Gordon, 2012). Barcodes have been used to prevent nurses from entering drug dose values themselves, however barcodes bring about a new set of safety critical problems as described in (R. Koppel et al., 2008). Among some of the problems: barcodes get pealed and broken, making them impossible to scan; barcodes go missing; and more.

In the book *Usability Engineering* (Nielsen, 1994), Nielsen describes that error should be prevented however, there are no mathematical engineering methods presented on how to to prevent (or reduce) error.

Shneiderman's rules (Shneiderman et al., 2013) also describe that systems should have simple and understandable error handling. Like Nielsen (1994) the guideline is for designing to prevent error and to provide useful error messages.

In ISO 9242-11:1998 there is a focus on *iterative design* (Nielsen, 1993). The iterative design process is commonly used in HCI since it allows designers to identify usability problems. The method involves developing a prototype of the user interface of a system, giving it to users to use it, noting any issues with the design then refining the user interface design based on what was learnt by observing users using the system.

When applied to safety critical systems such as entering drug doses into medical infusion pumps, one of the main aims of the user interface design is to design the system to prevent users from entering wrong values. In an iterative design process, a number entry system is developed and people are observed using it. The number entry system is then refined to design a number entry interface that reduces the number of errors that users make. Through this iterative process, number entry errors are significantly reduced after several iterations user interface design refinement, possibly bringing errors down to zero.

In the ideal situation where iterative design reduces number entry error to being undetectable in user trails, human error will still occur in hospitals when medical devices are used for longer periods of time. It is infeasible to run well designed user trials that are long enough to detect human error in number entry systems that are refined to reduce the likelihood of human error occurring. At this stage in design, analytic methods are useful to further refine number entry systems to reduce the harm caused by human error. In safety critical systems such as medical infusion pumps, it is worthwhile reducing harm regardless of how many fatalities are prevented.

2.4 Empirical studies for number entry system design

Azenkot, Bennett, and Ladner (2013) present a longitudinal evaluation of the DigiTaps system. DigiTaps is presented as an eyes-free interaction system with minimal audio feedback for touchscreen devices. The study by Azenkot et al. (2013) is intended for text entry by blind people and although it presents results on accuracy, it does not mention how big the errors made by the participants are. The system is not intended for use as a safety critical number entry system, rather as a system for blind people to enter numbers into their mobile devices.

Hesselmann, Heuten, and Boll (2011) present a technique for entering numbers into interactive muti-touch tabletop devices using 10 fingers and both hands. The method for number entry in the Tap2Count system by Hesselmann et al. (2011) is to tap the number of fingers for each digit on a tabletop so for example, if the digit 8 is to be input, all the fingers from one hand and three fingers from the other hand are to be tapped. The evaluation and discussion by Hesselmann et al. (2011) focusses on how well participants understand the system and whether they like it. This initial evaluation of 12 users showed no errors in this system.

Martin, Clark, Morgan, Crowe, and Murphy (2012) take a user-centred approach to requirements elicitation in medical device design, using a medical imaging device as a case study. The method of this research consisted of a brainstorming session with the device development team, interviews with potential users, data analysis from the previous two stages of the study, feedback of the data analysis to the development team and finally, interviews with the development team.

Oladimeji, Thimbleby, and Cox (2011) compare so-called serial and incremental number entry interface styles. They used eye tracking, and uncovered important design principles (including an explanation of why incremental interfaces are more dependable than numeric keypad interfaces for number entry). Chapter 4 in this thesis compares compares 5 different design features in up to 28 combinations: this scale of comparison complements Oladimeji, Thimbleby, and Cox (2011)'s empirical work by targeting subtle variations in interface layouts which previously were all classed under a single "incremental" heading.

Oladimeji et al. (2013) carry out a lab study to compare five different styles of number entry interfaces (shown in figure 2.2) using the criteria of speed and accuracy. The study indicated that error rates in lab experiments are low, and undetected error rates (i.e., errors nurses make that they do not notice) are even lower. This thesis takes a closer look at number entry systems. Chapter 1 uncovered that number entry interfaces that have the same hardware layout can be programmatically implemented to work differently and this may cause unnecessary harm. The results in the lab experiment by Oladimeji et al. (2013) show that the number of errors that participants make in lab studies are very low. To evaluate different implementations

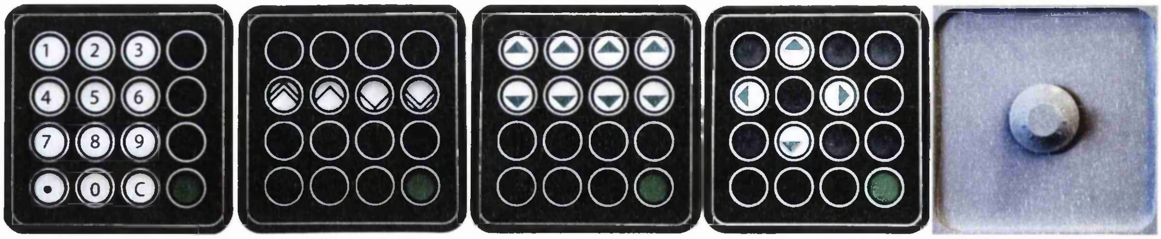


Figure 2.2: Number entry keyboard layouts studied by Oladjimeji et al. (2013). Image taken from (Oladjimeji et al., 2013).

of the same number entry system, the lab experiment method used in (Oladimeji et al., 2013) would have to be run for a longer time, making it impractical for obtaining significant results. This thesis introduces an analytic approach to number entry system design to further investigate how to design number entry systems to further reduce the harm caused by human error when empirical trials become impractical.

Medical device logs that can be retrieved from devices that are currently used in hospitals can be a rich source of empirical data. Medical device logs have been analysed by Lee et al. (2012) and Wiseman, Cox, and Brumby (2013). Lee et al. (2012) present valuable insight into: how infusion pumps are used in hospitals; how we can gain insight into hospital infrastructure; how much it costs for nurses to attend to alarms over a period of a year; and other interesting findings that can significantly improve healthcare systems. In the work presented in chapter 5, there is a different focus on retrieving information from logs. The concern in this thesis is to use medical device logs to answer very specific empirical questions to tailor the analytical method (that is introduced in this thesis) to the medical domain rather than to compare or discuss issues found by Lee et al. (2012).

Wiseman, Cox, and Brumby (2013) analysed logs from an infusion pump with a numeric keypad design. Wiseman, Cox, and Brumby (2013) found that the numbers used in medical scenarios are not uniformly distributed, but some digits are more common than others. In chapter 5, a similar study to Wiseman, Cox, and Brumby (2013) is carried out on logs from an infusion pump with a 5-key number entry system. Further log analysis is carried out in this thesis to discuss: how medical practitioners enter numbers; noticed error; and more.

2.5 Mathematical techniques in number entry system design

Safety critical work in software verification of safety-critical systems is a rich area. This area focuses on making sure that the safety critical programs do what they say that they are going to do. Techniques such as model checking (Clarke Jr., Grumberg, & Peled, 1999) and theorem proving (de Moura et al., 2004) are well established in safety critical implementations and have been used for several years.

There are automatic tools that help in the verification of these programs. The tools take two inputs: a mathematical model of the program to be verified; and a description of the property to be verified – generally in a formal logic. The verification tool then either verifies that the property is guaranteed to be true in every trace of the program model or says that the property does not hold everywhere. Some tools such as Symbolic Analysis Laboratory (SAL) (de Moura et al., 2004) and Prototype Verification System (PVS) (Shankar, 1996) give a counterexample of how the property does not hold, by means of a program trace (i.e. a trace in the model) that falsifies the property.

For correct verification of programs, the program to be verified has to be correctly represented in the input format of the automatic verification tool. For program verification to be successful, the program being verified is to be modelled correctly, and the properties being verified should be specified correctly and they should also be properties that we want to verify. Methods such as model checking and theorem proving do not make any conclusions about whether the properties being verified are properties that are required in a system. In the case of designing medical devices, an important research question is what properties do we want to verify using automatic verification tools?

Masci, Rukšėnas et al. have used model checking and theorem proving to verify properties about safety critical user interface design (Masci et al., 2011; Masci, Rukšėnas, et al., 2013). In (Masci et al., 2011) and (Masci, Rukšėnas, et al., 2013), the number entry system of two infusion pumps are modelled and they are mathematically compared to a model of a number entry system that is *'predictable'*, by the definition given by Masci, Rukšėnas, et al. (2013).

The benefit of the work in (Masci et al., 2011; Masci, Rukšėnas, et al., 2013) is that the automatic verification tool used do not only inform whether the properties being verified are true or not within a system. When a property that is being verified does not hold in the system, SAL and PVS provide a program trace to describe how the system does not satisfy the property. This is useful for discussing whether the program should satisfy the property in that trace or for changing to program to satisfy the property.

The work presented in this thesis complements the formal work by Masci, Rukšėnas, et al. (2013). From a formal methods perspective, the analytic method introduced in this thesis can be used to find out what properties should be verified using the mathematical tools, SAL and PVS. Therefore, the outcomes of the analytic method presented in this work are a set of properties that existing mathematical tools can verify about number entry systems.

Thimbleby makes an analytic argument against using seven segment displays in (Thimbleby, 2013b). Seven segment displays are ubiquitous in everyday devices and they are also common in commerical medical devices, such as the Graseby 500 infusion pump. Thimbleby highlights the problems with seven segment displays and makes several analytic arguments against using them. Among other problems, there are practical issues with the visibility and legibility of numbers on seven segment displays. Thimbleby argues that when faults occur with the LEDs of the displays, they can display a different number, for example 8 and 3.

2.6 Conclusions

The United States and in European Union have regulations in place for manufacturers to sell their devices in the relevant countries. This chapter presented the current state of some regulation, standards and guidance provided by the relevant regulation bodies. In regards the HCI and Usability, these regulations enforce standards such as ISO 62366 and ISO 9242. Most HCI principles used in these standards are based on Nielsen (1994), Norman (2002) and Shneiderman et al. (2013), and as highlighted in this chapter, they are not intended for safety critical design.

HCI research so far has focussed on users want and like rather than what they need in terms of safety. Safety critical HCI focuses on tradeoffs between usability, safety and design. In safety critical scenarios safety should be prioritised over how much the users of the devices like the user interface. In safety critical systems, usability should focus on balancing these three factors, as described by Oladimeji (2012):

- **Usability** – The users understanding how to perform tasks and making sure that the intended user group know how to perform the safety critical task. Usability also refers to properties that make the system more pleasant to use.
- **Safety** – This refers to allowing error detection in the interface, checking for syntax errors when the safety critical system requires input and making sure that the error caused by a small interaction does not cause potential of great harm.
- **Design** – This refers to the physical design of the device. The shape, button layout, space requirements, range of inputs covered and the precision of input.

In a medical scenario, if a safety feature is unusable it could do more harm than good. Under time pressure, being unable to set a device because the feature is unintelligible could be fatal.

From these three factors that are important for safety critical HCI, the current study methods available focus on either finding out whether users understand and like the device (interaction and form) or measuring speed. So far there are no methods for measuring safe interaction design.

One of the aims of safety critical design is to reduce the number of errors that occur. In iterative design, a system is developed, evaluated and redesigned (and iterate). The redesign should produce less errors than the first, and errors should be reduced through each iteration. Systems should be designed to have errors that are too few to detect using empirical evaluation techniques. At this point, it is not implied that a design will prevent *all* human error *all of the time.* Analytic and engineering techniques come into play to design systems when empirical techniques and design iterations become impractical.

This thesis presents an analytic analysis approach that aims to reduce harm caused by human error. This approach should be used with current research methods to ensure that users understand the features and perform tasks in required time frames. The next chapter presents the theoretical framework for the analytic approach for safety critical analysis.

Chapter 3

The Differential Formal Analysis method

The core method of the thesis is introduced in this chapter. Chapter 4 applies the method described here in a case study on 5-key number entry systems and raises discussion about the method.

Number entry is required for almost all clinical procedures (e.g., radiation treatment, drug infusion, patient records). Software errors and HCI options can have a significant impact on dependability (i.e., the ability of the clinician to successfully enter the number intended). We are concerned with quantifying error magnitudes in relation to number entry of prescribed *values*: for example an error that is 1% out is less significant than an error that is out by a factor of ten. This is not however the only role that numbers can play in a medical context. Numbers are also used in healthcare as *identifiers* such as patient identifiers, and here different techniques (such as checksums) should be used — an error of 1%, unless detected, will be a completely different patient. An approach beyond the scope of the present work is entering numeric values of *standard* values, which are both identifiers and numeric values (*e.g.* a menu of 10, 20, 50mg).

The method presented here is designed to be useful for studying safety critical, numerical interfaces that have subtle differences in implementation. Oladimeji et al. (2013) carried out empirical lab studies to find out which number entry interface button layout generates less errors, as previously discussed in chapter 2. There are statistically significant differences between designs and the best design, the 5-key interface is less likely to cause errors than other designs. The number of errors in the study (Oladimeji, Thimbleby, & Cox, 2011) are very low. Redesigning the number entry interfaces to reduce these error rates would result in lower error rates, however, if we bring down error rates to be undetectable in lab studies, number entry errors will still occur in hospital scenarios where numbers are entered over a longer period of time. After errors are reduced through iterative design, analytical techniques are necessary to design safer safety critical number entry systems.

The method presented in this chapter is an analytical approach for finding a number entry system that causes least harm when human error occurs. The approach can be used to study different number entry systems, as studied by Oladimeji et al. (2013) or to study subtle program implementation differences between designs with the same hardware layout. Statistically significant results for the best implementation can be established in a short time, and the method is customisable for different interface styles and different types of users.

3.1 The Differential Formal Analysis process

The Differential Formal Analysis (DFA) process starts by determining optional design features that are either implemented or not. A design is a combination of features, and all the combinations make up the design space. Stochastic Key Slip Simulation (SKSS) detailed in Section 3.2, is then used to rank the designs. SKSS entails generating a large number of key sequences that take us from one number to another and inserting a keying error (substitution, deletion, repetition, or key transposition) with probability p per keystroke. If the actual and intended result values differ by more than a magnitude k (say, 10), the error is counted; the designs are ranked according to the proportion of "out by k" errors.

3.2 Stochastic key slip simulation (SKSS)

The process is a stochastic simulation where *slips* are introduced into sequences of key presses in interactive number entry systems, to explore the trade-offs arising from various choices in the design of such systems, as previously developed by Thimbleby and Cairns (2010) for numeric keypad user interfaces.

The basic approach here simulates a human changing the display from one value to another, but allowing for — and analysing the sensitivity of the design to — human error. The user will make keying slips: repetition (or key bounce); transposition (switching two keys in the sequence); deletion (accidentally not pressing a key, or the device not registering a key press); substitution (pressing a different key than intended). These slips are modelled with some probability p per keystroke (which of course may depend on environmental factors), and the proposed designs need to be evaluated for the consequences of those slips. Some design features will make a design more sensitive to such slips: the more sensitive a design, the more likely it is that slips will lead to uncorrected unintended consequences; such sensitivity is best avoided.

It is routine in several safety-critical domains, such as healthcare, to consider "out by ten errors," where the number used is a factor of ten out from the number intended. This suggests a clear measure of design error sensitivity: namely to determine the dependency of out by ten errors on p. This can be achieved by running simulations.

3.3 Setters, solvers and slip models

A simulation is implemented in terms of **solvers**, **setters**, and **slip models**. Solvers and setters occur in matched pairs for each design. A solver generates user key sequences that solve the task in question; a slip model inserts slips into key sequences; a setter executes key sequences.

For number entry analysis, the task is to change the display from showing some number m to displaying n and entering it to the underlying application (hence, the fifth key, $\boxed{\text{OK}}$). Thus, each design's **solver** is a function that takes two numbers

m, n and generates a sequence of key presses to change the display from showing m to show n within the constraints of the chosen design, and then submits it by pressing $\boxed{\text{OK}}$. In the special case $m = n$, the solver generates $\boxed{\text{OK}}$.

Different approaches to user modelling are possible: the simplest (conceptually, if not in terms of implementation) is to compute the optimal sequence for the given task. In reality, users tend to find *satisfactory* rather than optimal solutions, and some truly optimal sequences may be cognitively too hard to determine (Simon, 1996). Therefore, the approach as used in this work finds the sequence with either monotonic left-to-right or right-to-left cursor motion, a realistic satisficing strategy. Complementary empirical studies would give better insight into how users enter numbers, however this satisficing solver is sufficient for the purpose of demonstrating the process.

In contrast, each design's **setter** takes an initial value m and a sequence of keystrokes σ, and returns a triple, $r = \langle r_v, r_n, r_e \rangle$, the result of applying σ to the system from the starting point m. r_v is the actual final value reached, r_n is the intended/target value, and r_e is true if an error occurred and was blocked — where blocking an error refers to blocking interaction and alerting the user when a user action does not result in a change in the display.

Some setters do not block errors under any circumstances (e.g., this is how some 5-key designs work). No blocked error does not imply $r_v = n$, since $\neg r_e$ means no error was *blocked* by the setter, not that no error occurred — for example, slips could occur when a different number is entered 'correctly' without triggering a blocked error.

A **slip model** generates a sequence of keystrokes with keying slips; specifically, given a key sequence σ and a probability p, $\mathit{slips}(\sigma, p)$ inserts slips at each keystroke of σ, each with independent probability p. The slip model is the way the analysis models human error. Then, for some device design d,

$$\mathit{set}_d(m, \mathit{slips}(\mathit{solve}_d(m, n), p))$$

"tries" to set the display to n given that it initially shows m. As p increases, this becomes increasingly unlikely. By using different setters, the idea is to find out which combinations of design features make the setter robust against errors introduced as a consequence of the slips that the slip model introduces.

3.4 Error sensitivity

Given a design d, a set of probabilities P, and a set T of task pairs (m, n), an **experiment** calculates for each $p \in P$ the set $S_{dT}(p)$ of results of running the appropriate solver, slip model and setter on the various (m, n) pairs:

$$S_{dT}(p) = \{set_d(m, slips(solve_d(m, n), p)) \colon (m, n) \in T\}$$

The **error sensitivity** $e_d(p)$ for a design d at p is then:

$$\frac{|\{r \in S_{dT}(p) \colon \neg r_e \wedge (r_v \geq k r_n \vee r_v \leq r_n/k)\}|}{|\{r \in S_{dT}(p) \colon \neg r_e\}|}$$

Take $k = 10$. That is, from each set of samples, the number of non-blocking paths that resulted in out-by-ten error are counted and divided by the total number of non-blocking paths. Note that *valid* random values of m and n are required, which may depend on the application.

The error sensitivity of a design may then be investigated, and compared with other designs. A better design is less sensitive to error, but since the error sensitivity depends (as defined) on p, a simpler measure is the mean gradient $de_d(p)/dp$ around typical p values (e.g., $p \approx 0.001$); in fact, for the case studies, for typical p, sensitivity is nearly linear, and hence effectively independent of p. A lower gradient is better.

Because of linearity, the best design decisions do not depend on the actual value of p; it is not necessary to perform experiments to determine p. In an important sense the resulting design is more robust — as it does not depend on specific assumptions of user performance (as measured by p). Similarly, if the results are established to be broadly independent of the mix of slip types, there is no need to model the user more realistically.

Chapter 4 demonstrates Differential Formal Analysis using SKSS using 5-key number entry. Out-by-ten error is used as the measure of dependability and this is particularly relevant to the chosen application of medical number entry.

3.5 Discussion

This work is motivated by the vast interaction differences between implementations of number entry systems in popular, commercial medical infusion pumps. The 5-key number entry system (see figure 4.1) is gaining popularity in infusion pumps from leading manufacturers such as BBraun and Zimed and analysing these pumps resulted that the same keying sequences in apparently identical number entry interfaces result in very different outcomes.

Consider the case where the starting screen displays 0 and our goal is to input a dose of 950 mL; on one commercial pump the key sequence ◄ ▲^{5} ◄ ▼ results in a display of 950 — but keying in the same sequence starting from the same state on a different commercial pump results in 000.1. Figure 3.1 shows how the user interface of both infusion pumps behaves with each keystroke.

A detailed and formal description of why this happens may be found in (Masci, Rukšėnas, et al., 2013). The main difference in the case of how the BBraun Infusomat Space works is that it uses a feature referred to as *memory* in (Masci, Rukšėnas, et al., 2013). The memory feature results in unpredictable results (by the definition of predictability defined in (Masci, Rukšėnas, et al., 2013)) and the paper concluded that devices that users perceive to be predictable and reliable should not implement the *memory* feature. Different interaction design choices lead to different values on the display (see figure 3.2 for some examples, and the case study in chapter 4); the aim is finding the best combination of choices (along with their rigorous specifications) to make the design more resilient to human error.

The method presented in this chapter does not determine which physical layout or button layout is best for a device. If empirically informed, this method can cater to the physical aspects of the device. Consider a device that has 5 buttons, if two of the buttons are very close to each other while the others are further apart, it is possibly more likely for the two buttons that are near to each other to be substituted

Key Press	Zimed AD	BBraun Infusomat Space
	$\underline{0}$	$\underline{0}$
◀	$\underline{0}0$	$\underline{0}0$
▲	$\underline{1}0$	$\underline{1}0$
▲	$\underline{2}0$	$\underline{2}0$
▲	$\underline{3}0$	$\underline{3}0$
▲	$\underline{4}0$	$\underline{4}0$
▲	$\underline{5}0$	$\underline{5}0$
◀	$\underline{0}50$	$\underline{0}50$
▼	$\underline{9}50$	$\underline{0}00.1$

Figure 3.1: A key sequence being input into Zimed AD and BBraun Infusomat Space from the same key sequence. This table shows the change in displays after pressing the key in the "Key Press" column. In the first row there is no key press to show the starting displays of the two devices.

Design choice	Press	Display
Arithmetic	▲	$0\underline{9}$ → $1\underline{0}$
Independent dial	▲	$0\underline{9}$ → $0\underline{0}$
Wrap	▶	$0\underline{9}$ → $\underline{0}9$
No wrap	▶	$0\underline{9}$ → $0\underline{9}$
Left start		$\underline{0}9$
Right start		$0\underline{9}$
Block underflow	▼	$0\underline{0}$ → $0\underline{0}$
Underflow & arithmetic	▼	$0\underline{0}$ → $9\underline{9}$
Underflow & independent dial	▼	$0\underline{0}$ → $0\underline{9}$

Figure 3.2: Examples of design choices. Note how underflow/blocking and dial/arithmetic interact. Different interpretations of these and other feature interactions affect the sensitivity of the user interface to user error.

for each other when keying in data. The probabilities of substitution errors for these buttons can be manipulated in this method to find out which implementation is safer for the device.

3.6 Conclusions

In safety critical domains, analytical techniques should be used with empirical techniques to design safe and accurate interaction. Previous empirical studies have shown that error rates from human number entry experiments are very low. To obtain statistically significant results when choosing the best implementation of one type of number entry interface, empirical techniques are impractical (since millions of numbers would need to be entered) and analytical techniques are the way forward for this purpose.

Differential formal analysis was introduced in this chapter. To provide rigorous support for safety critical design. The SKSS method simulates a user performing number entry tasks and simulates slips with a determined probability. The resultant value from the task is compared to the intended value and if greater than a set error magnitude, k, it should be counted. The design with the least out by k errors is the best design.

This method is demonstrated through a case study in chapter 4 where the case of 5-Key number entry is used. Chapter 4 details how 5-key number entry systems should be implemented for safety critical cases. The method raises several empirical questions. The Differential Formal Analysis process can be used to determine whether running the empirical experiments is important or not depending on whether the results from the empirical studies change the results from the Differential Formal Analysis. Examples of how Differential Formal Analysis can be used to determine whether or not to run an empirical study are shown in chapter 4 and other empirical findings that are determined to be important are used to empirically inform the Differential Formal Analysis process in chapter 5.

Chapter 4

A Differential Formal Analysis case study – 5-key number entry

5-Key number entry interfaces are used in infusion pumps for drug dose entry. This type of interface (shown in figure 4.1) has a display that shows the number being input and a cursor that highlights a digit that is being edited. Four arrow keys are used to edit the number and an $\boxed{\text{OK}}$ key submits the number.

Two infusion pumps on the market that use 5-key number entry for entering drug doses are the Zimed AD and BBraun Infusomat Space pumps shown in figures 4.2 and 4.3 respectively. Using 5-keys for number entry is attractive to manufacturers of medical devices because they are less expensive to manufacture than, say, number keypads. The main focus of this thesis is reducing harm caused by number entry errors and Oladimeji et al. (2013) show that 5-key interfaces have the least number of errors in a study comparing 5 types of number entry hardware layouts.

Oladimeji et al. (2013) use a lab experiment find the best hardware layout of a safety critical number entry system and this case study advances that work by showing that there are several ways of programming 5-key interfaces and finds the best way of programming the number entry interface that produced the best results in (Oladimeji et al., 2013). As a case study for the method presented in chapter 3, Differential Formal Analysis is applied on 5-key number entry systems to find the system that is least sensitive to human error.

This chapter described how Differential Formal Analysis was applied to 5-key

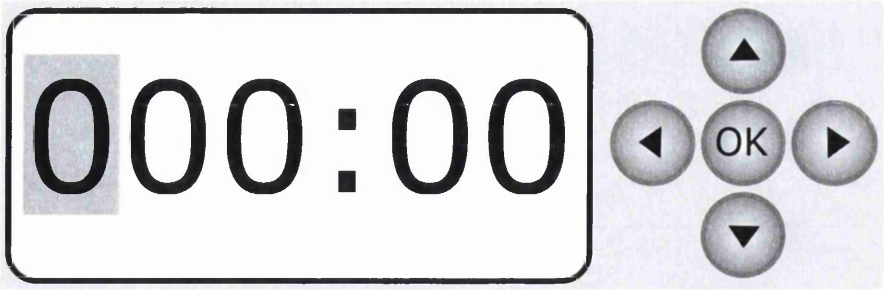


Figure 4.1: An example of 5-key user interface layout. Here the cursor is shown in the left-most position, and the display format is suitable for entering times, 0 minutes to 999:59 hours. Some 5-key interfaces omit the OK button as its use can be implied by the user performing any action with any non-arrow button.

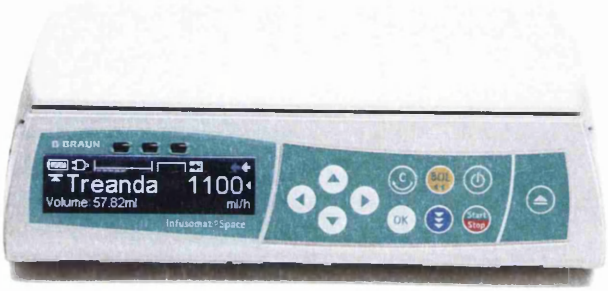


Figure 4.2: A BBraun Infusomat Space pump; a widely used drug infusion pump

number entry systems and presents results on how to best program this type of interface. Primarily, design features of 5-key interfaces are identified and described, then how differential formal analysis was implemented is detailed and finally, results are presented. The differential formal analysis raises empirical questions and they are also discussed in this chapter. The work in this chapter has been done by myself duplicated and discussed with three other researchers: Harold Thimbleby, Andy Gimblett and Paolo Masci for validation. This process makes this work scientifically rigorous and it has been key in improving number entry design safety.

4.1 Design features

The design features of 5-key number entry systems were derived by studying infusion pumps from different manufacturers. There are implementation variations relating to how the cursor works and how the digits work. These design variations make up the design features and are described here.

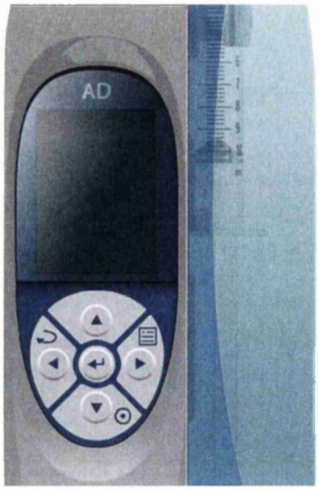


Figure 4.3: A Zimed AD pump

- **Left or Right Start** – In the starting screen, is the cursor on the left or on the right?

- **Cursor Wraparound** – When the cursor is at a display edge (leftmost or rightmost position), what happens when going beyond that edge? Does it wraparound to the opposite edge or stay at that edge?

- **Digit Wraparound** – When reaching the minimum or maximum number for a particular digit (0 or 9) what happens when attempting to go beyond it? Does it wraparound to the other edge or stay on the same number?

- **Arithmetic** – Another option for going beyond the edge for a particular digit is doing simple arithmetic operations. If at 09 and ▲ is pressed does it show 0 and add 1 to the next digit? (Therefore having a display showing the 10 ?)

- **Block errors** – If a user action does not change the interface, block interaction and alert the user.

A system that alerts the user to slips might be seen as less forgiving, but the payoff is (potentially) greater resilience. If the interface displays an alert, blocking further interaction until it is cleared, the user will usually become aware of the problem, and can recover from it. For example, suppose a user's task of "get from m to n" is interrupted because of a detected error, with the display in state m'; it is then reasonable to suppose that the user effectively abandons their old task, and adopts a new one, namely "get from m' to n."

4.2 Differential Formal Analysis implementation

SKSS was implemented in C# to analyse number entry designs with the design features described above.The probability of out by 10 error for each design was used to determine a rank.

The main points raised by this process were about how users key in numbers and how the features behave at the boundaries. To simulate number entry, we considered finding the best possible path from one number to another but we found that programming solvers for the best path was complex and it is unlikely that a user enters numbers in this way. We agreed to implement realistic, simple solvers that consider shortcuts users are likely to take; however these raised issues are worth studying empirically, especially since the domain is safety critical.

4.2.1 Solvers

Four solvers were implemented and a detailed description is given here. The solvers take number pairs *(a,b)* as parameters and generate a key sequence that changes the display from number a to number b. These four solvers were tested using 10^6 number pairs to ensure that they generated the correct key sequences with no errors. The four solvers are:

- Digit wraparound for a cursor left start interface
- Digit wraparound for a cursor right start interface
- No digit wraparound for a cursor left start interface
- No digit wraparound for a cursor right start interface

A solver for arithmetic has not been implemented and the solvers with no digit wraparound are used for the designs that implement arithmetic (choosing left and right start depending on the design). This assumes that users do not use the arithmetic feature or are unaware of it. Chapter 5 describes an empirical analysis of strategies used by clinicians in a hospital and the results indicate that this was a plausible assumption. The device that has an arithmetic design did not show any log of the arithmetic feature being used.

Broadly, there are two distinct solvers with a left and right solver for each. The difference between left and right start solvers is that left start is used on a left start interface and it keys in a sequence going from left to right where as the right start solver works symmetrically.

The digit wraparound solvers are used for interfaces that have digit wraparound enabled. A piecewise subtraction is performed on the number pair that is given as input to the solver and on each digit, the following logic is used to build the key sequence:

Let *d* be the piecewise difference result for the particular digit. Let *m* be the ceiling of maximum possible digit divided by 2. That is, if the maximum possible digit is 9, *m* would be 5. Let the maximum possible digit be *mD*. The keySequence.AddRange function adds key presses to the temporary key sequence and the repeatChar function repeats a keyPress for the desired number of times.

Listing 4.1: Digit wraparound solver logic

```
if ((d <= m) && diff[i] > 0) //e.g 2 -> 4 = UU
{
  keySequence.AddRange(repeatChar("U", d));
}
if (d > m) //e.g. 2 -> 9 = DDD
{
  keySequence.AddRange(repeatChar("D", mD[i] - d));
}
if (diff[i] <= -mD) //e.g. 9 -> 2 = UUU
{
  keySequence.AddRange(repeatChar("U", mD + d));
}
if ((d < 0) && d > -mD) // e.g. 7 -> 4 = DDD
```

```
{
  keySequence.AddRange(repeatChar("D", -d));
}
```

Therefore here are some examples of how this logic works.

- If a digit 2 should be changed to 4, UU is added to the key sequence
- If 2 should be changed to 9ok :), DDD is added to the key sequence
- If 9 should be changed to 2, UUU is added to the key sequence
- If 7 should be changed to 4, DDD is added to the key sequence

Solvers that do not have wraparound work in the following way. Let d be the piecewise difference result for the particular digit.

Listing 4.2: No wraparound solver logic

```
if (d < 0) {
  keySequence.AddRange (repeatChar ("D", -d));
}
if (d > 0) {
  keySequence.AddRange (repeatChar ("U", d));
}
```

Therefore, if the digit to be entered is greater than the value on the current display, the correct amount of up buttons are pressed and if the digit is smaller than the currently displayed digit, the correct number of down buttons are keyed.

4.3 Results

Figure 4.4 shows an example of dependence of error sensitivity on whether a feature is on or off. Block errors is the feature that most improves the design, followed by left start, cursor wraparound, arithmetic and vertical wraparound.

One empirical question raised from the Differential Formal Analysis process is *does the probability of keystroke error matter?* Figure 4.5 shows the results of 10

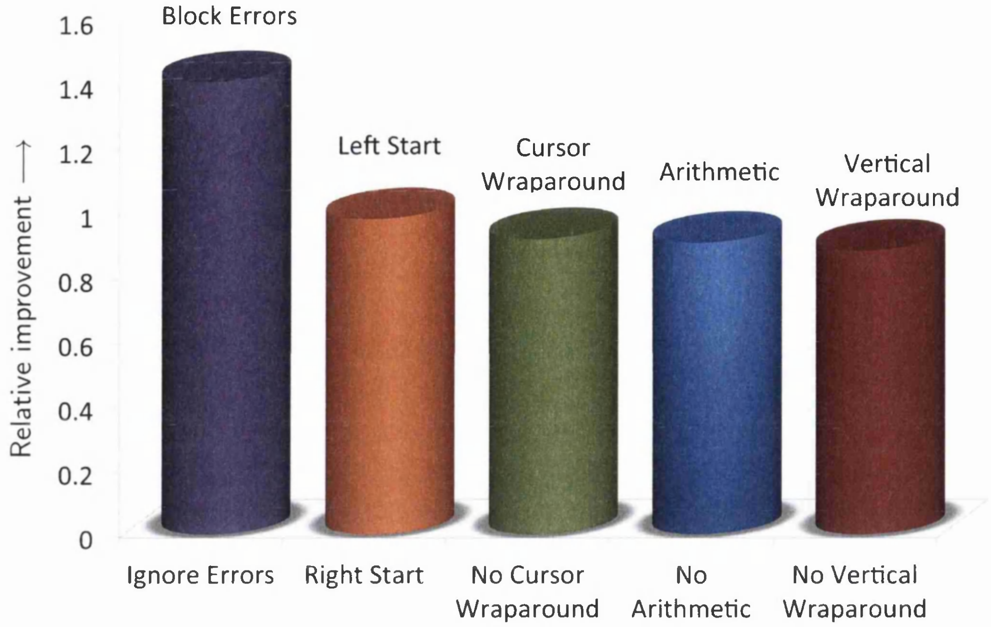


Figure 4.4: Summary of C# analysis findings. Features mentioned at the top of the bars are better, by the factor shown. Specifically, over all design choices, blocking errors reduces the mean sensitivity of the 5-key interface by a factor of 1.41, start at left is better than start at right by a factor of 0.98, cursor wraparound is better than no cursor wraparound by a factor of 0.91, arithmetic is better than no arithmetic by a factor of 0.9 and vertical wraparound gives us an improvement over no vertical wraparound by a factor of 0.88.

SKSS trials with varying keystroke slip probability varying between 0.0001 to 0.001 with a step of 0.0001. Each trial was run with 10^5 number pairs and the keystroke errors were chosen randomly from transposition, deletion, repetition and substitution with each error type having the same probability of occurring. Figure 4.5 shows typical results, plotting sensitivity against keystroke slip probabilities. Because the sensitivity/keystroke error probability has an excellent linear correlation ($R^2 = 0.995$ or better) the rank order — and hence the recommended design decisions — do not depend on the value of p; in other words, doing an experiment to determine p does not seem necessary.

Another important question is about the particular mix of types of errors. The parallel coordinates plot in figure 4.6 shows the relation of how the ranking scores for each different design change depending on what type of slips we have in SKSS. The vertical lines in the diagram represent each slip error type present in the ex-

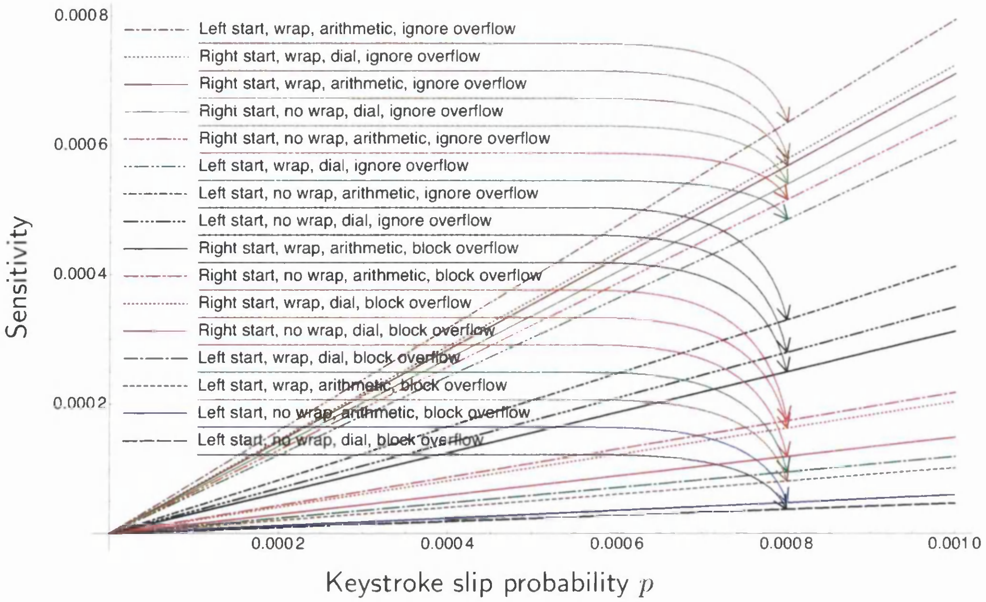


Figure 4.5: Sensitivity against p for various combinations of design feature (from Mathematica). The lower gradient lines represent better (less error-sensitive) designs.

periment: all slips with equal probability of being chosen; transposition errors only; deletion errors only; repetition errors only; substitution errors only. Each design is represented by a polyline passing through each of the vertical lines and the position of each vertex on the a line corresponds to the score value of each design. This visualisation suggests that designers should prioritise improving designs for repetition errors since the worst designs for the repetition errors experiment have the highest gradient when compared to the other experiments and the best designs in repetition errors have the best gradients when compared to the other experiments.

For most of our results we ranked our designs based on their resilience to out by 10 errors. We ran an experiment to find out what happens if we consider other magnitudes of error and we can see the relation of rankings in figure 4.7. Here we have a parallel coordinates visualisation that shows how the probability of error (on the vertical axes) changes when considering different out by k errors. Each polyline in the diagram represents a combination of design features. The lower the polyline across the vertical axes, the smaller the sensitivity of the design to keystroke errors (and, thus, the better the design). The interesting find in this trial is that the best four designs remain toward the bottom of the visualisation and we do not have interleaving between the best designs. There are some interesting interleavings in

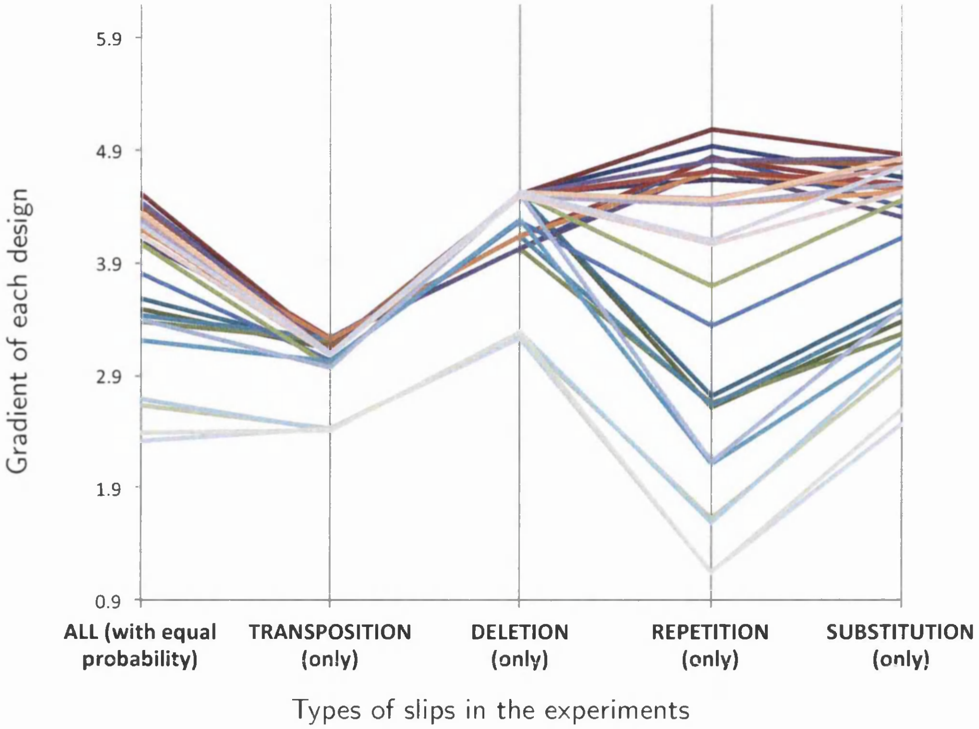


Figure 4.6: Parallel Coordinates visualisation showing the relation of design rankings depending on slips

the worse designs, however we are not interested in this since those designs with significantly higher sensitivity should not be implemented.

4.4 Discussion

Applying the Differential Formal Analysis process on 5-key number entry systems raises questions that would not have been raised otherwise. One can discuss the different features, the simulation method itself and the errors and combinations of errors.

One important empirical question that has not been addressed in this chapter and not addressed in this thesis is whether users understand the particular design features. Although the arithmetic feature seems to be a good idea from these results, if users do not understand the feature, it might be less safe. It is important for empirical trials to be carried out to address this question.

Another important feature to be studied is the block errors feature. Blocking errors is clearly important from the results presented in this chapter, but how can

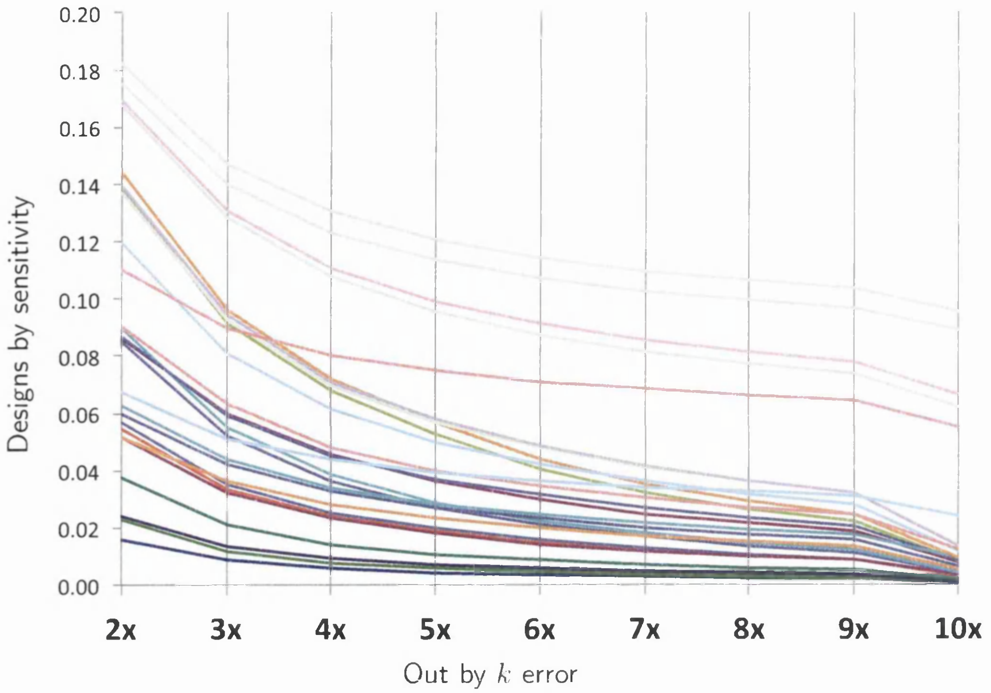


Figure 4.7: Out by k errors for different design types.

we block errors such that it works well in clinical settings? This is an important question to be explored and an important feature to be implemented in safety critical number entry.

After reading this chapter, one may also start thinking about new features that can be studied. This is one of the benefits of applying the Differential Fornal Analysis process and it is indeed important for a safety critical system to be well thought out. New features can certainly be created, it is important to explore these new features both to make sure that they are understood by users and to find the best combination through Differential Formal Analysis .

In this chapter, the results for 5-key number entry are derived using random numbers. Wiseman, Cox, and Brumby (2013) argue that the numbers used in clinical scenarios are not random. Finding out what types of numbers are used on 5-key devices in hospitals and whether they change the results derived from this chapter is a useful question to answer. This work is presented in chapter 5.

How solvers are implemented is another point of concern. How do we know that the solvers implemented in this chapter are indeed realistic? Chapter 5 describes how logs from 5-key devices used in a hospital were collected and number entry strategies were analysed. These findings were then used to inform the differential formal analysis process and the differences in results are discussed.

4.5 Use in procurement

Procurement is a dual of design: alternative designs are evaluated to choose which ones to purchase, whereas in design, evaluation informs which features to implement or improve. Procurement is concerned with evaluating finished products, so it would be natural to reverse engineer them and apply the proposed methodology to search for critical features that may distinguish the products on offer. However, because the companies selling the devices offered for procurement would then have a financial stake in the outcome, it is likely that an extra step would be feasible: the manufacturers could either provide models for evaluation (though this may accidentally leak proprietary information), or they could help procurement check the validity of their models (this better controls access to proprietary information). The approach proposed in this thesis could give procurers a way to evaluate which interface design works best towards reducing number entry error rates.

4.6 Empirical questions raised

From figure 4.4 we see that starting at the left is less sensitive to error than starting at the right. A possible explanation for this is that if, as a user performs a sequence of keystrokes that move from right (least significant) to left (most significant), then as errors accumulate the overall numerical effect gets larger and larger; from left to right, the opposite is true. On the other hand, perhaps in reality, starting on the far left raises the risk of confusion about which column the user is modifying — so in practice, it might turn out that right-to-left is the best strategy overall. Thus the empirical question: what do users do, and do they make slips we are not yet modelling when they start on the left or right?

We devised suitable and realistic strategies for our solvers, but there is no evidence that they accurately reflect how users find paths between numbers in 5-key interfaces. In particular, when wraparound and arithmetic are available, do users take advantage of them to reduce path length? Can they reliably find the shortest path, or is this strategy too complex to employ in reality?

The feature that we call "block errors" is important to implement, as we see from figure 4.4. A related empirical question, then, is how should "block errors" be implemented to best alert the user of a possible keying error? Do we stop interaction and ask the user to start over? Do we alert (through sounds, vibration, flashing etc.) and let the user continue entering the number? There are probably other ways of doing this: it is worth finding out and getting it right.

4.7 Conclusions

In this chapter we have seen how the Differential Formal Analysis process presented in chapter 3 can be applied to one type of safety critical system. We take the case of 5-key number entry since previous studies by Oladimeji et al. (2013) indicate that the 5-key number entry layout is least susceptible to errors from an empirical trial that compares five different button layouts for number entry.

The features for 5-key number entry systems were described and how the differential formal analysis process was implemented was detailed. Empirical questions can be raised about whether users would understand the features presented and whether users were simulated realistically in the process. Users' understanding of the features has not been explored in this thesis, yet it is an important step for future work.

Results about how best to implement 5-key number entry systems are presented. The most important feature to implement is the block errors feature. Although we know that it is important to block errors, we have not yet studied how it is best to do this in clinical settings. Finding this out is out of scope for this thesis but certainly important.

Empirical questions regarding the probability of keystroke error and the mix of errors that occur were raised. Before running time consuming and costly user trials to find out the answers for these questions, we ran the Differential Formal Analysis process to find out whether the answers to these questions matter. Our results show that these questions do not matter when running Differential Formal Analysis and thus saving time and money for running the experiments.

In the next chapter, medical device logs were analysed to answer some empirical questions raised by the Differential Formal Analysis process. We will see what types of numbers are entered into 5-key infusion pumps in a hospital setting, what strategies clinicians use to enter values and also have a look at what errors occur. The results from the infusion pump log analysis are used as parameters in the SKSS method and the results from running SKSS will be domain specific for medical infusion pumps.

Chapter 5

Using medical device logs for improving medical device design

A systematic analysis of combinations of features of a 5-key number system was presented in chapter 4. This chapter indicates the scale and types of quantitative errors to which the combinations are susceptible and is used as a starting point to investigate the *actual* error behaviour relating to one or another particular combination of design features. To do this, keystroke and event logs — taken from current infusion pumps in hospital use — are used to inform and extend the Differential Formal Analysis process, thus grounding it in empirical evidence. The logs combined with the analysis raise suggestions about how manufacturers could improve log mechanisms to further help interaction design, and hence improve device safety. By this means a detailed understanding can be obtained of the predictive power of the formal analysis. The broader aim is to produce a rigorous evaluation approach that can be used effectively in design without the need for extensive user trials.

Medical device logs have been analysed by Lee et al. (2012). Lee et al. (2012) present valuable insight into how infusion pumps are used in hospitals, how we can gain insight into hospital infrastructure, how much it costs for nurses to attend to alarms over a period of a year and other interesting findings that can significantly improve healthcare systems. In this thesis, there is a different focus on retrieving information from logs. The concern here is rather to answer very specific empirical questions to tailor the Differential Formal Analysis method to the medical domain rather than to compare or discuss issues found in (Lee et al., 2012).

Section 5.1 describes the methods used to analyse the log data retrieved from medical devices used in a hospital. The section describes in detail how *noticed errors* were detected, the distribution of numbers used and the strategies employed by the users for entering numbers. Section 5.2 presents the findings from the use logs after implementing the 5 key methods described, and presents results related to noticed errors.

Section 5.4 discusses how the empirical data were used to focus the 5-key analysis and its relevance to the medical domain. Section 5.5 discusses the results obtained by performing an empirically grounded analytical evaluation and compares it to previous findings.

Section 5.6 discusses the implications of this chapter; and finally, Section 5.7 draws conclusions about what was learnt from this chapter.

5.1 Medical device log analysis method

The medical device logs of 19 BBraun Infusomat Space infusion pumps from the device library of the Royal Free Hospital in London were retrieved. They are in everyday use in the Royal Free Hospital in a range of contexts and reflect typical use. The BBraun Infusomat Space is a volumetric infusion pump model that has 5 buttons to enter numbers: ▲ ▼ ◄ ► and OK. The number entry system is implemented so that the cursor starts at the right, the digits have simple arithmetic and there is no cursor or digit wraparound. The block errors feature is not present in this infusion pump.

BBraun Infusomat Space pump logs are split into two files. The first file (the device logs file) logs input at an event level while the second (the keystroke logs file) logs input at a keystroke level. The device logs file contains event log records with the following field structure:

- **Number** – An identifier for the event log.

- **Date** – The date when the event happened in DD/MM/YYYY format.

- **Time** – The time when the event happened in HH:MM:SS format.

- **Event** – A description of the event (e.g., "New VTBI set").

- **Value** – The value of the event in the form of either a number (e.g., 100) or a textual description (e.g., "ON"). Whether the value is a number or a textual description depends on the Event field.

- **Unit** – The unit of the value. When the Value field is a number, the Unit field would specify the unit of that value (e.g., mL), otherwise the Unit field is left blank.

The log records were in the following structure:

- **Number** – An identifier for the key log.

- **Date** – The date when the key was pressed in DD/MM/YYYY format.

- **Time** – The time when the key was pressed in HH:MM:SS format.

- **Key** – The key pressed (e.g., Arrow_Up).

- **Menu** – The menu in which the device was in when the key was pressed (e.g., "MAIN").

The first log relating to the 19 devices is from 22/03/2009 and the last log is from 13/12/2012. The devices have been in use in the hospital for 1,362 days, just over 3 years and 8 months worth of data. The sum of the total time of use of all of the devices is 4,368 days, almost 12 years in all, though the restriction to only the last 200 keystrokes limits the actual duration of user interaction available for analysis.

5.1.1 Noticed errors

The log data provides a source for occurrences of number entry error. The log data does not provide information about whether the input value is correct or not. For this to be possible it would be necessary to compare the log data with the doctor's prescription for the infusion. This was not possible both because prescriptions contain sensitive patient data and anonymized prescriptions are not available, and because of the difficulty of correlating prescriptions with this large source of data.

However the logs can be used to detect *noticed errors*. These errors can be observed in a log when a number is entered and apparently corrected by entering another number a short while after. These corrections are made rapidly, suggesting that the user is correcting a slip rather than correcting a mistake. Strictly speaking, it is also possible that the corrected number is erroneous — that is, the user made a mistake in performing the correction. These noticed errors understate the sum of number entry errors, however they provide data about a subset.

The time frame related to noticed errors is set to be 2 minutes in the analyses described here. That is, input values are considered erroneous if they are set less than 2 minutes apart until the value stops being changed within 2 minutes after the previous value was set. This time frame was chosen because it is a reasonable time frame to notice and correct an error. A time frame of 1 minute was previously considered, but in the case of Listing 5.2, the correct value seemed to be 0.80 mL which we suspected was not correct. The time frame was increased to 2 minutes and got what we see in Listing 5.2. 1000.00 mL is a more reasonable number for an infusion. We considered 3 minutes to see whether the number of errors changed but they did not so for this case, 2 minutes is reasonable.

Noticed errors give some insight into error magnitudes and what caused the errors and it is also interesting to find out how frequently near misses occur. Noticed errors are defined to occur in the logs when values are set a small time frame apart. Listing 5.1 is an example of two log entries. The first input value is considered to be a noticed error, and the second value to be the number that was intended.

Listing 5.1: VTBIs set together less than a minute apart

```
1891, 25/10/2012, 12:02:48, New VTBI set, 10, ml
1892, 25/10/2012, 12:03:04, New VTBI set, 100, ml
```

In the case of Listing 5.1, it can be seen that a new VTBI is set 16 seconds after the first VTBI. The error that is noticed has a magnitude of 10 (also known as an out by 10 error). At the keystroke level, it may be assumed that the clinician missed pressing the "left" key.

There are instances in the logs where several VTBIs are set in quick succession, for example in Listing 5.2.

Listing 5.2: More than one error in quick succession

```
1538, 29/10/2012, 19:32:10, New VTBI set, 1.00, ml
1539, 29/10/2012, 19:32:44, New VTBI set, 1.07, ml
1540, 29/10/2012, 19:33:06, New VTBI set, 1.87, ml
1541, 29/10/2012, 19:33:33, New VTBI set, 1.86, ml
1542, 29/10/2012, 19:33:59, New VTBI set, 0.80, ml
1543, 29/10/2012, 19:34:59, New VTBI set, 1.00, ml
1544, 29/10/2012, 19:35:57, New VTBI set, 10000.00, ml
1545, 29/10/2012, 19:36:26, New VTBI set, 1000.00, ml
```

Listing 5.2 appears to indicate that the first 7 input VTBIs were erroneous and that the correct VTBI is the last one entered (1000.00 mL). To determine error magnitudes, the first 7 input numbers are compared to the final, which is considered to be the intended number. This gives insight into how large the error would have been had it been unnoticed, and what keystrokes were erroneous in the first keystroke sequence.

5.1.2 Numbers used in infusions

The method described in chapter 3 generates random numbers as a means of exploring the probable errors generated by the different designs. As already mentioned and argued in (Wiseman, Cox, & Brumby, 2012) the number entry distribution in infusion pumps does not follow the rectangular distribution of the random numbers generated in chapter 3. Noticed errors are used to assess the error distribution in the logs. Numbers that are near misses are not considered, and numbers that end in ".00" are truncated to integers.

5.1.3 Number entry strategies

The keystroke logs also provide insight into the strategies that practitioners use to enter numbers into the infusion pumps. The BBraun Infusomat Space logs are spit into two files. The rates and VTBIs recorded in the device log file were used to find the corresponding sequence of keystrokes that were needed to input that number in the keystroke log file by comparing the date and time of the logs. To assess number entry strategies the VTBIs and Rates entered were considered irrespective of whether a noticed error occurred. It was not possible to identify individuals and therefore individual strategies. The results therefore ignore the possibility of individual differences. All the logged strategies employed by clinicians were identified before counting how many times the particular strategy was used. From this the probability that a particular strategy be used was derived.

5.2 Medical device log analysis results

The methods described in Section 5.1 were used to derive the noticed error, digit and number distributions and number entry strategies. This section discusses the results obtained from the analysis before describing the implications of the results.

5.2.1 Error Analysis

The total number of VTBIs set in all 19 devices together was 1,409 and the total noticed errors detected by our method was 103 which gives us an error percentage of 7.13% of inputs. The total number of Rates set in all 19 devices together was 1,171 and the total noticed errors detected was 198 giving a percentage error of 16.91%. The error rates are different between VTBIs and Rates because they are both set individually on the device.

The devices did not have the same percentage of error rates and figure 5.1 shows the percentage error rates for Rates and VTBIs for each individual device. Errors in entering Rates are higher for 17 of the 19 devices.

Although there are higher errors for rates, rates have more small errors and in

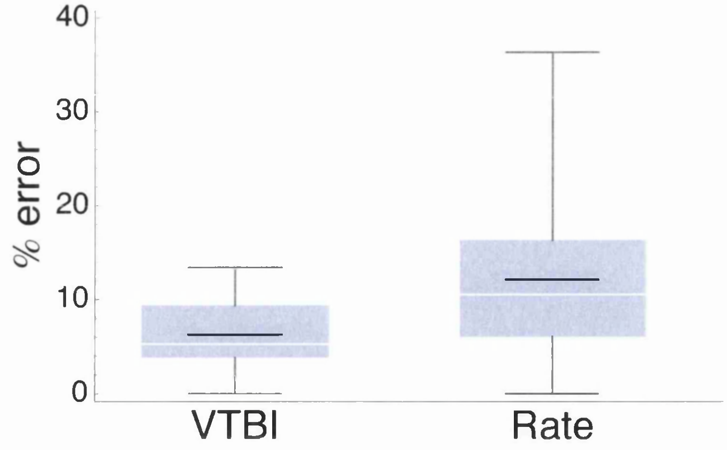


Figure 5.1: Frequency of noticed errors, as percentages for VTBI and Rate input. The graph shows the mean (black bars), median (white bars), 25% quartiles (top and bottom edges of box), and range (top and bottom bars) of the $N = 19$ device logs.

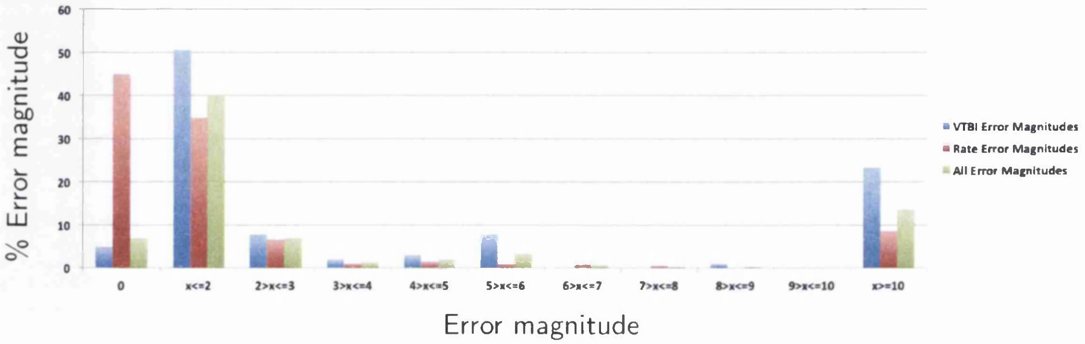


Figure 5.2: The x-axis of this bar graph shows the error magnitude and the y-axis shows the percentage frequency of each error magnitude. The frequency of the error magnitude was divided by the sum of errors and multiplied by 100. The graph shows the percentage frequencies for VTBI errors, Rate errors and combined VTBI and Rate.

VTBIs, whereas VTBIs have a larger number of higher magnitude errors, specifically errors out by 10 (or more).

5.2.2 Digit and number distributions

The frequencies of individual digits were determined by considering only those numbers entered that were not noticed errors. Figure 5.3 shows a bar graph of the frequency of occurrence of digits 0–9 and decimal point for VTBIs only, Rates only, as well as both VTBIs and Rates together. The digit 0 is the most frequently input digit and 4, 7, 8 and 9 are relatively rarely entered.

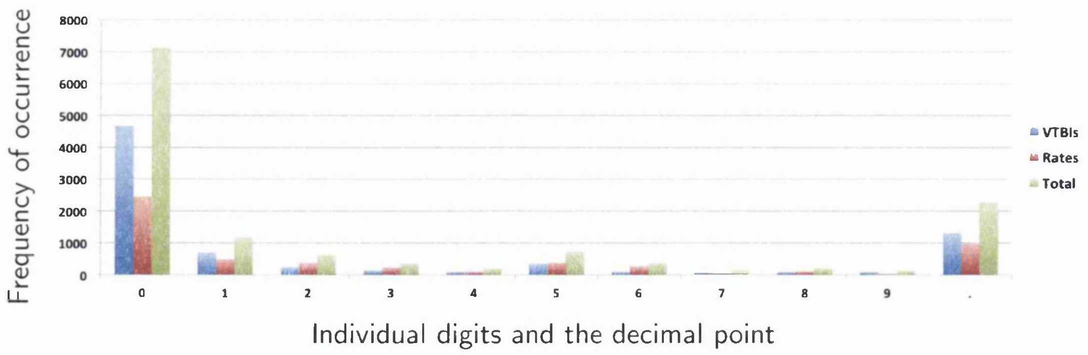


Figure 5.3: A bar graph showing the frequencies of occurrence of digits in VTBIs, Rates and All VTBIs and Rates combined. The noticed errors were removed from this analysis.

5.2.3 Number entry strategies

Since the BBraun Infusomat Space logs only store the last 200 keystrokes, it was only possible to retrieve 68 number entry keystroke strategies from the numbers entered. Of interest is how clinicians go about entering the number. The features of the BBraun Infusomat Space 5-key number entry interface (see Section 4) relevant here are:

- No digit wraparound
- No cursor wraparound
- Arithmetic
- The cursor starts on the *right*
- No error blocking

Strategies relating to 68 numbers were considered.

Arithmetic

Although the BBraun Infusomat Space has an arithmetic feature, no strategy appears to take advantage of this feature. For example, to enter the value 9, from a screen showing 0 it is possible either to press ▲ nine times or to key in ◄ ▲ ► ▼. In other words 10 is first input then units with ▼ is used to subtract 1 from the value displayed. Although using the arithmetic feature reduces the number of keystrokes in a lot of the numbers input, clinicians have not taken advantage of it.

Left and right start

The BBraun Infusomat Space number entry system has a right start cursor position. Instances were observed of practitioners entering numbers such as 955 (from the starting screen of 0) by first pressing ◄ three times then entering the number from left to right. This means that a left to right input strategy is used on an interface which starts on the right.

From 60 of the numbers that were entered only 1 digit needed changing. For example when the number 100 is entered on a starting screen of 0, the key sequence used is ◄ ◄ ▲. It is impossible to tell in these cases whether the practitioner intended to use a left-to-right input strategy or a right-to-left input strategy because to be able to tell this, at least two digits need to be changed.

From the rest of the eight keystroke logs of entered numbers, 5 numbers were entered using a left-to-right strategy and 3 numbers were entered using a right-to-left strategy.

Overshooting and correction

Several instances of overshooting and correction behaviour were visible in the logs. For example if the clinician aims to enter the digit 3, then ▲ ▲ ▲ ▲ ▲ ▼ ▼ are pressed. In this case, there is an overshoot and correction of magnitude 2. The user notices that there is an overshoot by two ▲ presses and corrects the overshoot by pressing the button which performs the opposite action, in this case ▼.

This behaviour can be discerned in his style of interface because each number entry button has clear opposite buttons, for example ▲ and ▼ and ◄ and ► are opposites. Such overshoot and correction behaviour will not occur in a number entry interface that use a numeric keypad.

The logs indicate nine instances of overshoot and correction, making the probability of overshoot and correction 0.01. In 8 of the instances, the magnitude of overshoot and correction was 1 and in the other instance it was 2.

5.3 Implications from log analysis

This study was focused towards answering empirical questions raised by the analysis presented in chapter 3. This theoretical analysis is, so far, general and not specific to any domain. Using the empirical results derived from the logs it was possible to tailor the Differential Formal Analysis method to give results specifically for this medical domain.

Particular number entry strategies were used by clinicians that could be implemented in the automated analysis. A question to be considered is whether the addition of these new strategies has any effect on the design rankings described in chapter 3. Specifically, it suggests that left-to-right entry behaviour will occur on designs that start out on the right and right-to-left behaviour on designs which start on the left.

The observed left-to-right strategy from the logs might be because these logs were collected from a hospital in London and most of the clinicians using the devices used English or a European language as first language that writes from left-to-right. It is possible that logs involving clinicians using other languages which write from right-to-left, would have resulted in a higher proportion of right-to-left strategies. Therefore, in analysing the designs, both left-to-right and right-to-left strategies should be implemented on designs that start from left and right. This is an interesting issue for manufacturers to consider for the localisation of their devices since the extra keystrokes required to start number entry from the desired digit may lead to errors.

Another behavior to include in strategies is overshooting and correction behaviour. Although the overshooting and corrections that are detected in this analysis lead to the intended input values by the user, it is possible to have keystroke errors within that overshoot and recovery sequence. This should be modelled and this may change analysis results about the best 5-key number entry designs. The magnitude of the overshoot and correction should be based on the findings in this study.

5.4 Stochastic key slip simulation method

The Differential Formal Analysis method ranks number entry designs based on out by 10 error rates. The core of the method is the Stochastic Key Slip Simulation procedure (SKSS) which needs tailoring to the findings from the logs.

The SKSS method works by randomly generating N pairs (a, b), where N is the number of tasks. A *solver* uses a *strategy* to generate a key sequence to go from a to b. Then for each key in that key sequence, with a probability p per keystroke a key slip is injected that is either repetition, where a key is repeated, substitution, where another key is entered erroneously, transposition, where the sequence of two keys is transposed, or deletion where pressing a key is omitted. After inserting the key slips using probability p, the resulting key sequence is used to simulate keying in that sequence on a design which starts from a display showing the number a. The resultant value is then compared to the original intended value b to find out whether the error is out by 10 or not.

To base the simulations on the empirical findings, instead of generating random number pairs, numbers are generated that follow the same digit distributions found in the log analysis. The new strategies generate key sequences from a to b with left-to-right start on right start interfaces and right-to left start on left start interfaces. The probability of starting from the left was set to 0.625.

Overshooting and correction behaviour is also implemented. Before injecting key slips to a key sequence, overshoots are inserted along with corrections with a probability o per keystroke. The overshoot and correction in this case had a magnitude $m = 1$ 89% of the time and $m = 2$ 11% of the time.

5.5 Stochastic key slip simulation results

The previous implementations of Differential Formal Analysis were extended to reflect findings from the medical device logs. New solvers were implemented to reflect the new left-to-right strategies for right start designs and right-to-left strategies for left start designs. These solvers were tested with 10,000 number pairs to make sure that they work correctly and do not cause errors themselves.

Overshoot and correction behavior as observed in the keystroke logs was also implemented. This was implemented by performing an overshoot and correction behavior with probability p_o per keystroke. For each keystroke upon which overshoot and recovery was performed, this was achieved with a magnitude m by inserting that keystroke m times, then correcting by inserting m keystrokes which perform the opposite action. For example, consider a key sequence ◄ ▲. Overshooting and correction on the second keystroke with $m = 2$, the key sequence would be ◄ ▲ ▲ ▲ ▼ ▼.

After implementing the solvers with this overshoot and correction behaviour, they were tested with 10,000 tasks, a number of observations could be made. Designs that do not block errors and do not allow wraparound are the designs that cause errors on this type of overshoot and correction. All designs that block errors result in 0 errors from these trials, as well as designs that enable digit and/or arithmetic wraparound and cursor wraparound.

Consider a design with no cursor wraparound and no block errors with a display showing 0.00. A plausible key sequence for the value of 0.05 is ► ► ▲^{5}. If overshooting on the second ► and correcting by $m = 2$ the key sequence would become ►^{4} ◄^{2} ▲^{5}. Since the display does not wraparound and does not block, the two ► key presses in the overshooting do not change the display, but the correction sequence of two ◄ key presses move the cursor to the left. This means that the rest of the key sequence would be changing the wrong digit and in this case, the result from the key sequence with the inserted overshoot and correction would be a display showing 5.00, that is a value 100 times bigger than intended. If the same design had the block errors feature enabled, this large error would not have occurred because the interface would block on the first extra ► key press.

5.5.1 Probability of keystroke error

While the probability of overshooting and correction and the magnitude probabilities of the overshoot can be discerned from the device logs one variable that is still unknown is the probability per keystroke error. In order to decide whether this matters consider a graph of probability per keystroke error against percentage of

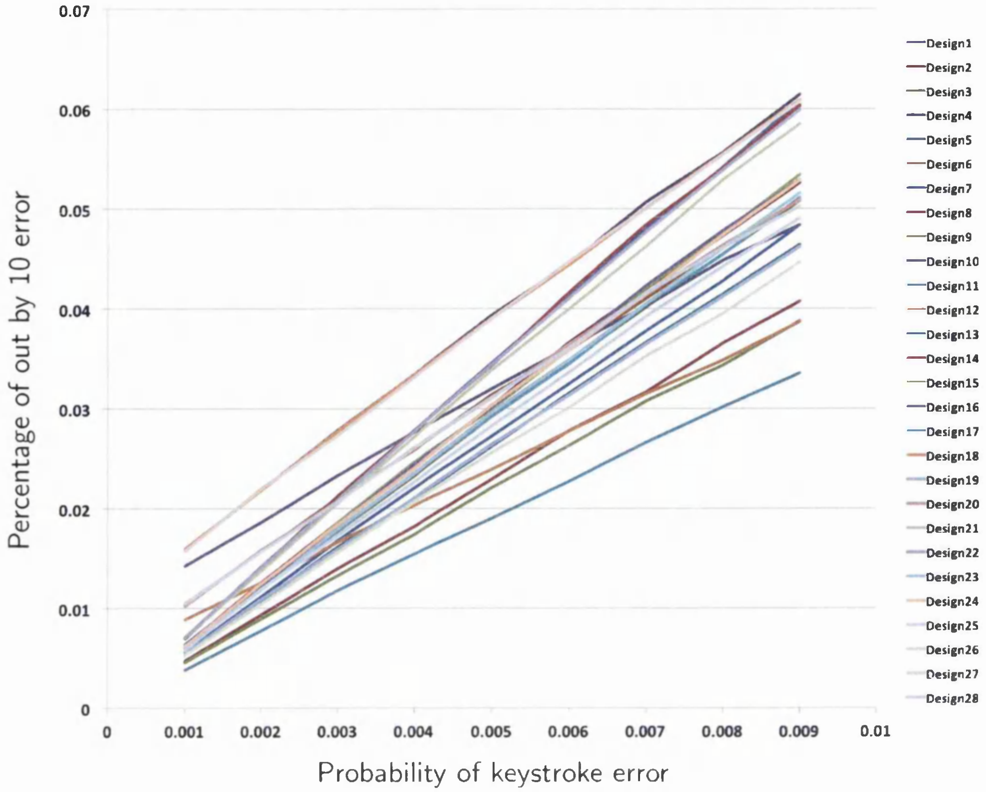


Figure 5.4: This graph plots probability per keystroke error against percentage of out by 10 errors. The lower the gradient of the line, the better the design. One can see from the graph that the best design is clearly best no matter what the probability per keystroke error is.

out by 10 errors. The results from figure 5.4 which were generated using 1,000,000 medical numbers, the probability of overshooting and correcting was set to 0.01 and for each overshoot and correction, the probability that $m = 1$ was set to 0.89 and that $m = 2$ was set at 0.11. The probability of choosing a strategy that starts from left-to-right for setting the number was set to 0.625 and from right-to-left at 0.375.

In figure 5.4, the design with the lower gradient is the best design since it is the one that caused least out by 10 errors. The best design is clearly best for any probability per keystroke error. This means that it is not necessary to decide empirically what p is.

5.5.2 Feature analysis

From figure 5.4, the feature combination of the design that came out the best has digit wraparound, no arithmetic, no cursor wraparound, starts on the right and blocks errors.

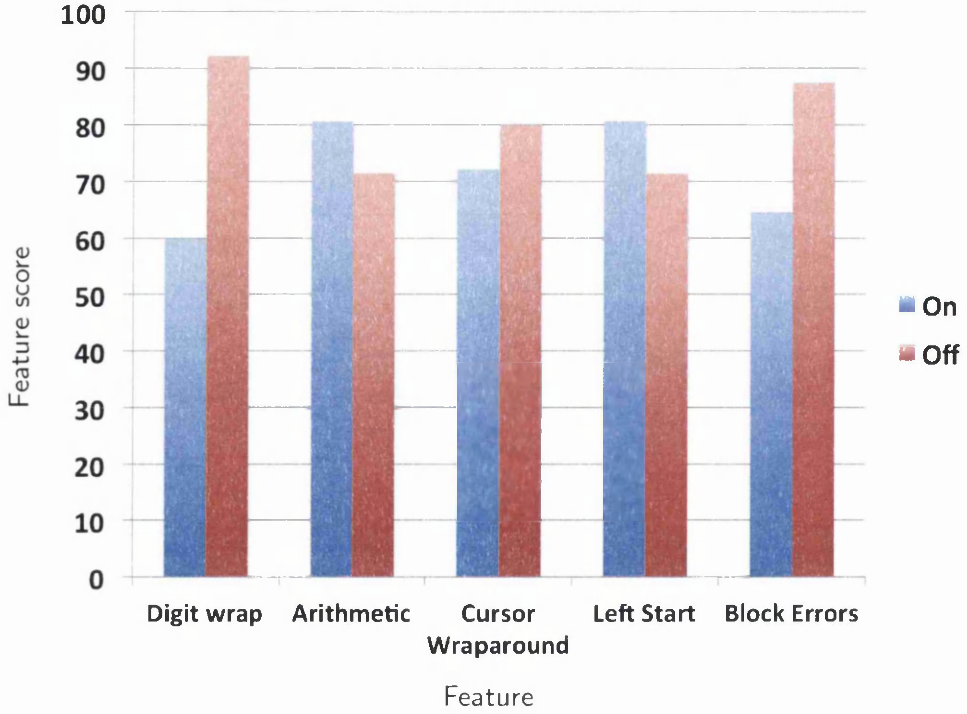


Figure 5.5: This bar chart shows whether features are better on or off. We used the gradients of the trend lines of each design from the probability per keystroke error analysis to score each feature. The features are scored by summing these gradients in designs which had the feature on and off respectively. A lower score means that the feature is better.

Figure 5.5 indicates how each feature by itself contributes to out by 10 errors. The feature name is given on the x-axis and the y-axis provides two scores for each feature, one for when it is enabled and on for when it is not. The score of the feature was obtained by summing the gradient of the trend lines of the graphs in figure 5.4 where a design had the feature on and off respectively. The lower the score, the better the feature. Designs with both digit and cursor wraparound do better than other designs. This is consistent with the earlier prediction about errors generated from the overshoot and correction behaviour. Blocking errors is an important feature to have and no arithmetic is better than arithmetic and right start is better than left start.

5.5.3 Error magnitude frequencies

To compare the theoretical model with the logs, SKSS trials were run for the design with the same features as the BBraun Infusomat Space pump. Between the theoretical model and the log analysis, one point of comparison is the frequency of error magnitudes. 1,000,000 number entry tasks were executed with medical numbers with the SKSS settings described previously. Rather than only looking at out by 10 errors, all the error magnitudes were considered to compare them to the error magnitudes graph for the noticed errors in the device logs in figure 5.2.

The error magnitude frequencies from the simulations were compared to those from the log analysis magnitude frequencies. The shape of the bar graph was not quite the same. A point that may have caused the discrepancy between what the model generated and what was found in the logs is that the probabilities of each error type occurring are unknown. From the logs it was not possible to find this out. Figure 5.6 shows bar charts for error magnitude against the percentage frequency of that error magnitude for all error types equally likely to occur, and each error type individually. Although predictions show that smaller errors are much higher in proportion to those observed in the logs, the bar graphs for repetition errors are very similar in shape to the one seen in the log analysis with higher out by 10 and out by 5 error rates.

This might indicate that repetition errors are far more frequent on 5-key number entry systems than any other type of error. Substitution errors for example, do seem less likely on 5-key systems because of the small number of buttons. However, it might mean that repetition errors are the ones that are *noticed*, and can be detected. It is possible that it is less likely for practitioners to notice substitution, transposition and deletion errors than they are to notice repetition errors.

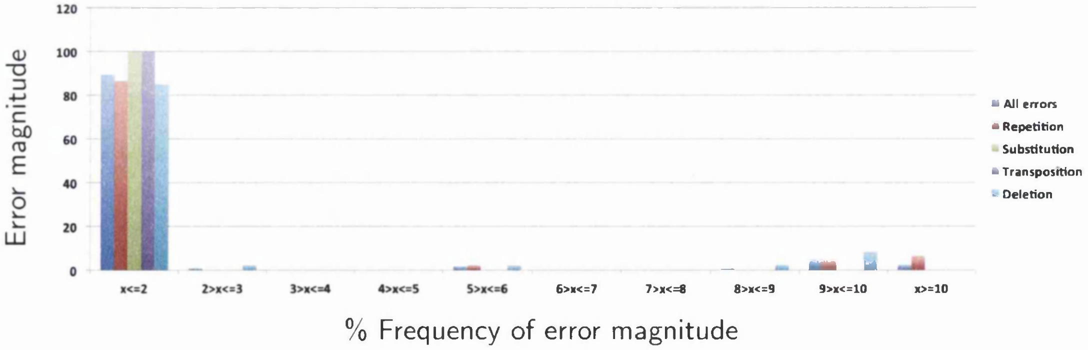


Figure 5.6: A bar graph showing the frequencies of error magnitudes when running simulations on with all possible error types, and each error type individually.

5.6 Discussion

This chapter focuses specifically on retrieving information from real clinical logs to help inform theoretical methods. Here the implications of our findings and their relation to the analysis method are discussed. Other possibilities for using the rich source of data that can be collected in logs.

5.6.1 Implications for theoretical methods

Our analysis shows that some empirical questions that were raised in chapter 3 can be answered in the context of a particular domain by analysing medical device logs. Retrieving use logs provides a large volume of situated use data that can be collected in a way that is not intrusive. In particular, the data indicate that the numbers used in infusion pumps are not the random numbers used for the study in chapter 3. To apply the Differential Formal Analysis technique to the medical domain, it would be interesting to run the study using the numbers obtained from the devices rather than random numbers with arbitrary distributions.

The number entry strategies used in this chapter can also be used to inform analytical methods. Implementing the strategies found in the logs in Differential Formal Analysis has made the analysis closer to reality and it would be interesting to compare results from previous trials to the empirically informed results.

In the case of strategies, the overshooting and correction behaviour is not apparent in the presented Differential Formal Analysis and since errors can occur within that behaviour it is worth implementing and checking whether the design rankings change.

5.7 Conclusions

In this chapter a case was made for empirically informing the Differential Formal Analysis process to compare results for 5-key number entry systems to the results previously presented in chapter 4.

One of the challenges of the work presented in this chapter was that the available keystroke logs were only of the last 200 keystrokes. Although number entry strategies were derivable from the limited data, for different and equally useful usability studies, 200 keystrokes is very little. Indeed depending on the frequency of usage of the pump, the 200 keystrokes logs covered a duration ranging from 95 days to just over four hours. For the logs we analysed in this study, after removing the outlier log that spanned 95 days, the keystroke logs covered a mean duration of 100 hours, with a standard deviation of 22 hours.

We suggest that all device keystroke logs are kept until the devices are taken for maintenance, typically approximately every six months. Based on an approximation derived from the logs analysed, this feature implies that the device keeps a record of the last 8,600 keystrokes. This would provide a rich set of data that could be processed and analysed periodically. Timing on all medical devices should be also be synchronised to allow accurate cross device log analyses. In addition, keystroke logs should contain detailed information that would make it possible to distinguish between different modes of interaction with device user interface widgets. For instance it should be clear if the user performed a click action (i.e., a press quickly followed by a release) or a press and hold action. All these would allow for a more accurate and deterministic playback of user interactions on devices.

Although the focus here is on finding out very specific empirical questions, the log data can be used to obtain other interesting insights into how these devices are used in hospitals.

Important further work that manufacturers of medical devices should carry out is on improving the use logs on these devices and use them to perform empirical and analytical analyses to improve the safety of their device designs.

In this thesis, so far the Differential Formal Analysis process has been introduced and demonstrated through a case study on 5-key number entry. The results for 5-key number entry were tailored for the medical domain by empirically informing the Differential Formal Analysis process with medical data. We have seen that a seemingly simple user interface can be implemented in numerous different ways and by going through the Differential Formal Analysis process, healthy discussions about features, strategies, and other important issues are raised. This is useful and beneficial in safety critical domains, and Differential Formal Analysis encourages it.

In the next chapter, a critical analysis of Differential Formal Analysis is made. Chapter 7.1 discusses future work that is useful to carry out and chapter 7 makes concluding remarks.

Chapter 6

Triangulating Stochastic Key Slip Simulation and Empirical Techniques

Dependable interactive applications require different methodologies than conventional usability approaches. For example, a standard laboratory experiment may find that users prefer one system to another, or that they make fewer errors or are faster. This is certainly useful information, but (except for very simple systems) a lab study cannot cover all features (let alone all states and transitions) of a system. If the interaction design has bugs — actual software bugs or poor boundary cases in the user interface — then human participant-based evaluation may not help enough. For complex systems, and for critical applications, reliance on user testing alone may not be good enough to assure a system has as few design defects as possible.

Good practice in user interface design is *iterative design*, where a system is designed, evaluated and then redesigned or otherwise improved, and the process is repeated. (Iterative design is enshrined in ISO standard 9241.) Thus iterative design acknowledges that we do not know how best to design (if we did, we would not need to evaluate designs and iterate them). Iteration also allows users to change their minds or express new insights into their requirements as they experience working prototypes. Equally, then, we ought to acknowledge we do not really know how to best evaluate a design either; why don't we also do "iterative evaluation," using and improving different techniques for evaluation — a sort-of meta iterative design? One way to do this would be to use radically different techniques, and then explore their different results in detail. Discrepancies require exploration, and either indicate inappropriate or inapplicable aspects of the evaluation methodologies or interesting aspects of the user interface design for further exploration. The point is in a safety critical application, one really wants to discover blindspots in the design process. Radically different evaluation methods are likely to complement each other in their ability to contribute to improving dependability.

In dependable design we want to find designs that reduce the probability of design defects affecting users or the success of the tasks they perform. In particular in safety critical design, we want to reduce error and the consequences of error, and to ensure the probabilities of design defects inducing problems to be as low as possible. Unlike conventional HCI, we are not primarily interested in understanding the user and their likes and dislikes, we want to find risky behaviours that may happen and make them less likely to happen (or happen and have unwanted consequences).

We therefore want to select or invent engineering features that reduce bad outcomes when a user interface is used. Unfortunately as we improve the safety of a design, the probability that users do unsafe actions reduces, and the time it takes to do statistically valid experiments with users therefore increases. We cannot rely on a user evaluation that finds zero defects, as we do not know whether this means there are no defects or that the users in the experiment failed to find any. Logically, testing never shows the absence of problems.

One solution is to stress users so that their error rate is increased. This is problematic for two reasons: is there increased error rate representative of what users would really do without the artificial stress, and once we have stressed them to some rate and eliminated user interface problems, what do we do as we try to further improve the design? We can never stress users more and more indefinitely. Rather than rely entirely on increasingly expensive experiments with human users, however we do it, there is a point at which it becomes worthwhile and insightful to emphasise alternative evaluation methods that do not rely on human users. In this chapter, statistical analytic techniques are used. The human experiments give us an idea how users behave, and then we simulate human behaviour — along with human errors — using fast computer programs. It is then trivial to perform experiments that in a few minutes simulate impractically long conventional experiments. There is an interesting balance between the psychological validity and the scale and ease with which such experiments can be done.

Another way of looking at this triangulation process is to imagine a conventional usability experiment, say, with N users, maybe 10, as participants. Typically, one user turns out to be an outlier, and they are examined closely and then discarded as unrepresentative. In the approach presented in this thesis, one of the users is a "robot" and of course is unrepresentative, but we know exactly why they are unrepresentative, so we can think through in detail whether for each feature we are interested in whether the robot performance is a good or poor indicator of required design improvements. As explained below, there is a very interesting feature of robots: the probability that they perform certain actions can be precisely controlled. We may then find that the best design changes are the same regardless (within broad limits) of those probabilities. In other words, sometimes we have parameterised the robot to be realistic (we got an initial value of some probability from real human performance) but it turns out the exact values do not matter.

This chapter argues that there is a significant variation in the safety of common number entry user interface design choices, at least under the assumptions used, which are based on real clinical data from infusion pumps.

The results show that safety should be assessed by a combination of empirical and analytic methods, and it thus follows that better user interfaces can be chosen on the basis of the assessments. This result is of use to manufacturers (who wish to design safer systems), to procurers (who wish to buy safer systems), and to patients (who wish to have safer treatment). In the long run, one could imagine that rigorous assessments of safety would be displayed prominently on devices (Thimbleby, 2013a), thus increasing awareness and hence would encourage more appropriate choices of designs for the uses to which they may be put.

Of course, safety is a complex design trade-off, and higher or lower safety as it is measured in this current work, has to be balanced against other criteria, including details of the user's task (e.g., entering a new drug dose rate is a different task than adjusting an existing drug rate). Making at least some safety assessments available, even if not the whole story, is an important step forward.

The present work uses the combination of an analytic technique based on Stochastic Key Slip Simulation (SKSS) that is described in chapter 3, with a lab study presented in (Oladimeji et al., 2013). This analytic technique allows us to analyse number entry interfaces to a detailed level of abstraction and the setup of a lab study allows us to log all the keystrokes we need and know when a number is entered erroneously.

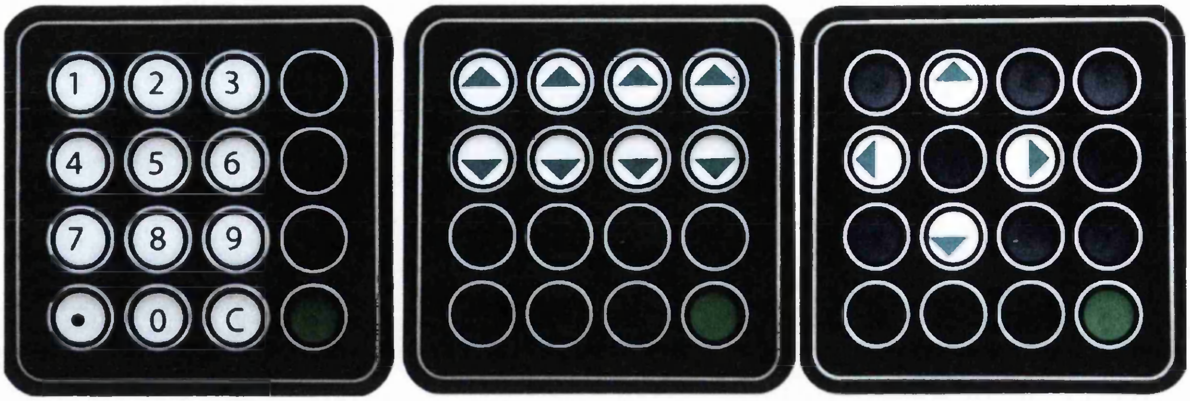
The implementation of the SKSS method has been extended to work on different types of number entry interfaces. This extension gives insight into the types of systems that can be modelled using this analytic method.

6.1 Previous work

KLM and GOMS, (Card, Newell, & Moran, 2000) are well-established analytic methods that are useful for obtaining a measure of time to perform a specified goal. These techniques generally assume no use errors and evaluate unit tasks (CogTool is a tool that partly automates this process). In reality, a user may make errors while entering a number and this requires careful analysis. The focus of our in the present work is error rate rather than time: for many applications, making a UI safer, and finding out how to make them safer, is more important than making them faster.

In safety critical systems, the safest design is not necessarily the fastest or most appealing to users. In safety critical domains, having a design that reduces errors is desirable. However, there are design trade-offs and an appropriate balance between speed and safety is required. Often making a device safer will make it slower (car brakes are a good example).

So far, in this thesis, an analytical method to study five-key number entry systems has been developed. This gave insights into how a single UI can be implemented in several different ways, some of which are more likely to cause serious harm than others.



(a) Number pad (b) Up-down (c) Five-key

Figure 6.1: The different keyboard configurations of UIs used in this study. All keypad layouts were constructed using the same physical former. This figure does not show the the device itself nor its conventional numeric display.

So far, the analytic analyses on number entry interfaces has been carried out on five-key interfaces in this thesis and on the number pad in by Thimbleby and Cairns (2010). In this chapter, the analytic method is extended to evaluate and compare the number entry systems studied by Oladimeji et al. (2013).

The results from the log analysis from chapter 5 are used in the analytical method to make the results specific to the domain of medical devices.

In safety critical medical device design, it is critical to triangulate number entry error from both theoretical and empirical perspectives. In doing this, insight into better use of the different methods and their role in the manufacturing process is presented in order to reduce unnecessary harm and death from bad interaction design.

6.2 Types of number entry UIs

The study presented in this chapter focuses on three different number entry styles shown in figure 6.1, that are discussed in the following subsections.

Since this study is based on empirical work on actual infusion pump use in a hospital, it remains beyond the scope of the present work to analyse other styles of UIs, such as those that rely on selecting from a menu of numeric values from lists, though it should be noted that some medical devices give the user a menu of standard values in this way.

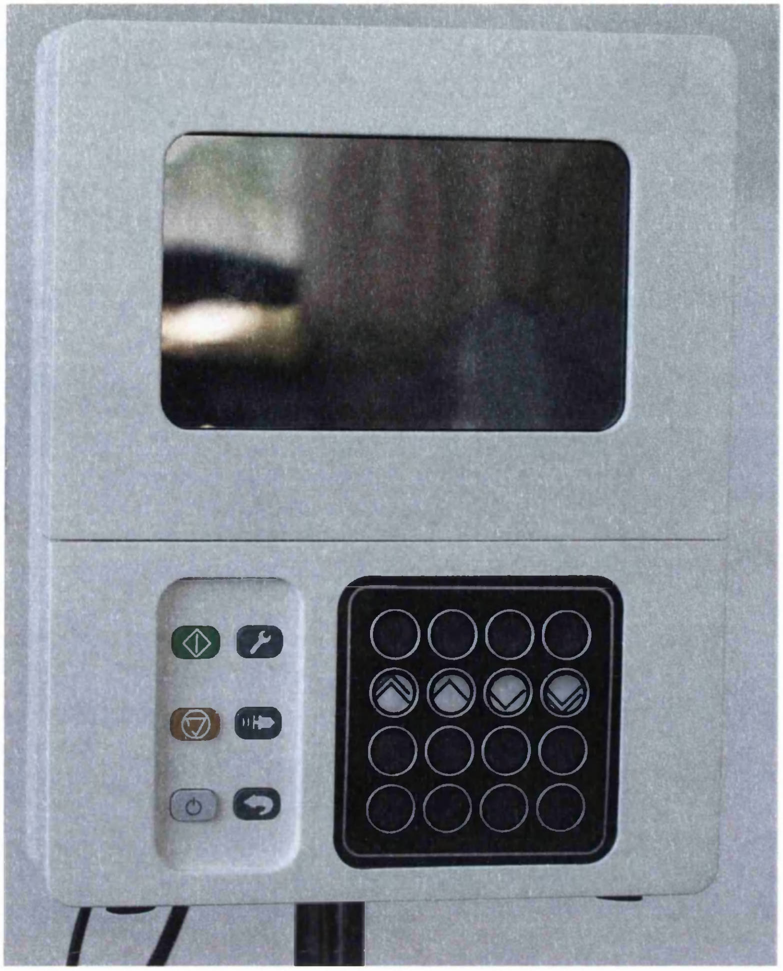


Figure 6.2: The physical prototype used in the lab study.

6.2.1 Number pad

The number pad UI allows number entry using a 12-key numeric keypad in telephone-style layout (see figure 1a). It has a decimal point and a cancel key. The decimal point key appends at most one decimal point to the number on the display. The cancel key deletes the rightmost character on the display. Inevitably, if the user keys more than one decimal point, the cancel key's behaviour is defective, as it will delete more keystrokes than the user expects — this behaviour is typical of many number entry UIs with a cancel key.

6.2.2 Up-down

The up-down UI has eight buttons arranged in two rows and four columns. The top row buttons increase individual digits in the number, and the bottom row buttons reduce the the corresponding digits. In this UI, the rightmost column matches the hundredth place value and it is used to increase or decrease the value by 0.01. This UI uses the arithmetic configuration, described in chapter 4, which means the effect of decreasing a digit from 0 or increasing a digit from 9 is carried over to adjacent digits: for example, if the display is 20.56, decreasing the 0 would change the display to 19.56, thus changing two digits in this case.

6.2.3 Five key

The five key UI has four buttons arranged in a navigation style (up, down, left, right) and a button to enter the number. The left and right buttons move a cursor on the screen that selects a place value in the number, and the up and down buttons increase or decrease the selected digit. Like the up-down interface, this UI works in the arithmetic configuration.

6.2.4 Chevron keys, knobs and other styles of UI

Chevron-key interfaces (typically with upward facing chevron buttons and downward facing chevron buttons in a row) are also commonly used in medical devices. In contrast to the three styles we have evaluated, chevrons use a continuous interaction style: values change while the button is held down, and often a button can be held down longer to change the value faster. This type of UI is better suited to dynamic closed-loop feedback analysis (Niezen, 2013) and is not evaluated in the present work.

6.3 Experiments

In this chapter, the SKSS implementation from chapter 3 is extended to analyse more user interfaces that have been previously analysed empirically by Oladimeji et al. (2013). This section briefly describes the lab study from the previous work so that the differences between running empirical and analytic experiments can be highlighted. A description of how the previous SKSS implementation was extended to evaluate three of the user interfaces evaluated in the lab study is presented in

Error	Out by 2	Out by 10	Out by 100
All errors	83.7	16.3	0
Deletion	0	0	0
Repetition	100	0	0
Substitution	82	18	0
Transposition	0	0	0

Table 6.1: Result of percentage error rates for the up-down interface from the analytic study.

this chapter.

6.3.1 Calibration on real data

Log files were obtained from 60 syringe pumps in clinical use from the university hospital in Swansea. The log files were anonymous and contained no personal information. 30 numbers used as rate and volume settings were randomly sampled from the logs to inform the analytic analysis.All the numbers had a decimal part and ranged in value 0.26–83.3. A third of the selected numbers used had a precision of 2 decimal places. The same 30 numbers were used for both the lab study and the analytic trials.

6.3.2 Lab study

In the lab study by Oladimeji et al. (2013), 33 participants (22 female) took part in the experiment. All participants experienced all the interfaces and the same numbers were entered on all the interfaces. Participants were trained on each UI before commencing trials on the interface. Ten numbers were used in a practice session and 20 were used in the experiment. The order in which the UIs and the numbers were encountered were randomised for all participants.

6.3.3 Analytic study

For the analytic study, the implementation of the Stochastic Key Slip Simulation method introduced in chapter 3 is extended to accept a state machine described in JavaScript Object Notation (JSON). The model discovery method, developed by

Gimblett and Thimbleby (2010), was ran on the number entry interfaces used in the lab study to generate a JSON state machine of each number entry interface. The SKSS implementation used these JSON models to ensure that the number entry interface models used in both the lab and analytic study were identical.

For each number entry state machine, each vertex represented every possible display of the number entry system and the arcs represented the transitions that happened after each button press. Therefore if, for example, in a keypad interface the display is currently showing 0, we are currently at the state named '0' and if 1 is pressed, the outgoing '1' arc from the '0' state takes us to the state named '1', therefore displaying 1.

In both the lab and analytic study, the number entry interfaces displayed numbers from 0 to 100 with accuracy of two decimal places. Representing number entry systems state machines in this way leads to large state machines. Even though each number entry interface had the same range and the same accuracy, the state machines are of different sizes: The number pad state machine has 11,204 states, the up-down machine has 10,001 states and the five-key machine has 50,005 states.

The five-key state machine is exactly five times the size of the up-down state machine. Both the five-key and up-down interfaces work by selecting which digit to edit and editing that particular digit using the up and down buttons. The difference between the two is that in the up-down interface, the user selects the digit to edit by selecting the correct button to press, while in the five-key interface, the user selects the digit using the left and right buttons that will highlight the selected digit. The key difference is that the five-key interface gives the user a visual cue of which digit is selected while the up-down interface relies on the user looking at the buttons. The state machine of a five-key interface will always be n times bigger than the state machine of an up-down interface where n is the number of digits displayed.

The state machine for the number pad is a different size than the up-down and the five-key. This is because the number-pad interface is a form of sequential number entry where the decimal point has to be input by the user, leaving the possibility of states with naked decimal points. This is not possible in up-down and five-key since both interfaces show a constant display width with unchanged digits at '0'.

The SKSS method works by randomly generating N pairs (a, b), where N is the number of tasks. A *solver* uses a *strategy* to generate a key sequence to go from a to b. Then for each key in that key sequence, with a probability p per keystroke a key slip is injected which is either (1) repetition, where a key is repeated, (2) substitution, where another key is entered erroneously, (3) transposition, where the sequence of two keys is transposed, or (4) deletion where pressing a key is omitted. After inserting the key slips using probability p, the resulting key sequence is used to simulate keying in that sequence on a design that starts from a display showing the number a. The resultant value is then compared to the original intended value b to find out whether the error is out by 10 or not.

For for this analytic study, SKSS was run with 10^5 number pairs for each UI and five times for each interface: in the first trial, all types of keying error (repetition, deletion, substitution and transposition) were equally likely to happen; and in the rest of the trials the key slips were analysed individually to see how the type of key slip error contributed to the results. For this study, the solver found the actual shortest path in the JSON model between the input number and the intended number.

The keystroke error probability for SKSS was set to 0.01. In chapters 4 and 5 of this thesis it is shown that the real-world empirical value of keystroke error probability does not matter, since a range of keystroke error probabilities were tried and the rankings for different designs did not change with the different probabilities.

The key sequences used to set the values on each interface are the shortest possible key sequence to input that value in each individual interface.

6.4 Results

Figure 6.3 shows a paired bar chart of the percentage of errors that occurred for each of the number entry interface styles, comparing the lab study results from (Oladimeji et al., 2013) and the analytic study from this present work.

In the lab study, the up-down interface had mostly out by 2 errors, less out by 10 errors and no out by 100 errors. These proportions were the same in the analytic

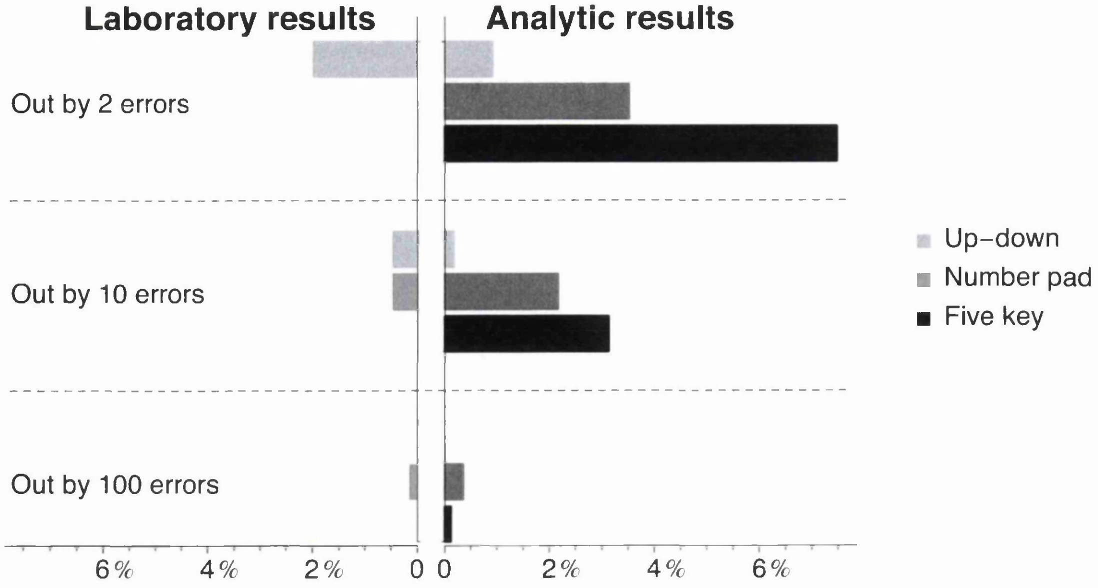


Figure 6.3: Comparing laboratory and analytic results, showing out by 2, out by 10 and out by 100 errors for each style of interface. The analytic results here are for the trial with each error occurring with equal probability.

Error	Out by 2	Out by 10	Out by 100
All errors	58.2	35.8	6
Deletion	56.1	33.9	10
Repetition	49.8	50.2	0
Substitution	86.8	9.1	4.1
Transposition	48.4	51.6	0

Table 6.2: Result of the percentage error rates for the number pad interface from the analytic study.

study but the percentage of errors in the analytic study were smaller for this type of interface. The up-down interface is the only interface that does not cause out by 100 errors in the analytic study. The reason for this becomes clear when we look at results from the other trials running each error category separately in the next sub-section.

The number pad does not have any out by 2 errors in the lab study but has few out by 10 and out by 100 errors. In the lab study, the number pad turned out to be the interface that causes high magnitude errors, the type that would be dangerous in the medical domain. In the analytic study, the number pad had errors of all magnitudes and from this study it is also the interface with the highest out

Error	Out by 2	Out by 10	Out by 100
All errors	69.7	29.2	1.1
Deletion	48.1	49.9	2.1
Repetition	100	0	0
Substitution	69.5	29.6	1
Transposition	99	1	0

Table 6.3: Result of the percentage errors for the five-key interface from the analytic study.

by 100 errors.

In the lab study, there were no errors at all for the five-key interface. In the analytic method, key-slip errors are injected with equal probability across all interface styles. With the same error injection probability rates across designs, five-key interfaces have more errors than observed in the lab study. This is expected since errors were deliberately injected in the five-key trial.

The percentage error rates of the two types of studies are different. In the lab trial, there were no errors on the five-key interface. If the lab trial was run for a longer period of time, there would have been errors. The lab study shows that the error rates for the five-key interfaces are very low. Analytic methods can run trials that would take an unreasonably long time to do using lab trials, however, further work needs to be done to better inform the analytic methods with empirical results to simulate human behaviour more closely.

In the analytic results, each interface had highest out by 2 errors, followed by lower out by 10 errors, and out by 100 errors were the lowest for each interface. Although the five-key interface has higher out by 2 and out by 10 errors than the number pad, the number pad has higher out by 100 errors. In the lab study, the up-down interface had highest out by 2 errors, lower out by 10 errors and lowest out by 100 errors but the number pad had lowest out by 2 errors (none), highest out by 10 errors and low out by 100 errors. The lab results indicate that the number pad is more likely to generate higher magnitude errors and the out by 100 errors observed in the analytic study confirm that number-pads should be avoided in safety critical number entry systems.

6.4.1 Error analysis

This section now focusus on the analytic study to look at how each error type contributed to the results. This sort of analysis is difficult to do from a lab based study because error rates are very low.

Tables 6.1, 6.2 and 6.3 show percentage of errors that occurred for five-key, number pad and up-down interfaces when running five individual trials with 10^5 numbers. These tables show how each error type (deletion, repetition, substitution and transposition) affect the result of all error types combined.

Figures 6.4, 6.5 and 6.6 show the ratios of out by 2, out by 10 and out by 100 errors respectively between all interfaces for the five different trials.

The up-down interface does not have any out by 2, out by 10 or out by 100 errors caused by deletion and transposition errors. Since the correct key sequence to enter any number in a trial is a sequence of ▲ buttons for each digit, deletion errors would be very small (less than out by 2). Transposition errors have a small probability of causing any error at all. If the hundreds digit is being incremented five times and one of those five keystrokes is transposed, this does not result in an error. A transposition error will only occur in the location in the key sequence where the two subsequent keys are different. In the case of the up-down interface this would be when one digit is set and the next digit is about to be set. Repetition errors in the up-down interface only cause errors of small magnitudes, and no out by 10 or out by 100 errors were detected. In general, this interface is less the least likely to cause high magnitude errors.

The number-pad interface has the most out by 100 error percentages for all types of errors. From the bar graph in figure 6.6 it is clear that the number pad consistently has a higher ratio of out by 100 error. The results from the analytic trial show that repetition errors do not cause any out by 100 errors in the number pad.

In five-key interfaces, repetition errors are very unlikely to cause high magnitude errors. The analytic results for repetition errors detected no out by 10 or out by 100 errors. Transposition errors in this case are also much less likely to cause errors of high magnitude with 99% of transposition errors being out by 2 errors, 1% being out by 10 and no out by 100 errors. The cause for the errors in five-key interfaces being of lower magnitude in transposition and repetition errors is similar to what happens in up-down interfaces.

The error analysis presented here gives insight into why the analytic study produced the results it did. Up-down interfaces resulted in lower error rates because there are two types of errors (deletion and transposition) that do not generate any errors at all even in a trial simulating 10^5 number entry tasks. This interface designs out these two classes of error. Up-down interfaces also show that three types of errors do not produce any out by 10 or out by 100 errors — that is, high magnitude errors. This is desirable in a safety critical domain where high magnitude number entry errors can be fatal.

6.5 Discussion

Safety critical number entry is prevalent in the medical domain. Infusion pumps that are commonly used in hospitals worldwide use a variety of the number entry systems studied in this chapter. The results from the previous lab study, that this current chapter makes comparisons to (Oladimeji et al., 2013), suggest that number pads should not be used in hospitals. The results presented in this present chapter make an analytic argument to further substantiate the claims made using the lab study.

Figure 6.6 shows that in the analytic trial, the biggest percentage of out by 100 errors are from the number pad interface. This is true for all five trials carried out. Errors of this magnitude would be fatal for most drugs administered, thus this style of interface should be avoided in medical device design.

Figure 6.4 is another graph drawn from the results in the analytic trial. Five-key interfaces are sensitive to substitution errors. This sensitivity can be reduced if the buttons on the device are designed to be far apart from each other, such as requiring two hands to input the number. The current implementation of SKSS does not take into consideration how far the buttons are on a device.

If the probability of the user making an error in an empirical experiment lasting t hours (including preparation, participant briefing, etc) is p, then the expected number of hours that experiments need to be performed to log at least one error is

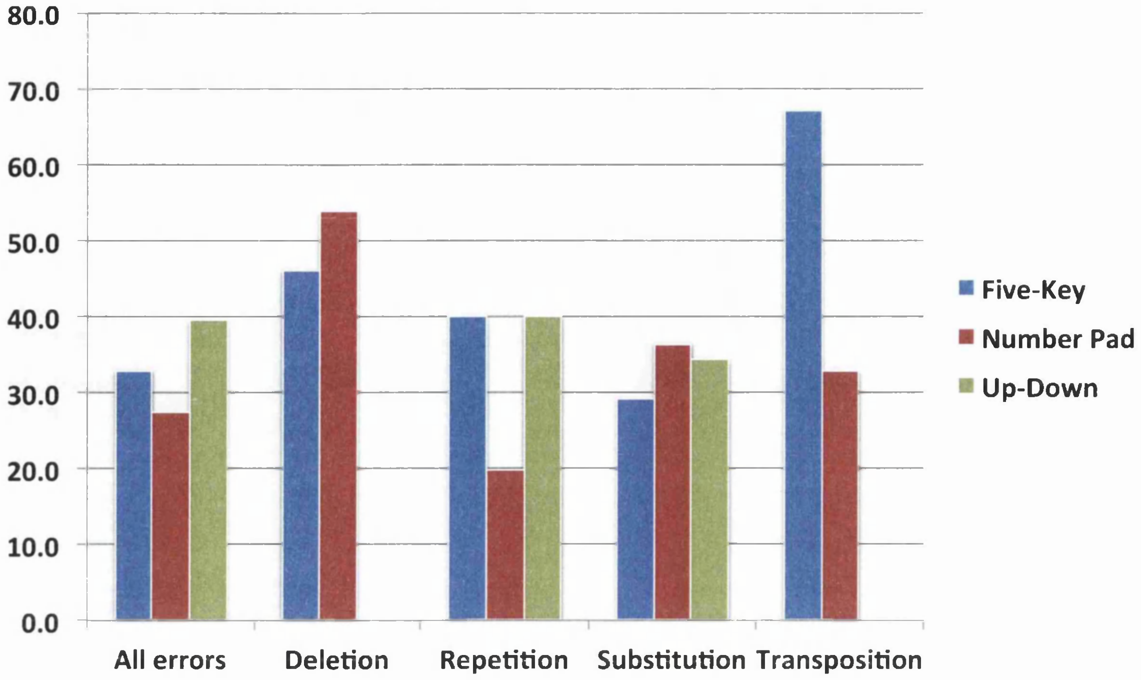


Figure 6.4: Out by 2 error rates for each design in the five separate trials. All errors refers to the trial that was run with the four error types occurring with equal probability while the rest of the trials were run with only one error type being injected.

roughly t/p (t and p may not be constant, etc). For a typical $t = 1$, $p = 0.01$, this time exceeds the working week, and practically one starts to have to run experiments over months or even years to get enough data to say statistically useful things. Seen like this, empirical experiments have limitations; in contrast, analytic methods can simulate a user by computer program and therefore be performed very fast and generate quantities of data for analysis. Interestingly, in experiments looking for design defects that may affect safety, the actual probabilities are not as interesting as designing to reduce them. We have found that our analysis methods suggest insightful design recommendations despite the uncertainties of real user performance data. Illustrating this argument, below, we will show how our analytic technique identified some design problems with a "5 key" user interface that had not been detected by empirical experiments.

In safety critical domains such as hospitals, human error eventually happens. It is our aim to reduce the harm caused by human error, when it does occur. Even a well-designed lab study may not be able to sample all the errors that will occur during use for each interface. For example, during the lab study the five-key interface

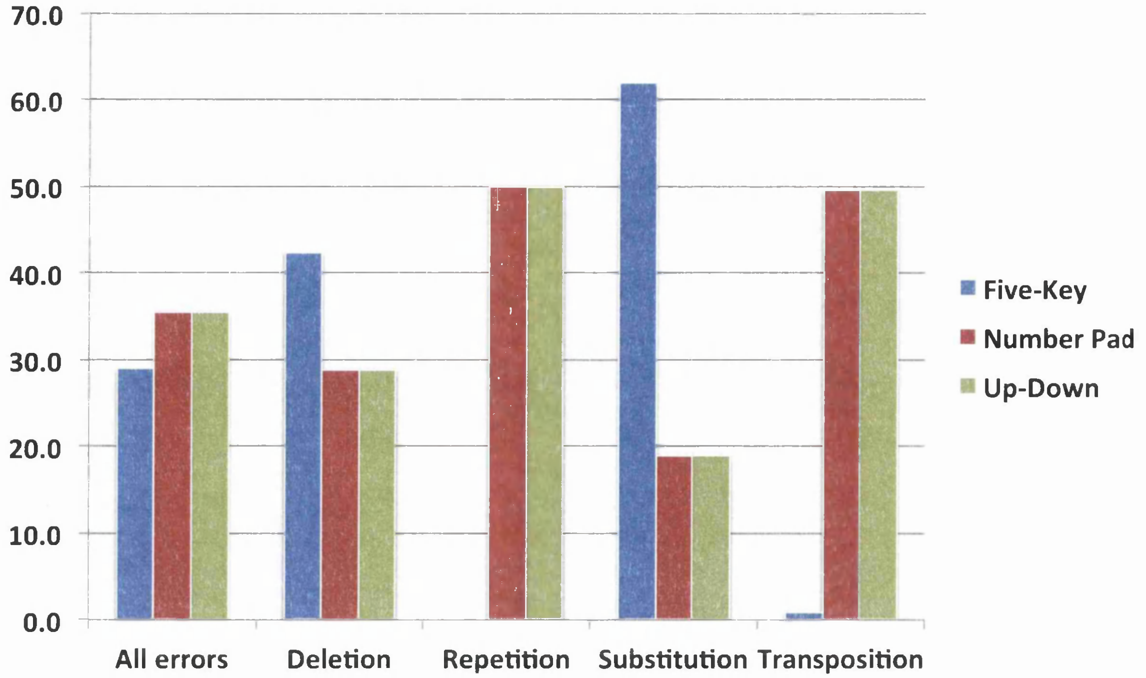


Figure 6.5: Out by 10 error rates for each design in the five separate trials. All errors refers to the trial that was run with the four error types occurring with equal probability while the rest of the trials were run with only one error type being injected.

and number pad did not show any out by 2 errors; an implausible result. In general, it is not feasible to run a lab study for a long enough time, analytic studies can be used in complement to explore the bigger picture, instead of assuming that the interface design is error free.

6.5.1 Differences in results

The version of SKSS reported in this chapter simulates key slips in a user's task with equal probability and SKSS simulates the task many more times than is feasible in a typical user study. Consequently and generally, the number of reported errors is a lot more in the analytical study than the lab study.

Analytic studies are important in the design process for engineering out error. By design, key slip errors due to repetition and transposition of keys are less severe on the up-down interface. Consequently from the SKSS analyses, most of the errors on the up-down interfaces were out by 2 errors. Empirical results complement this method. The low number of out by 10 errors on the up-down interface highlights a current limitation in the key slip injection mechanism used in the analytic study

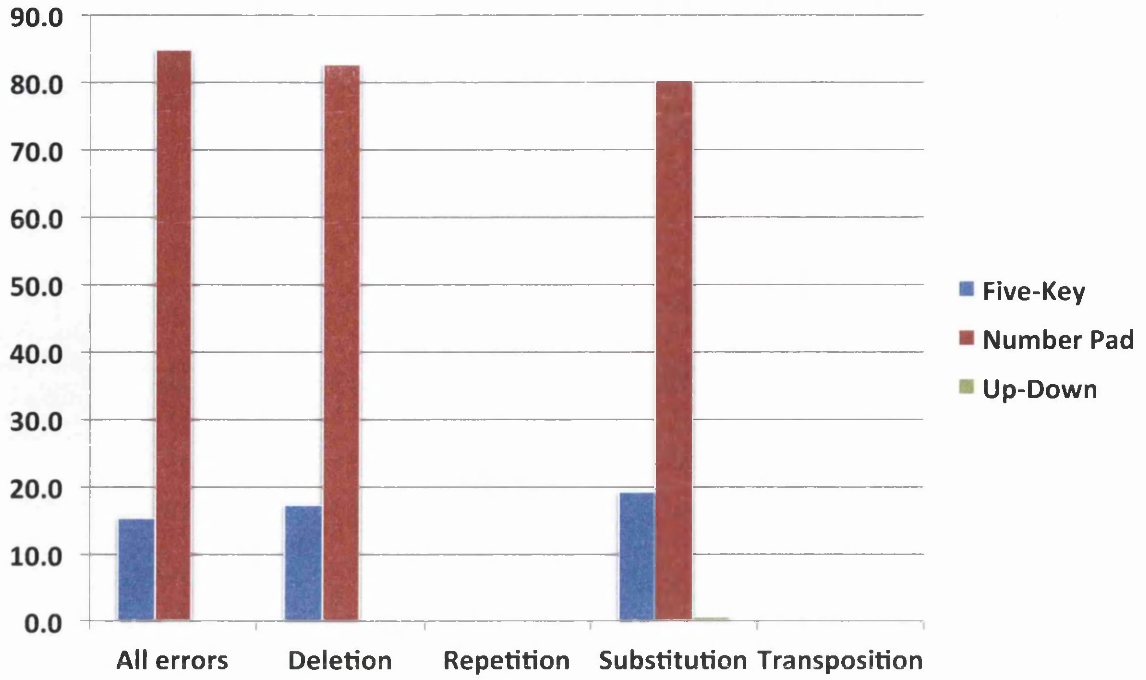


Figure 6.6: Out by 100 error rates for each design in the five separate trials. All errors refers to the trial that was run with the four error types occurring with equal probability while the rest of the trials were run with only one error type being injected.

that is currently not as sophisticated as some of the errors that were manifest in the lab study. For example, on the up-down interface, some users made out by ten errors because they shifted the place values of the digits to be entered. One user entered 1.11 instead of 11.1 (Oladimeji et al., 2013). This means that SKSS is currently only reporting a certain class of fundamental key slip errors but misses out cognitive errors that users make when using devices. This reiterates the point in this methodology about iterative evaluation; modelling cognitive-type errors in SKSS based on user behaviour from lab studies would enhance the method to cover more types of error.

In modelling both up-down and five-key number entry systems we see that conceptually, the only difference between them is that five-key uses a visual cue on the display while the up-down interface does not. From an eye tracking lab study reported by Oladimeji, Thimbleby, and Cox (2011), interfaces where users looked at the display while entering numbers led to fewer undetected errors. Currently, SKSS does not account for the differences in undetected error rates that are as a result of differences in interface design. The cursor based design of the five-key interface requires a level of user attention on the screen. This implies that users are probably more likely to notice errors on this interface than on the numeric keypad interface. An improvement on the SKSS method would vary the probability of error based on user interfaces to account for the possible differences in error detection rates, which can only be found out empirically.

6.6 Conclusions

Number entry user interface design can be improved by triangulating empirical *and* analytic methods for evaluation. Three types of number entry systems were used in this study: the number pad, up-down, and five-key interfaces. A SKSS implementation was calibrated with a large hospital data set and applied to three different styles of interfaces. The results were then compared to previous empirical results presented in (Oladimeji et al., 2013).

The coverage of an analytical method, such as SKSS as used here, can be far more exhaustive than the small sample of errors that can be studied in a user experiment. Conversely, the types of real user errors occurring in the lab are more complex than those currently modelled using analytical methods.

Such insights will help improve future versions of the analytic methods, especially with respect to typical user error types. Empirical methods help in making sure that users understand the different designs while analytical methods help in designing the fine details of the systems that are essential in critical design such as those of medical devices. The approaches are complementary: while research papers might contribute research to either approach exclusively, real design — particularly safety critical UI design — requires both.

HCI contributions lie on a spectrum from practice to research. For research, this chapter shows both analytic and empirical methods need developing, together, to more reliably understand safety critical design issues. For practical system development — where we wish to improve a particular design rather than uncover underlying design principles — the main insight is that designing number entry systems by using both types of analysis methods is essential, and while neither method is perfect, the combination of methods raises critical questions that will need to be addressed in further development.

Chapter 7

Conclusions & further work

Skilled users make slips. As far as possible, interactive systems should be designed to detect and help users manage as many slips as possible to help avoid slips turning into errors that lead to adverse situations. In the medical domain — where entering drug doses into interactive medical devices is safety critical — we wish to reduce the number of drug over and under doses. This thesis has shown that even such "simple" user interfaces have a variety of subtle design choices that can be used in combination to make a significant difference to their sensitivity to user error. In particular, this thesis recommends that user interfaces attempt to block user error (e.g., beeping or otherwise reporting detectable errors to the user, rather than ignoring them — as is common practice). This and other recommendations are based on a very diverse set of formal simulations, and thus are independent of the usual implicit design assumptions. These are significant results that can lead to practical applications in real, safety critical environments.

Differential formal analysis is a new methodology that should be used to complement user trials for more rigorous evaluation of safety critical number entry systems. Subtle interaction design choices in number entry lead to drastically different outcomes; it is crucial that these choices are explored and that a design that is more resilient to human error is implemented. More broadly, the Differential formal analysis process leads to important discussions about empirical questions about how practitioners use number entry systems in practice.

Logs from 19 medical infusion pumps that were used for three years in a UK hospital were collected and analysed to answer some of the empirical questions raised by the differential formal analysis process. An analysis was carried out on the numbers that were infused; on the noticed errors that were detectable from the logs; and on the strategies that practitioners used to input numbers. One outcome from this analysis is that medical device logs can be a rich source of data and they can be improved to make usability analyses better. Chapter 5 makes the results from the differential formal analysis process domain specific to medical device design.

A wider view on number entry design was taken in chapter 6. This chapter served to generalise the differential formal analysis process to number entry systems other than 5-key number entry systems and also to compare results to a lab study carried out on the same number entry systems. This chapter highlights the importance of using both analytical and empirical techniques in safety critical design. Empirical techniques are important to design systems that are resilient to human error. Eventually, safety critical systems are refined to reduce human error to be undetectable in user studies. It is important to have an analytic tool that is calibrated with the real world to further refine the system in order to further reduce harm caused by human error.

7.1 Further work

This thesis has shown that safety critical number entry research is important, this section outlines further work to carry on the important area of research.

7.1.1 Further empirical and analytic iterations

Differential Formal Analysis is a method that exposes empirical questions such as: how do users enter numbers? Medical device logs have been analysed to uncover possible strategies that users take when inputting numbers. The results from the log analysis have been used in the differential formal analysis process to see how the various real world strategies effect the simulated results.

In the method of differential formal analysis, there are two types of empirical questions raised.

- Those that can use the differential formal analysis process to find out whether they matter or not.

- Those that the differential formal analysis process cannot find out whether they matter or not. In this case, it is worthwhile carrying out the empirical studies to further inform the differential formal analysis process.

For the first type of study a stochastic experiment can be designed to see whether the results change depending on the empirical outcomes. A good example of this is demonstrated in this thesis in chapter 4 where the question of whether the error probability rate of entering numbers matters is answered. For the experiment in chapter 4, SKSS trials with varying probabilities were run. The results from these trials showed that the design rankings did not change depending on the error probability, therefore finding out the actual value of the error probability is unnecessary.

A question that cannot be answered through Differential Formal Analysis is whether users understand the particular features that are studied. In this thesis, reasonable features that are found in real world devices have been analysed, however, studying whether the features are understandable by users is an important issue. The arithmetic feature proves to be a good feature to have through this type of analysis, however, an empirical study should be carried out into whether users understand features, such as arithmetic. This is an important study since if a design is not understood by users, this would increase the probability of users making slips. This thesis shows how designs rank when the same error probabilities are applied, however the error probabilities might in reality be different in different designs.

7.1.2 Transfer errors

Chapter 4 shows that there are various ways of implementing five-key number entry systems. The various designs are made up of combinations of features described in the chapter: cursor starting position; digit wraparound; cursor wraparound; arithmetic; and block errors. Chapter 4 provides results of what design – made up of a combination of features – reduces the harm caused by human error.

Given that the best implementation of a 5-key interface is found, upgrading medical devices in hospitals to software versions that implement the best implementation might result in *"transfer errors."* A transfer error occurs if a clinician uses a key sequence that results in the intended value on one number entry system implementation but a different value on a system upgrade or different medical device.

SKSS can be used to find the harm caused by transfer errors between the different number entry designs. When upgrading software versions, it is critical to consider the harm cause by transfer errors and upgrade systems accordingly.

7.1.3 Improving medical device logs

Currently, medical device logs are not detailed enough to perform some types of usability analyses. More accurate logging of how clinicians interact with medical devices would give invaluable interaction design insight. Infusion pump log files from three different manufacturers show that limited keystrokes are logged, timestamps are sometimes coarsely grained, and events are difficult to match. Better logging to enable better usability studies and better incident investigations are necessary.

7.1.4 Continuous number entry systems

In chapter 6, the SKSS method was extended to analyse more number entry systems that were studied empirically by Oladimeji et al. (2013).

There are types of number entry systems – continuous number entry systems – that are not currently evaluated by SKSS. As future work, one should explore reducing the harm caused by human error on these types of number entry systems.

7.1.5 Better modelling of human error

From the comparison of the analytic analysis to the empirical analysis in chapter 6, it was found that some errors that were made in the empirical study were not modelled in the analytic study.

More iterations on the study carried out in chapter 6 should be carried out in order to improve the results.

The current SKSS implementation, models *slip errors*, however there are more types of errors that are made by people that can be modelled to be included in the analytic technique used in this thesis.

Cognitive errors are one type of errors that have been seen in the empirical trial. For example, in the Up-Down interface, if the number 10 is to be entered, a person might place their hand on a button further to the left than intended. The result would be a display showing 100, rather than 10. In the current implementation of the SKSS method this type of error *might* occur, as two substitution errors (by the same button) in succession. In the current implementation, this sort of error is less likely to happen, therefore better human error modelling would improve the analytic technique.

Another aspect of human error that is not considered in this thesis are errors caused from environmental aspects such as distractions. It is currently unknown how these environmental factors contribute to number entry error.

7.1.6 Improved physical designs and hardware modelling

This thesis analyses number entry systems found in infusion pumps that are currently on the market. With the improvement of electronics, and accessibility and lower cost of new hardware – such as touch screens – important further work is designing better hardware layouts for safety critical design.

The current implementation of SKSS does not consider the physical layout of the device. In a device that has a button layout where the ◄ and ► are far apart – possibly far enough apart for the ◄ button to be keyed with the left hand and the ► to be keyed with the right hand – it is less likely to have a substitution error where the ► key is substituted with the ◄ key.

Touchscreens bring in a wide range of possibilities for number entry design. There are various graphical number entry techniques that can be implemented on touchscreens that can be designed to reduce harm caused by human error. Touchscreens have not been evaluated in medical settings, however, there are scenarios where they might not be practical, such as in a device used on a search and rescue helicopter where the environment might cause error when interacting with a touchscreen.

7.1.7 Data entry

This thesis has mainly focussed on safety critical number entry interfaces. Data entry as a whole has not been explored.

Primarily, when entering drug doses into infusion pumps, three parameters are input: (i) the **volume** of the fluid being infused; (ii) the **rate** of infusion and (iii) the **time** (or duration) of the infusion. These three parameters are used together where $Volume = Rate \times Time$.

In some commercial infusion pumps, two of these values are required as input and the third value is calculated. In order to reduce error, one might explore a system where the three values are input and then checked automatically by the pump's software to make sure that the three values obey the formula of $Volume = Rate \times Time$. One might find errors in this way and alert the user to input the values again.

Beyond number entry, data also comes in the form of text. The research methods presented should be explored to analyse text entry. This would be useful in safety critical domains as well as for commercial products, such as mobile phone touchscreen keypads.

7.2 Conclusions

This thesis shows that the problems with number entry systems design are important to address. In safety critical scenarios such as busy hospital wards, better number entry systems design can prevent unnecessary adverse events which could be fatal. It is important to choose the hardware layout that is more resilient to human error, and further than that, it is important to choose the best interaction (program code) design for the specific hardware layout.

This thesis presents an analytical and evidence based method that can help improve medical devices. It is important for number entry systems that look the same, to behave in the same way. The method presented in this thesis should be used to find the design that is more resilient to human error and regulatory bodies should enforce a standardisation on the interaction design of number entry systems that employ the same hardware layout. This would reduce unnecessary adverse events.

The work in this thesis shows that we can begin to solve the problems in medical device design. The further work outlined in this chapter shows that this is a good start for a research program that extends this work to make medical device design more reliable.

References

Aceves, C. M., Oladimeji, P., Thimbleby, H., Lee, P., & Aceves, M. (2013). Are prescribed infusions running as intended? Quantitative analysis of log files from infusion pumps. *British Journal of Nursing*, *22*(191), 15–21.

Annett, J. (2003). Hierarchical task analysis. *Handbook of cognitive task design*, 17–35.

Arney, D., Jetley, R., & Jones, P. (2007). Formal methods based development of a PCA infusion pump reference model: Generic infusion pump (GIP) project. In *Proceedings of the High Confidence Medical Device Software and Systems (HCMDSS)* (pp. 23–33).

Azenkot, S., Bennett, C. L., & Ladner, R. E. (2013). DigiTaps: Eyes-Free Number Entry on Touchscreens with Minimal Audio Feedback. In *Proceedings of the 26th annual ACM symposium on User interface software and technology - UIST '13* (pp. 85–90). New York, New York, USA: ACM Press. doi: 10.1145/2501988.2502056

Baber, C., & Stanton, N. (2001). Analytical prototyping of personal technologies: using predictions of time and error to evaluate user interfaces. In *Interact* (Vol. 1, pp. 585–592).

Baber, C., & Stanton, N. A. (1996). Human error identification techniques applied to public technology: predictions compared with observed use. *Applied Ergonomics*, *27*(2), 119–131.

Brown, A. B., & Patterson, D. A. (2000). To Err is Human: Building a Safer Health System. In L. T. Kohn, J. M. Corrigan, & M. S. Donaldson (Eds.), (p. 312). National Academies Press.

BS EN. (2008). *EN ISO 62366:2008 Medical devices - Application of usability* *engineering to medical devices* (Vol. 3).

BS IEC. (2001). ISO 61882:2001 Hazard and operability studies (HAZOP studies) -- Application guide.

Burford, B. (1993). Designing adaptive atms. *University of Birmingham. Unpublished MSc Thesis.*

Carayon, P. (2010, September). Human factors in patient safety as an innovation. *Applied ergonomics*, *41*(5), 657–65. doi: 10.1016/j.apergo.2009.12.011

Card, S. K., Newell, A., & Moran, T. P. (2000). *The Psychology of Human-Computer Interaction.* Hillsdale, NJ, USA: L. Erlbaum Associates Inc.

Clark, G., Courtney, T., Daly, D., Deavours, D., Derisavi, S., Doyle, J. M., ... Webster, P. (2001). The Möbius Modeling Tool. In *Proceedings of the 9th international Workshop on Petri Nets and Performance Models (PNPM'01)* (pp. 241–250).

Clarke Jr., E. M., Grumberg, O., & Peled, D. A. (1999). *Model checking.* Cambridge, MA, USA: MIT Press.

Cox, A., & Cairns, P. (Eds.). (2008). *Research Methods for Human-Computer Interaction.* Cambridge University Press.

Curzon, P., Eslambolchilar, P., Gimblett, A., **Cauchi, A.**, Lee, P., Li, Y., ... Thimbleby, H. (2011). *Towards Dependable Number Entry for Medical Devices CHI-MED: Computer-Human Interaction for Medical Devices.*

Curzon, P., Rukšėnas, R., & Blandford, A. (2007, October). An approach to formal verification of human computer interaction. *Form. Asp. Comput.*, *19*(4), 513–550. doi: 10.1007/s00165-007-0035-6

Dain, S. (2002, January). Normal accidents: human error and medical equipment design. *The heart surgery forum*, *5*(3), 254–7.

Dean, B., Schachter, M., Vincent, C., & Barber, N. (2002, December). Prescribing errors in hospital inpatients: their incidence and clinical significance. *Quality & safety in health care*, *11*(4), 340–344. doi: 10.1136/qhc.11.4.340

de Moura, L., Owre, S., Ruess, H., Rushby, J., Shankar, N., Sorea, M., & Tiwari, A. (2004). SAL 2. *CAV, volume 3114 of Lecture Notes in Computer Science*, 496–500.

Eslambolchilar, P., Webster, J., & Niezen, G. (2013). The Evolution of Number Entry: A Case Study of the Telephone. *Human-Computer Interaction INTERACT 2013*, 538–545. doi: 10.1007/978-3-642-40480-1_37

European Commission. (2013). *CE marking - gain access to the European market.* Retrieved from http://www.tuv-sud.co.uk/uk-en/activity/product-certification/european-approvals/ce-marking

Fields, R. E. (2001). *Analysis of erroneous actions in the design of critical systems.* Unpublished doctoral dissertation, University of York.

Garmer, K., Liljegren, E., Osvalder, A.-L., & Dahlman, S. (2002, March). Application of usability testing to the development of medical equipment. Usability testing of a frequently used infusion pump and a new user interface for an infusion pump developed with a Human Factors approach. *International Journal of Industrial Ergonomics*, *29*(3), 145–159. doi: 10.1016/S0169-8141(01)00060-9

Gimblett, A., & Thimbleby, H. (2010). User interface model discovery: towards a generic approach. In *Proceedings of the 2nd ACM SIGCHI symposium on Engineering interactive computing systems - EICS '10* (pp. 145–154). New York, New York, USA: ACM Press. doi: 10.1145/1822018.1822041

Ginsburg, G. (2005, June). Human factors engineering: a tool for medical device evaluation in hospital procurement decision-making. *Journal of biomedical informatics*, *38*(3), 213–9. doi: 10.1016/j.jbi.2004.11.008

Glendon, A. I., Clarke, S., & McKenna, E. (2006). *Human safety and risk management.* CRC Press.

Graham, M. J., Kubose, T. K., Jordan, D., Zhang, J., Johnson, T. R., & Patel, V. L. (2004, November). Heuristic evaluation of infusion pumps: implications for patient safety in Intensive Care Units. *International Journal of Medical Informatics*, *73*(11-12), 771–9. doi: 10.1016/j.ijmedinf.2004.08.002

Hesselmann, T., Heuten, W., & Boll, S. (2011). Tap2Count: Numerical Input for Interactive Tabletops. In *Proceedings of the ACM International Conference on Interactive Tabletops and Surfaces* (pp. 256–257). New York, NY, USA: ACM. doi: 10.1145/2076354.2076403

Horsky, J., Zhang, J., & Patel, V. L. (2005, August). To err is not entirely human: complex technology and user cognition. *Journal of Biomedical Informatics*, *38*(4), 264–6. doi: 10.1016/j.jbi.2005.05.002

Institute of Safe Medication Practices Canada. (2007). *Fluorouracil Incident Root Cause Analysis.*

Isokoski, P., & Käki, M. (2002). Comparison of two touchpad-based methods for numeric entry. In *Proceedings of the SIGCHI conference on Human factors in computing systems Changing our world, changing ourselves - CHI '02* (pp. 25 – 32). New York, New York, USA: ACM Press. doi: 10.1145/503376.503382

Jaspers, M. W. M. (2009, May). A comparison of usability methods for testing interactive health technologies: methodological aspects and empirical evidence. *International Journal of Medical Informatics*, *78*(5), 340–53. doi: 10.1016/j.ijmedinf.2008.10.002

Koppel, R., Wetterneck, T., Telles, J., & Karsh, B. (2008). Workarounds to Barcode Medication Administration Systems: Their occurrences, causes, and threats to patient safety. *Journal of the American Medical informatics Association.* doi: 10.1197/jamia.M2616.Introduction

Koppel, R. J., & Gordon, S. (2012). *First, Do Less Harm: Confronting the Inconvenient Problems of Patient Safety (The Culture and Politics of Health Care Work).* New York, New York, USA: ILR Press.

Lazar, J., Feng, J. H., & Hochheiser, H. (2009). *Research methods in Human-Computer Interaction.* John Wiley & Sons.

Lee, P. T., Thompson, F., & Thimbleby, H. (2012). Analysis of Infusion Pump Error Logs and their Significance for Health Care. *British Journal of Nursing*, *21*(64).

Li, K. Y., Xing, S. B., Sun, S., Liu, E., Di, J., Wang, J., … Lewis, A. (2013). MediCHI: Safer Interaction in Medical Devices. In *CHI '13 Extended Abstracts on Human Factors in Computing Systems on - CHI EA '13* (p. 3267). New York, New York, USA: ACM Press. doi: 10.1145/2468356.2479663

Li, Y., Oladimeji, P., Monroy, C., **Cauchi, A.**, Thimbleby, H., Furniss, D., … Blandford, A. (2011, December). Design of interactive medical devices: Feedback and its improvement. In *2011 IEEE International Symposium on IT in Medicine and Education* (pp. 204–208). IEEE. doi: 10.1109/ITiME.2011.6132022

Lin, L., Vicente, K. J., & Doyle, D. J. (2001, August). Patient safety, potential adverse drug events, and medical device design: a human factors engineering approach. *Journal of Biomedical Informatics*, *34*(4), 274–84. doi: 10.1006/jbin.2001.1028

Martin, J. L., Clark, D. J., Morgan, S. P., Crowe, J. a., & Murphy, E. (2012, January). A user-centred approach to requirements elicitation in medical device development: a case study from an industry perspective. *Applied Ergonomics*, *43*(1), 184–90. doi: 10.1016/j.apergo.2011.05.002

Masci, P., Ayoub, A., Curzon, P., & Lee, I. (2013). Model-based development of the Generic PCA infusion pump user interface prototype in PVS. In *In Proceedings of the 32nd International Conference on Computer Safety, Reliability and Security, SAFECOMP* (pp. 228–240). doi: 10.1007/978-3-642-40793-2_21

Masci, P., Rukšėnas, R., Oladimeji, P., **Cauchi, A.**, Gimblett, A., Li, Y., ... Thimbleby, H. (2013, April). The benefits of formalising design guidelines: a case study on the predictability of drug infusion pumps. *Innovations in Systems and Software Engineering*, 1–21. doi: 10.1007/s11334-013-0200-4

Masci, P., Rukšėnas, R., Oladimeji, P., **Cauchi, A.**, Li, Y., Curzon, P., ... Gimblett, A. (2011). On formalising interactive number entry on infusion pumps. *Electronic Communications of the EASST*, *45*, 1 – 15.

NHS National Patient Safety Agency. (2008). A risk matrix for risk managers. (January).

Nielsen, J. (1993, November). Iterative user-interface design. *Computer*, *26*(11), 32–41. doi: 10.1109/2.241424

Nielsen, J. (1994). *Usability Engineering.* Morgan Kaufmann Publishers.

Niezen, G. (2013). *A continuous interaction approach to interactive medical device design.* Paris, France.

Norman, D. (2002). *The design of everyday things.* MIT Press.

NSAI Standards. (2012). I.S. EN ISO 14971:2012 Medcal devices - Application of risk management to medical devices. , *14971*.

Oladimeji, P. (2012). Towards safer number entry in interactive medical systems. In *Proceedings of the 4th ACM SIGCHI Symposium on Engineering Interactive Computing Systems* (pp. 329–332). New York, NY, USA: ACM. doi: 10.1145/2305484.2305543

Oladimeji, P., Li, Y., **Cauchi, A.**, Eslambolchilar, P., Gimblett, A., Lee, P., & Thimbleby, H. (2011). Visualising Medical Device Logs. *1st BCS Health Wales workshop*.

Oladimeji, P., Thimbleby, H., & Cox, A. (2011). Number entry interfaces and their effects on error detection. In *In Proceedings of the 13th IFIP TC 13 International Conference on Human-Computer Interaction* (pp. 178–185). Springer-Verlag.

Oladimeji, P., Thimbleby, H., & Cox, A. (2013). A performance review of number entry interfaces. In *Human-Computer Interaction, INTERACT 2013* (pp. 365–382). doi: 10.1007/978-3-642-40483-2_26

Ostrom, C. M. (2011, April). Nurse's suicide follows tragedy. *The Seattle Times Company*.

Polson, P. G., Lewis, C., Rieman, J., & Wharton, C. (1992, May). Cognitive walkthroughs: a method for theory-based evaluation of user interfaces. *International Journal of Man-Machine Studies*, *36*(5), 741–773. doi: 10.1016/0020-7373(92)90039-N

Reason, J. (1990). *Human Error.* Cambridge University Press.

Reason, J. (2000, March). Human Error: Models and Management. *BMJ (Clinical research ed.)*, *320*(7237), 768–770.

Rich, S. (2008, June). How human factors lead to medical device adverse events. *Nursing*, *38*(6), 62–3. doi: 10.1097/01.NURSE.0000320363.32444.d8

Shankar, N. (1996). PVS: Combining specification, proof checking, and model checking. In M. Srivas & A. Camilleri (Eds.), *Formal Methods in Computer-Aided Design* (Vol. 1166, p. 257-264). Springer Berlin Heidelberg. doi: 10.1007/BFb0031813

Shneiderman, B., Plaisant, C., Cohen, M., & Jacobs, S. (2013). *Designing The*

User Interface: Strategies for Effective Human-Computer Interaction (Fifth Edit ed.). Addison-Wesley Publishing Company.

Shuren, J. E. (2010). *Medical Devices Letter to Infusion Pump Manufacturers.*

Simon, H. A. (1996). *The sciences of the artificial (3rd ed.).* Cambridge, MA, USA: MIT Press.

Stanton, N. A. (2003). Human-error identification in human–computer interaction. *The human-computer interaction handbook*, 371383.

Stanton, N. A., & Baber, C. (2005). Validating task analysis for error identification: reliability and validity of a human error prediction technique. *Ergonomics*, *48*(9), 1097–1113.

Cauchi, A. (2012). Differential formal analysis: evaluating safer 5-key number entry user interface designs. *EICS '12 Proceedings of the 4th ACM SIGCHI symposium on Engineering interactive computing systems*, 317–320. doi: 10.1145/2305484.2305540

Cauchi, A. (2013). Using Differential Formal Analysis for Dependable Number Entry. In *EICS'13 Proceedings of the 5th ACM SIGCHI symposium on Engineering interactive computing systems* (pp. 155–158). doi: 10.1145/2480296.2480339

Cauchi, A., Gimblett, A., Thimbleby, H., Curzon, P., & Masci, P. (2012). Safer 5-key number entry user interfaces using Differential Formal Analysis. *Proceedings of HCI 2012, The 26th BCS Conference on Human Computer Interaction*, 29–38.

Cauchi, A., Thimbleby, H., Oladimeji, P., & Harrison, M. (2013). Using medical device logs for improving medical device design. In *Proceedings of the IEEE International Conference on Healthcare Informatics* (pp. 56–65).

Thimbleby, H. (2007a). Interaction walkthrough: evaluation of safety critical interactive systems. In *Proceedings of the 13th international conference on Interactive systems: Design, specification, and verification* (pp. 52–66). Berlin, Heidelberg: Springer-Verlag.

Thimbleby, H. (2007b). *Press On — Principles of Interaction Programming.* MIT Press.

Thimbleby, H. (2013a). Improving Safety in Medical Devices and Systems. In *Proceedings of the International Conference on Health Informatics* (pp. 1–13). IEEE.

Thimbleby, H. (2013b). Reasons to question seven segment displays. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems - CHI '13*, 1431. doi: 10.1145/2470654.2466190

Thimbleby, H., & Cairns, P. (2010, October). Reducing number entry errors: solving a widespread, serious problem. *Journal of the Royal Society, Interface / the Royal Society*, *7*(51), 1429–39. doi: 10.1098/rsif.2010.0112

Thimbleby, H., Gimblett, A., & **Cauchi, A.** (2011). Buffer Automata: A UI Architecture Prioritising HCI Concerns for Interactive Devices. In *Proceedings of the 3rd ACM SIGCHI symposium on Engineering interactive computing systems - EICS '11* (p. 73). New York, New York, USA: ACM Press. doi: 10.1145/1996461.1996497

Thimbleby, H., **Cauchi, A.**, & Gimblett, A. (2012). Simulation to evaluate alternative approaches to blocking use errors. *Journal of medical devices*, *6*(1), 017502.1.

Thimbleby, H., **Cauchi, A.**, Gimblett, A., & Oladimeji, P. (2011). Goal-based design improves interaction dependability. *Proceedings of the Digital Engagement Conference, 2011*, 8.

US FDA. (2013). *510(k) Submission Process.* Retrieved 28/12/2013, from `http://www.fda.gov/medicaldevices/deviceregulationandguidance/howtomarketyourdevice/premarketsubmissions/premarketnotification510k/ucm070201.htm`

US Food and Drug Administration. (2010). *Guidance for Industry and FDA Staff-Total Product Life Cycle: Infusion Pump-Premarket Notification 510k Submissions* (Vol. 510). Retrieved from `http://www.fda.gov/medicalDevices/DeviceRegulationandGuidance/GuidanceDocuments/ucm206153.htm` van Someren, M. W., Barnard, Y. F., & Sandberg, J. A. (1994). *The think aloud method: A practical guide to modelling cognitive processes.* Academic Press.

Vicente, K. J., Kada-Bekhaled, K., Hillel, G., Cassano, A., & Orser, B. A. (2003). Programming errors contribute to death from patient-controlled analgesia: case report and estimate of probability. *Canadian Journal of Anaesthesia*, *50*(4), 328–332.

Ward, J. R., & Clarkson, P. J. (2004). An analysis of medical device-related errors: prevalence and possible solutions. *Journal of medical engineering & technology*, *28*(1), 2–21. doi: 10.1080/0309190031000123747

Webster, J., Eslambolchilar, P., & Thimbleby, H. (2012). From rotary telephones to universal number entry systems. In *Proceedings of the 2012 ACM Conference on Ubiquitous Computing - UbiComp '12* (p. 596). New York, New York, USA: ACM Press. doi: 10.1145/2370216.2370322

Wiegmann, D., & Shappell, S. (2012). *A Human Error Approah to Avaiation Accident Analysis: The Human Factors Analysis and Classification System.* Ashgate Publishing.

Wiseman, S. (2012). *A Case for Number Entry.* Unpublished doctoral dissertation, University College London.

Wiseman, S., Cairns, P., & Cox, A. (2011). A taxonomy of number entry error. In *BCS-HCI '11 Proceedings of the 25th BCS Conference on Human-Computer Interaction* (pp. 187–196).

Wiseman, S., Cox, A., & Brumby, D. (2012). Designing for the taskL What Numbers are Really Used in Hospitals? In *Proceedings of the 2012 ACM annual conference extended abstracts on Human Factors in Computing Systems Extended Abstracts - CHI EA '12* (p. 1733). New York, New York, USA: ACM Press. doi: 10.1145/2212776.2223701

Wiseman, S., Cox, A. L., & Brumby, D. P. (2013, January). Designing Devices With the Task in Mind: Which Numbers Are Really Used in Hospitals? *Human Factors: The Journal of the Human Factors and Ergonomics Society*, *55*(1), 61–74. doi: 10.1177/0018720812471988

Wiseman, S., Cox, A. L., Brumby, D. P., Gould, S. J., & O'Carroll, S. (2013). Using checksums to detect human error. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems - CHI '13* (p. 2403). New York, New York, USA: ACM Press. doi: 10.1145/2470654.2481332

Wolfram, S. (2003). *The Mathematica Book* (5th ed.). Wolfram Media, Incorporated.

Yamaoka, T., & Baber, C. (2000). Three point task analysis and human error estimation. In *Proceedings of the human interface symposium 2000* (pp. 395–398).

Zhang, J., Patel, V. L., Johnson, T. R., & Shortliffe, E. H. (2004, June). A cognitive taxonomy of medical errors. *Journal of Biomedical Informatics*, *37*(3), 193–204. doi: 10.1016/j.jbi.2004.04.004

Zhang, Y., Jones, P. L., & Jetley, R. (2010, March). A hazard analysis for a generic insulin infusion pump. *Journal of diabetes science and technology*, *4*(2), 263–83.