Bowen, J. & Coben, D. (2019). Safety First: Combining Task Models of Medical Devices with Numeracy Skills and Technical Competence. *Adults Learning Mathematics: An International Journal*, *14*(2), 39-60.

Safety First: Combining Task Models of Medical Devices with Numeracy Skills and Technical Competence

Judy Bowen
University of Waikato, New Zealand
<jbowen@waikato.ac.nz>
Diana Coben
King's College London and University of East Anglia, UK
<diana.coben@kcl.ac.uk>

Abstract

We propose that by more closely aligning interdisciplinary work in (a) numeracy education for medication dosage calculations and (b) model-driven design for medical devices that are used for delivery of medication we may help address the incident-rate in incorrect medication calculations and delivery, given that such devices commonly require the user to engage with numerical information via a digital interface. We demonstrate the use of task models as a way of supporting safe, effective and efficient delivery of medication to the patient, taking as our example the use of infusion and syringe pumps in Nursing. This work indicates a new way of facilitating knowledge transfer between numeracy education and medical device design and usage, using task models. We aim to support medical professionals' and students' numeracy education as well as to inform the design of medical devices based on a better understanding of the use and potential errors of medication delivery by trained professionals.

Key words: Task models; safety-critical systems; Nursing numeracy; syringe pumps; human-centred computing; interaction design.

Introduction

In the course of their work, Nurses and other healthcare professionals frequently engage with medical equipment via digital interfaces. Those interfaces typically involve activities such as entering numbers, checking that equipment is correctly calibrated, measuring and recording data, etc., and sometimes calculation. Patient safety is paramount so the stakes are high.

This article outlines interdisciplinary work in progress bringing together the authors' research interests and expertise in computer science and adult numeracy in safety-critical work contexts in order to address an important issue: how best to support safe, effective and efficient delivery of medication to the patient when that delivery involves the healthcare professional engaging with medical equipment via a digital interface?

Our research interests and expertise are as follows. Bowen is a Computer Science researcher in formal methods and modelling for safety critical software. She has been focusing on medical devices such as syringe drivers and infusion pumps for the past seven years. She is interested in users, usability, correctness and the safety properties of medical devices. Coben is a specialist in adult numeracy practices and education, especially in safety-critical work contexts, who has a particular research interest in numeracy for Nursing spanning 25 years.

In the safety-critical context of Nursing, Bowen's research has focused on: engineering interactive safety-critical systems such as medical infusion pumps and syringe drivers; seeking to ensure devices behave correctly at all times; and seeking to improve design so that devices are easier and safer to use. Meanwhile, Coben's research has sought to identify and support the development of numeracy skills required to practise safely, effectively and efficiently, including calculating and delivering medication dosages correctly using diverse delivery mechanisms, many of which have digital interfaces. She is a member of an international interdisciplinary team investigating medication dosage calculation problem-solving (MDC-PS).

We bring together concepts of technology design (both functional correctness and usability concerns), numeracy and medication delivery competency. We consider the use of task models (outlined below) as a step towards achieving this. Our wider research seeks to address the following questions:

- How can we use formal models of medical devices in conjunction with task models to improve the design and development process?
- How can we use task models to identify mismatches in user knowledge and education with device usage?
- How can we use task models of medication calculations and technical competence to inform user education and the design of medical devices?
- How can we use comparisons of task models of devices and medication delivery to identify potential use errors?

The work in this paper outlines the groundwork needed to answer these questions by examining the use of task models in the domains of medical device design and usage in the context of the development of competency for nurses. These questions relate to all medical personnel. In this paper we focus on nurses because they are most likely to deliver medication directly to the patient: as such, they are the last line of defence in medicines management in the healthcare context (Gottlieb, 2012).

Although we discuss a number of different types of models in this work, the key focus is on task models, which we will use in a number of different ways. Task models are a description of a goal that a user wishes to achieve. They can be used abstractly to describe user actions as a set of steps and subtasks required to achieve the goal (via decomposition of the high level goal) and can also be used to describe concrete steps and actions a user may take to achieve that goal (for example, when using a medical device). Task models provide documentation comprising flexible and expressive notations with precise semantics and support effective design of devices and their evaluation in terms of usability (Paternò, Santoro, & Spano, nd). We believe task models have the potential to be a useful tool for educators seeking to support safe and effective healthcare numeracy practice through authentic education. We give a more detailed description of task modelling later.

Problems with medical devices

Common problems with medical devices include:

- Number entry: e.g., confusing methods of number entry; varying methods of number position and decimal point handling; number roll-over problems;
- 'Same but different': i.e., devices that look identical but have different firmware, leading to subtle differences in use;
- 'Similar but different': i.e., devices that look similar but behave differently.

In this paper we focus on syringe drivers. These are devices designed to deliver intravenous medication to the patient automatically, obviating the need for multiple injections. Intravenous medication forms a major part of hospital inpatient care throughout the world, and most of this is delivered via infusion pumps and syringe drivers. These and other medical devices are considered to be safety-critical interactive systems. That is, they are systems or devices that involve user interaction and which have the potential to be hazardous if they malfunction or if user errors occur. Infusion pumps and syringe drivers have been implicated in numerous adverse events (i.e., injuries "resulting from a medical intervention", Kohn, Corrigan, & Donaldson, 2000), the results of which range from minor patient inconvenience through to serious harm or even death (FDA, 2017).

A recent UK example illustrates the problem of 'same but different' devices. In this case, the devices are the Graseby MS16A and MS26 syringe pumps, both of which were used in UK palliative care until they were phased out in 2015. An article in *The Daily Telegraph* newspaper reported:

> While one model is set to deliver opioids over 24 hours, the other, which looks identical, performs the same task in 60 minutes, meaning patients risk being given a day's worth of drugs in just one hour.
> Doctors have said they resorted to sticking on makeshift aluminium strips so they could tell the difference.
> […]
> Dr Richard Ian Reid […] told police he could not remember which Graseby device was which, saying they were "totally confusing" and "really dangerous".
> (Bodkin, 2018)

This is symptomatic of an ongoing global problem. Between 2005 and 2009 the FDA received approximately 56,000 reports of adverse events associated with the use of infusion pumps, including numerous injuries and deaths. In the same period manufacturers recalled 87 infusion pumps to address identified safety concerns. Since 2009 the situation has improved very little, despite increased focus on these issues.

Underlying causes of these problems include: user interfaces that are not fit for purpose; inappropriate interaction between elements of devices; and issues in user engagement with devices - all factors that are not treated by manufacturers or regulators with the same rigour as the underlying functionality of the device. The lack of a common method for verifying device correctness prior to FDA or other regulatory approval also causes problems, as does the absence of common methods for reasoning about user interaction at a formal level to allow for model-checking and theorem-proving.

The FDA has launched a project aimed at improving this situation (the Generic Infusion Pump project[1]) which focusses on the design and development of the technology. This work, and similar research in this domain, is based around either software engineering - seeking to ensure the devices behave correctly at all times (Campos & Harrison, 2011), or usability research - seeking to improve the design so that devices are easier to use (Thimbleby, 2015). However, there are alternative, complementary approaches in other disciplines which have the same goal of reducing medication delivery errors. These include training medical personnel in the use of medical devices (Vipond, 2016) and in the numeracy skills required to calculate medication dosages correctly for different delivery mechanisms (Coben & Weeks, 2014) in order to ameliorate the risk of user error.

When we focus on 'user error' as a technological problem we make assumptions. For example, if an error is made leading to the wrong amount (volume) of medication being delivered to a patient we may assume that the medical professional setting up the infusion (whom we henceforth refer to as the 'user') has started with the correct values for setting up an infusion and made a number-entry error. Indeed, number-entry errors are a common occurrence in this domain (Thimbleby, 2015). However, in

[1] https://rtg.cis.upenn.edu/gip/

practice it may be impossible to determine at which point the error has actually been made. It may be that rather than a number entry error, it is in fact a calculation error in which the user starts the infusion set-up with wrongly calculated values that they subsequently enter into the device correctly.

Numeracy education for medical staff focusses on ensuring that they are able to set up and perform the necessary calculations to convert prescribed medication into the appropriate values and measures for their delivery mechanisms, which may be tablets of a given strength (e.g., 5mg), or liquid solutions containing given amounts of medication (e.g., 20mg per 2ml). It is essential that this precursor to actual medication delivery is correct and that the medical personnel can then safely and correctly deliver the correct medication doses using the technology provided.

A possible solution

We are working towards finding an effective solution to the problem outlined above through our proposed programme of interdisciplinary research. This has two main elements: engineering interactive systems; and authentic numeracy education based on a sound model of competence. The engineering element entails using model-driven development techniques for interactive systems in the design and implementation of medical devices. Models of software and hardware/software hybrid systems (or cyber-physical systems) are abstract descriptions focussing on specific properties of that system. The models can be at differing levels of formality where informal designs and prototypes can be used to describe the 'look and feel' of the interface to a system to non-technical stakeholders; and fully formal mathematically-based models of functionality can be used for formal reasoning based on logical tools such as model-checkers. The authentic numeracy education element is based on a sound model of competence which was developed for medication dosage calculation problem solving (MDC-PS), and based on research set out in the 'Safety in Numbers' Virtual Special Issue of *Nurse Education in Practice*[2]. These two elements are described below.

Figure 1, below, gives an overview of our approach. The arrows are labelled with the key steps of our research. Step 1 relates to research questions 1 and 2 above. In this paper we describe how we can use existing models of interactive systems (particularly formal models of safety critical systems) (Bowen & Reeves, 2008, 2013) to derive task models and interaction sequences, and how this provides information about use and user knowledge. We discuss Step 2 from the perspective of technical competence models designed to inform numeracy education for healthcare personnel, particularly nurses (Weeks et al., 2013). Finally, we demonstrate how questions 3 and 4 can be addressed by relating these diverse task models to each other and comparing them (Step 3).

This work indicates a new way of facilitating knowledge transfer between numeracy education and medical device design and usage, using task models. We aim to support healthcare professionals' and students' numeracy education as well as to inform the design of medical devices based on a better understanding of the use of, and potential errors in medication delivery by trained healthcare professionals.

We discuss this next in the context of medication dosage calculation problem solving (MDC-PS) in Nursing.

[2] https://www.nurseeducationinpractice.com/content/safety

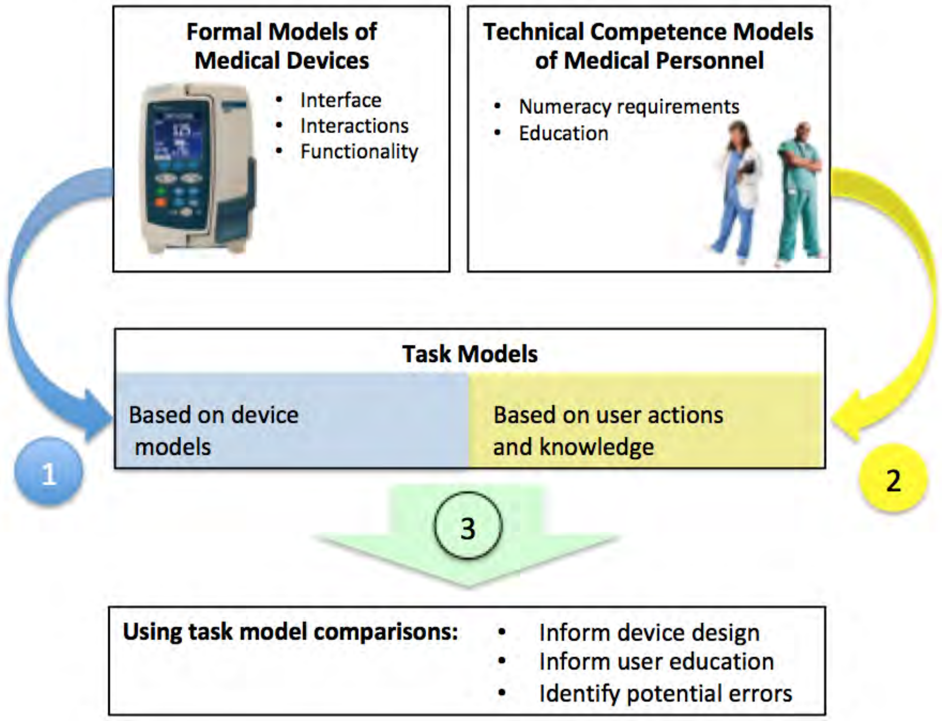

Figure 1: Overview

Competence in medication dosage calculation problem solving in Nursing

Weeks et al. (2013) have proposed a competence model for MDC-PS which represents the intersection between the ability to interpret the dosage calculation problem and accurately set up rate equations (conceptual competence), the correct calculation of accurate numerical values for the dose and rate of administration (calculation competence) and the selection of appropriate measurement vehicles and accurate measurement of the dose and rate of administration (technical measurement competence), see Figure 2 (Weeks et al., 2013, p. e25).

Specifically, the MDC-PS competence domains are characterised as follows:

1. Conceptual competence refers to the need to:

- Understand the elements of prescription charts, dispensed medication labels and medication data sheets and monographs, and to subsequently extract the numerical information necessary to set up the dosage problem correctly.
- Position the numerical information appropriately and correctly in an equation format for calculation.

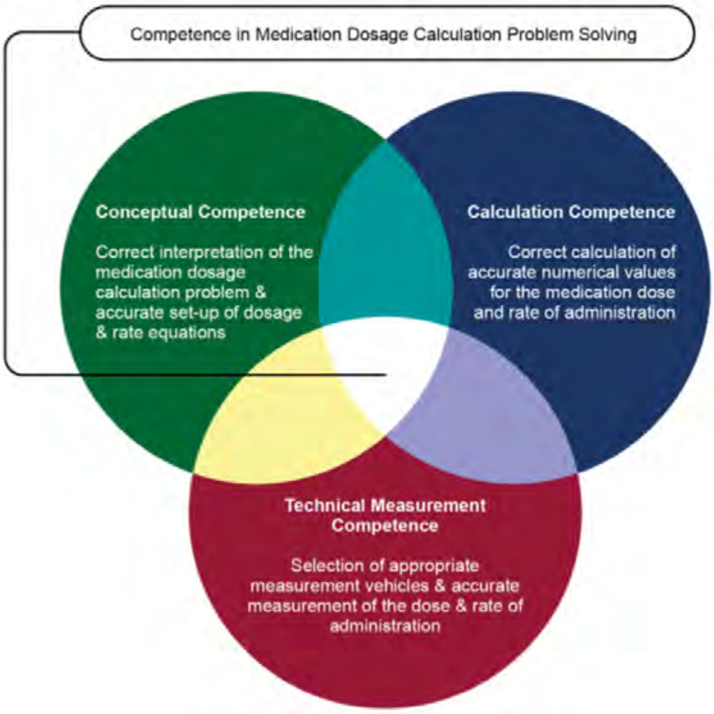

Figure 2 MDC-PS competence model (Weeks et al., 2013, p. e25)

2. Calculation competence refers to the need to:

- Correctly apply arithmetical operations and compute an accurate numerical value within a safe and acceptable tolerance range for the prescribed medication dose, and/or rate of administration.

3. Technical Measurement competence refers to the need to:

- Select an appropriate medication administration measurement vehicle (tablet or capsule, oral liquid medicine measurement cup, syringe, infusion pump, etc.)
- Accurately transform the calculated numerical value to the context of the measurement device/formulation and measure the correct dose of prescribed medication; and/or administer the correct rate of prescribed medication/IV infusion fluid.

The competence model for MDC-PS comprises the intersection between: the ability to interpret the dosage calculation problem and accurately set up rate equations (conceptual competence); the correct calculation of accurate numerical values for the dose and rate of administration (calculation competence); and the selection of appropriate measurement vehicles and accurate measurement of the dose and rate of administration (technical measurement competence). An uncorrected error in any one or more of these domains will result in a medication dosage error in the practice setting (Weeks et al., 2013). Technical measurement competence is particularly relevant to our focus in this paper since it involves the use of medication delivery devices.

Model-based design for interactive systems

Model-based design and engineering for safety-critical software and hardware (such as the syringe driver) take many forms and can be considered across two different dimensions. The first dimension is the type of model and where it falls on the scale of formality - from fully formal approaches, such as the use of formal specifications, verification, refinement, etc. (Blandford et al., 2011; Bowen & Reeves, 2007; Harrison, Masci, Campos, & Curzon, 2017) to the informal methods of user-centred design and more 'agile' design approaches (Blandford, Buchanan, Curzon, Furniss, & Thimbleby, 2010) as well as everything in between (Calvary, Coutaz, & Thevenin, 2001). The second dimension is how, and where, the models are used: at the design stage before any implementation occurs; as part of the final testing and sign-off of the implemented solution; or throughout the design process.

Other research approaches seek to consider the ranges of both dimensions by focussing on the integration of formal specification techniques (ideally suited in safety-critical domains) with user-centred design approaches (ideally suited for interactive systems) (Bowen & Reeves, 2006). This allows consideration of both design and functionality using formal and informal methods at the requirements stage and throughout development. It also lends itself to integrated testing approaches as well as post-implementation and reverse-engineering analysis methods (Bowen, 2015; Bowen & Reeves, 2008). While much of this work does not explicitly incorporate task models, they are usually considered as implicit inputs into the user-centred design approach that forms the basis for user interface (UI) and interaction models.

Task analysis and task models in interactive system design

Task analysis is aimed at understanding how a user completes a defined task. It allows us to both analyse what a person is required to do to achieve a certain goal (the task) as well as analyse the effort (both cognitive and physical) required to do this. There are a large number of methods, notations and tools used for task analysis within both computer science and psychology (where the origins of task analysis can be found in applied behaviour analysis). The choice of which to use typically depends on the formality of the design process and the use of the task model within that process.

While basic hierarchical decomposition may be sufficient in the early stages of development to support understanding of user requirements, and proved popular when task analysis began to be incorporated into the domain of human-computer interaction (HCI) (see Shepherd, 1989, for example), the use of task models has evolved within interactive system design and development to include goal-based analysis and conceptual models which are used at various stages throughout the design life-cycle. Work by Paternò et al. has extended this further to a comprehensive notation for both task and dialog modelling based on concur-task trees (CTT) (Paternò, Mancini, & Meniconi, 1997) which includes a variety of logical and temporal operators for ordering and iteration, as well as tools to support the creation of, and reasoning about, such models (Mori, Paternò, & Santoro, 2002).

Palanque and colleagues have incorporated these CTT into a petri-net based modelling and development environment for use within safety-critical interactive system design (Barboni, Ladry, Navarre, Palanque, & Winckler, 2010; Martinie, Palanque, Ragosta, & Fahssi, 2013). These types of extensions, and others (e.g., Dittmar & Forbrig, 2003) allow the use of straightforward hierarchical models for complex reasoning about safety, user collaborations, reconfigurable human-machine interfaces, etc.. As such, their work demonstrates the flexibility of task modelling and the ability to incorporate it into formal modelling for a wider range of uses. This supports the goals of our work and motivates us in the use of task models in the manner we propose.

Of most relevance here is the use of Palanque and colleagues' petri-net environment to support training of operators of safety-critical systems. This takes a similar approach to our own, by using task models to link specific device models to another domain (in their case, training procedures and programs) (Martinie, Palanque, Navarre, Winckler, & Poupart, 2011). While their aims are different in that they seek to develop appropriate training programs by integrating task models within the simulation environment PetShop (Palanque, Ladry, Navarre, & Barboni, 2009) the use of task-models to bridge cross-domain knowledge demonstrates their applicability in such approaches.

Integrated tools and simulation environments such as PetShop provide the ability to incorporate different concepts and approaches (in this case, task models, interactive system models and training procedures) into a single environment. However, the downside of this is that in order to take advantage of such a tool, everything has to be modelled and developed within this one tool. Frequently, the diversity of artefacts within interactive system development and the heterogeneity of different groups within the design team means this is not the most suitable choice.

Unlike the approaches seen in Instructional Task Analysis, which use task decomposition as a means to decide what skills and knowledge are required by users of a particular system, here we look at the skills and knowledge of clinicians required to administer medication and compare them with the tasks of using technological systems to deliver such medication.

Modelling users

Although task models describe user behaviours, these are at a level of the actions required to achieve a goal. That is, they typically assume correct or optimal behaviours. Comparisons between such models and actual user behaviours in practice can be used to identify, and even model, errors based on missteps or slips (see Johnson, 2011, for example) but task models alone do not identify such errors. More frequently they might be used as part of approaches such as “key-stroke level models” (KLM) (Card, Moran, & Newell, 1980) and “goals, operators, methods and selection” (GOMS) (Card, Moran, & Newell, 1983) which seek to identify cognitive load and effort, which in turn may suggest potential for error.

In contrast, work that does seek to model user cognition and identify potential errors based on this makes different types of assumptions. Blandford et al. have focussed on the effect cognition has on user behaviour (Blandford, Butterworth, & Good, 1997) and extended this to consider distributed cognition for use with multi-user systems (Blandford & Furniss, 2005). More recently they have looked at the effect distributed cognition has on healthcare practices (Berndt, Furniss, & Blandford, 2015; Rajkomar & Blandford, 2012) both in clinical settings and in the home. We also consider the work of Curzon et al. who use salience as an important property in understanding cognition and the effect this has on interactive system design (Rukšėnas, Back, Curzon, & Blandford, 2008). These examples categorise different types of causal effects (distributed behaviours, salience) and their potential to lead to error and then use these to either improve processes or improve the design of safety-critical systems. Rather than consider user behaviours based on such concepts, we instead focus on the driving factor, the key knowledge that users have (or should have) prior to undertaking particular tasks and the competence they demonstrate in performing these tasks. This professional knowledge base has been synthesised in the European Qualifications Framework for Lifelong Learning (EQF) as “The proven ability to use knowledge, skills and personal, social and/or methodological abilities, in work or study situations and in professional and personal development” (European Communities Education and Culture, 2008). This notion of professional clinical competence underpins our research.

Creating links using task models

Formal models of medical devices

As discussed in the previous section, a variety of different models and modelling techniques exist for interactive systems. Here we focus on the presentation model approach described in Bowen and Reeves (2017) which uses several different models at varying levels of abstraction throughout the design and development process. This enables both formal verification of properties such as safety, as well as supporting prototyping and lightweight UI design. Interface designs can be described by the interactive elements (widgets) of the design and their intended behaviours. As such, a formal structure can be given to the narrative behind prototypes, personas, storyboards, etc., which (among other things) removes ambiguity. Although a full explanation of these is beyond the scope of this paper we introduce the basic elements by way of an example which we subsequently expand upon.

Consider the syringe driver shown in Figure 3. This is a modal device, that is, the behaviour of each of its buttons is dependent on the current mode of the system. A simple presentation model can be constructed which describes each mode as a collection of the available widgets. These are described in a tuple (a finite sequence of elements) giving a name to the widgets, their type (input/output) and their intended behaviour. For example, the device has a mode which enables a user to enter the volume of medication to be infused, which we call 'SetVolume'. The presentation model of the 'SetVolume' mode might include:

SetVolume is

OnButton, ActionControl, (I_Init),

UpButton, ActionControl, (S_IncVolume),

DownButton, ActionControl, (S_DecVolume),

Display, Responder, (S_IncVolume, S_DecVolume),

..

..

as well as the rest of the widgets, which we omit here for brevity. Each mode of the device is described in a similar way. The button behaviour names are prefixed with either an "I_" which indicates it is a behaviour relating to interface navigation (mode change) or an "S_" if it relates to system functionality.

Figure 3: T34 Syringe driver

The lightweight presentation model can be linked to other models which then give formal meaning and semantics to these simple tuples. Of most interest here is the relationship to user actions to complete tasks and the availability of the widgets and their behaviours in different modes. This is described in a presentation interaction model (PIM) which is a state transition diagram with each state representing the presentation model of a mode and transitions showing the behaviours which enable a user to switch between these modes.

From these formal models of the device we can derive interaction sequences. An interaction sequence is the set of actions and interactions that a user performs with a given system to complete a task (Turner, Bowen, & Reeves, 2017). As such, they describe actions a user undertakes for a given task from a given device state. For example, if the device is in the "SetVolume" mode with a current volume value of "0", then setting the volume to be infused to 10ml might be represented as

"PressUpButton[10]" which is shorthand for "PressUpButton" listed ten times (Bowen & Reeves, 2008). Interaction sequences can be generated automatically from the device models and can be used to represent optimal paths of actions (as in the set volume example here) or may include additional, arbitrary or erroneous actions. Such interaction sequences are, therefore, a type of task model (describing the steps to complete a task) but they are sequential rather than hierarchical and at the lowest level of abstraction.

These types of descriptions are model-specific in that they are explicitly tied to the detail of the formal model. Task models, however, can also be device-specific, where they describe the task as it pertains to a specific device (which we discuss next) or they may be independent of both of these (i.e., generic) and pertain solely to the goals of the user in terms of *what* they want to achieve without the low-level details of *how*.

Generic task models can be used to inform device design because their inclusion in a typical user-centred design (UCD) approach means that the formal models derived from UCD artefacts have this information embodied within them. That is, a prototype or storyboard created to examine initial design ideas is partly based upon the user tasks (as well as requirements, guidelines, safety regulations, etc.).

Model-specific task models, such as interaction sequences, can be used iteratively to help refine design artefacts. For example, our interaction sequence step "PressUpButton[10]" (Bowen & Reeves, 2008) described above may lead to a design evolution where a long press on the 'Up' button increases a value in increments of 10. This leads to shorter interaction sequences for some tasks, which might be seen as better for user experience and usability.

We have introduced a number of different models and types of models here. In Figure 1 we described how different categories of models could be related to each other and derived from each other, in Table 1 we give a summary of the introduced models and show how they fit into this structure.

Table 1: Models used in different parts of the processes

Formal Models of Medical Devices	**Technical Competence Models for Numeracy**
Presentation models of device interfaces	MDC-PS (from Fig. 2)
Interaction sequences of device use	
Task Models	
Based on user tasks with devices	Based on user actions and knowledge
Device specific task models	Numeracy tasks

Task models of medical devices

The syringe driver in Figure 3 is a CME Niki T34 (henceforward referred to as the "T34 syringe driver"). It is used to deliver a pre-determined amount of medication from a syringe to a patient over a pre-defined period of time. The device has eight buttons that the user can interact with, as well as a small screen which provides information and feedback. In order to set up medication delivery the user needs to undertake several different tasks, some using the device (inserting the syringe, setting up the dosage rate, etc.) and some independently from the device (calculating correct volume and time according to the prescribed dose, getting the medication, etc.).

Each of the tasks can be combined into a single action, “Set up infusion”, which can then be broken down into hierarchical steps in a typical task analysis fashion. So, we can either create a model for the generic task that a user would perform with this type of device (device-independent) or specialise it for this actual device (device-dependent). If we again consider the task of setting up an infusion, this can be decomposed into the following 5 sub-tasks:

- turn on syringe driver
- insert syringe into driver
- set volume to be infused
- set time for infusion
- start infusion

There are a number of assumptions made before this task can be carried out which are reliant on previous tasks (such as selecting the appropriate syringe type and size and drawing up the medication) having been successfully completed. A full task model would include all of these, but for now we focus on just the task actions relating to the medical device, which is typical when using such models as part of interactive system development and reasoning.

All of the steps can be broken down further and Steps 3-4 can include both iteration and repetition between the steps. There are also temporal relations between steps (some steps must be completed before others, it is possible to go back to a previous step, etc.). Here we use the CTT notation (Paternò et al., 1997) to describe some of these properties; this is done for convenience as CTT includes the relevant operators to describe these properties as well as an editing tool to create the hierarchical models (Mori et al., 2002). However, there are several other notations that could be used and we do not propose that one particular hierarchical notation is necessarily more suitable than any other.

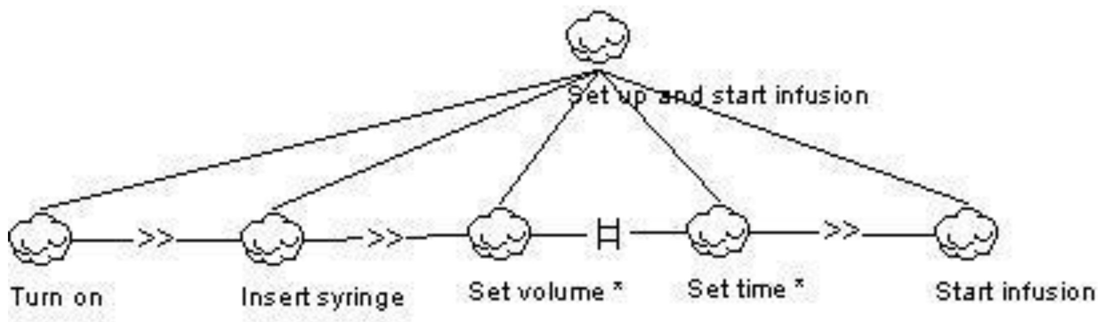

Figure 4: Task model for set up and start infusion

Figure 4 shows the top level view of these tasks. The >> operator signifies a sequence such that the first action must be completed before the next can begin. The |=| signifies that it does not matter in which order these actions are completed, and the * represents iteration. The tasks at the second level of the tree can be further decomposed, for example, in Figure 5 we see the detail of the sub-task required to set the time of the infusion.

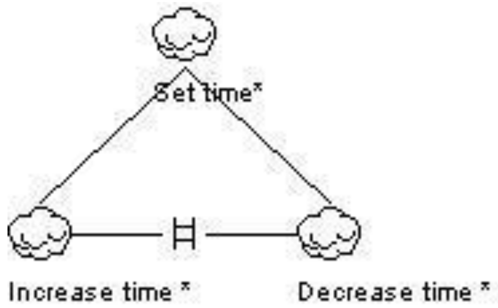

Figure 5: Sub-task model for setting time

The task models shown here focus on user actions only. That is, they do not depend on the particular device or implementation. If we were to specialise the models for the T34 syringe driver, shown in Figure 3, for example, the "Set up infusion" and "Set time" tasks would be represented by the task models shown in Figures 6 and 7.

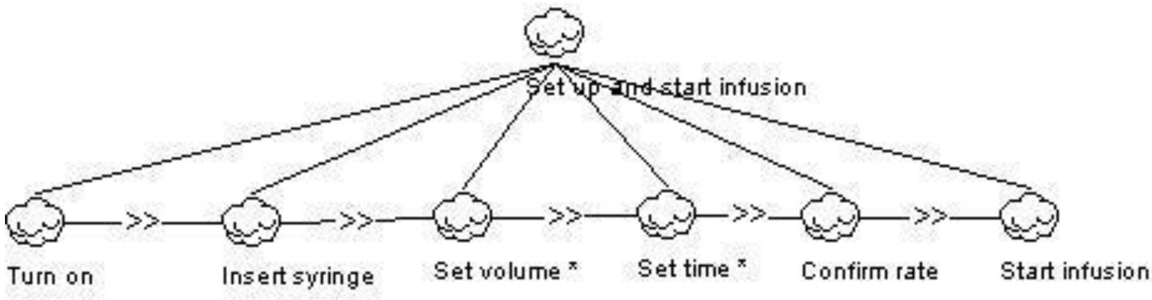

Figure 6: Task model to set up infusion on T34 syringe driver

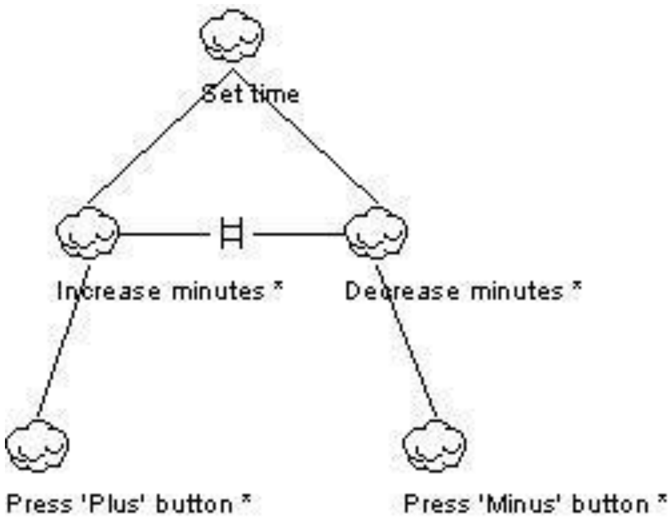

Figure 7: Task model to set time on T34 syringe driver

We can see that the "Set up start infusion" model of Figure 6 differs from the generic task model of Figure 4 in two ways. Firstly there is now a defined ordering between the "Set volume" and "Set time" tasks, as the T34 syringe driver mandates this order. Secondly there is an additional subtask, "Confirm rate", which again is an action that must be performed when using this particular device but which may not be expected by a user. This mismatch between the model-dependent task model and the device-dependent task model indicates that further work is required.

The formal models describe an intended device based on user tasks and subsequent design decisions. If the implemented device differs (as is made apparent by the different task models) then we should identify why/how the changes occurred (were they intended or based on additional requirements not included in the model, for example?). They also suggest a mismatch between user expectations (as embodied in the UI design models) and the actual implementation. Now that it has been identified we might use it to inform user training or to warn of potential user error.

Such comparisons can also be useful when considering multiple devices with similar functionality (or the same devices with different firmware). If defined orderings of actions differ between different instances of devices, this is an area which may also lead to confusion or user error and so enables us to flag a potential problem. A device-specific task model can also be decomposed down to the level of the interaction sequence, so the lowest nodes on the tree represent interactions with actual widgets, such as "press upButton". This allows us to start combining the interaction sequences of device models with model-specific task models to ensure that they are consistent.

In this work we also wish to consider a comparison of task models for devices with models of user intentions based on numeracy skills and technical competence. We discuss this next.

Task models and numeracy/mathematics education

There is an extensive research literature on mathematical task design (Watson & Ohtani, 2015), likewise on mathematical modelling (e.g., Galbraith, Henn, & Niss, 2007), but the literature on 'task models' *per se* relates to human-computer interaction in engineering and computer science (e.g., Paternò, 2001) rather than mathematics or numeracy education. We suggest that bringing insights from these fields together could benefit both. In particular, we suggest that such an integrated approach could improve our understanding of the numeracy demands of various user interface scenarios and the best way of ensuring that users can meet these demands, including through both education and training of users and improved interface design of devices.

Our focus is on user interfaces - and users interfacing - with safety-critical medical equipment involving digital inputs and/or reading, recording and interpreting of outputs by users, on the education and training required for this to be done competently, i.e., efficiently, effectively, and, above all, safely, and on the implications of this for the design of device interfaces.

In so doing we are bringing together research and ways of thinking developed in different academic disciplines and professional domains, with very different relationships to - and conceptions of - the notion of task modelling. In fact, task modelling in the sense outlined above has not hitherto been used in mathematics education research as far as we are aware. The nearest cognate concepts used in the mathematics education field are mathematical modelling and task design.

Mathematical modelling is the process of applying mathematics to a real-world problem with a view to understanding it (Niss, Blum, & Galbraith, 2007) while task design refers to the design of mathematical tasks for educational purposes. As such, as the editors of a recent book detailing an International Commission on Mathematical Instruction (ICMI) study on task design state, "Task design is at the heart of effective mathematics teaching and learning" (Watson & Ohtani, 2015, p. vi). However, "Despite the recent growth spurt of design studies within mathematics education, the specificity of the principles that inform task design in a precise way remains both underdeveloped and, even when somewhat developed, under-reported." (Kieran, Doorman, & Ohtani, 2015, p. 74). Meanwhile, a recent review of the state of the art in mathematical modelling notes its "local" focus, eschewing input from other disciplines:

> In research into the teaching and learning of mathematical modelling there is a strong emphasis on developing “home grown theories” where the focus is on “particular *local theories*” such as the modelling cycle and modelling competencies rather than *general theories* from outside the field (Geiger & Frejd, 2015).
>
> (Stillman, 2019, p. 6)

We hope that tool models may provide such a theoretical (and practical) input from outside the field of mathematics education.

We suggest that meaningful mathematical tasks model activity in the real world in an authentic way (Palm, 2009; Weeks et al., 2013). In this case that means taking into account the various ways in which nurses engage with the mathematical demands of the digital interfaces they encounter in their professional practice. This variability has implications for the specification of the relevant task models. The competence model presented in Figure 2 exemplifies this approach with regard to MDC-PS.

As well as being authentic, we also suggest that such tasks should be mathematically-rich and pitched at an appropriate level of challenge for the learner. The principles of MDC-PS described above have been embodied in an e-learning environment called safeMedicate®[3] which has been developed by Authentic World Ltd. This contains authentic clinical dosage calculation problems and supports the development and assessment of competence in dosage calculation problem solving within five skills-based modules. We present examples from these problems to demonstrate the use of task models as a mechanism for structuring the steps and activities users are required to undertake to complete the problems, which therefore represents the steps required to perform the task in a real clinical setting.

Figure 8 shows a screenshot of one of the example problems from the ‘Advanced Injectable Medicines Therapy, Continuous Infusion’ module.

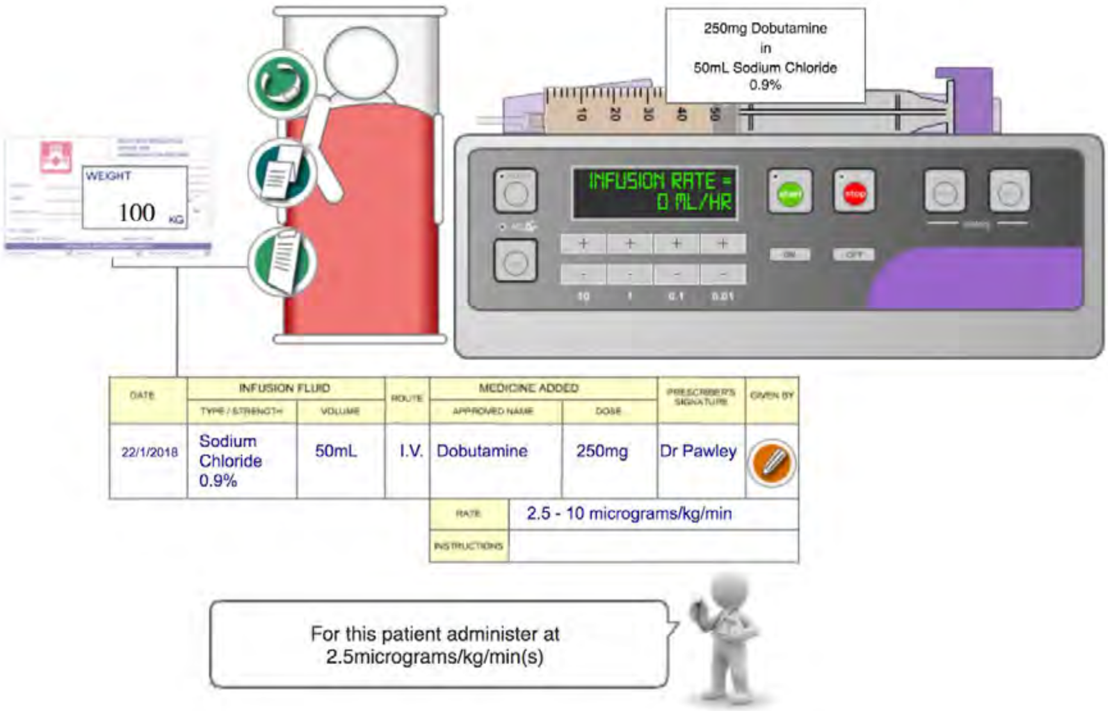

Figure 8: Screenshot of example problem from safeMedicate®, ©Authentic World Ltd.

[3] https://www.safemedicate.com/

The problem describes the task of setting up a syringe driver to deliver the described medication. However, before the user can actually perform the set-up of the syringe driver there are a number of steps they must complete first, such as checking all of the details to ensure that the patient, prescription, medication, etc., are correct and match with each other. Figure 9 shows a partial task model for these initial steps (note that some of these can be decomposed further if required).

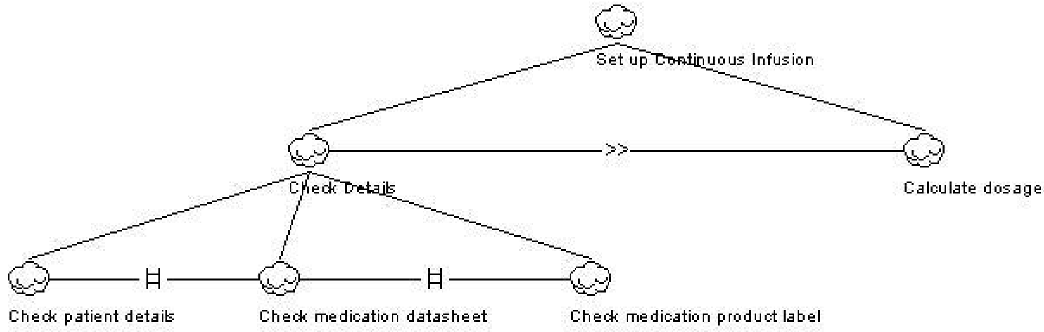

Figure 9: Task model for initial steps to set up infusion

Figure 10 shows the task decomposition for the 'Calculate dosage' task. Again, this is based on the problem example from safeMedicate® and Figure 11 shows the model answer for completing this correctly, which forms the basis for the task model.

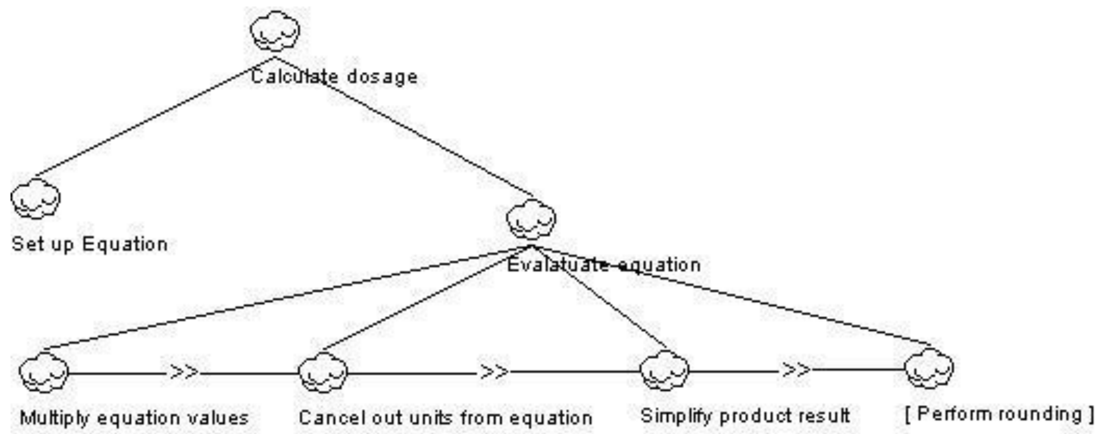

Figure 10: Task model for calculating dosage

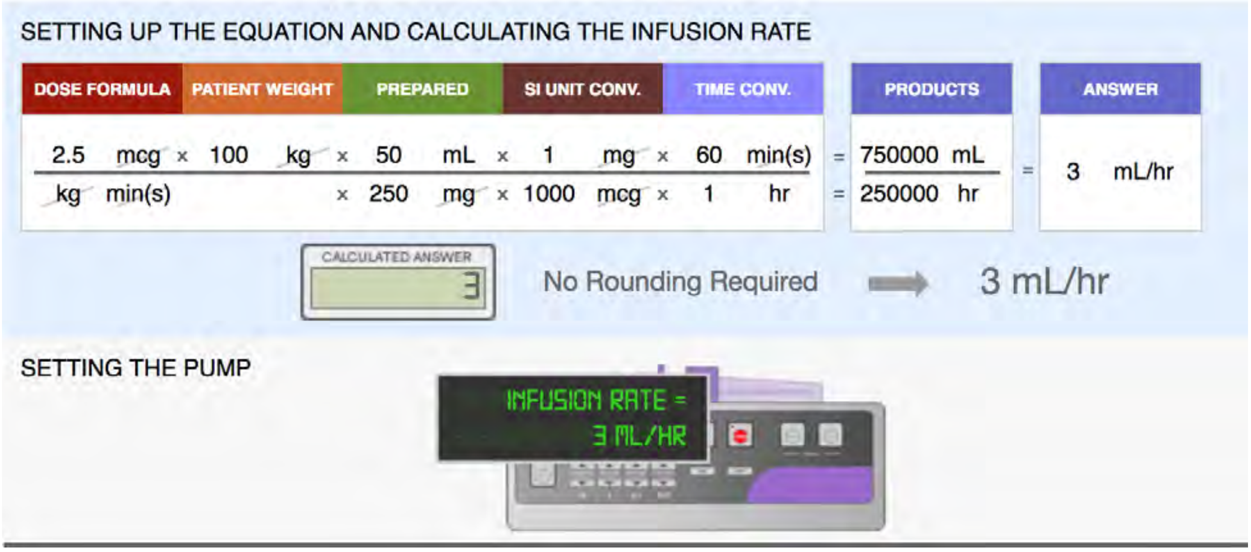

Figure 11: Model answer screenshot from safeMedicate®, ©Authentic World Ltd.

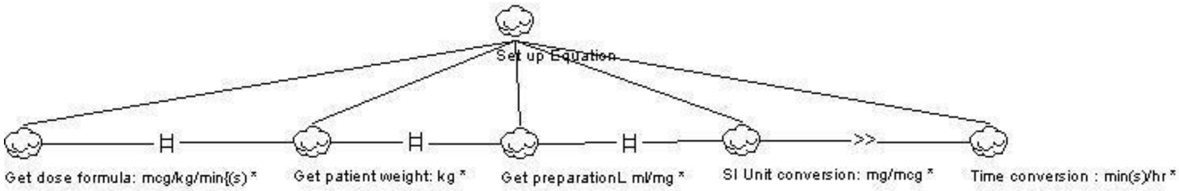

Figure 12: Set up equation task model for specified task

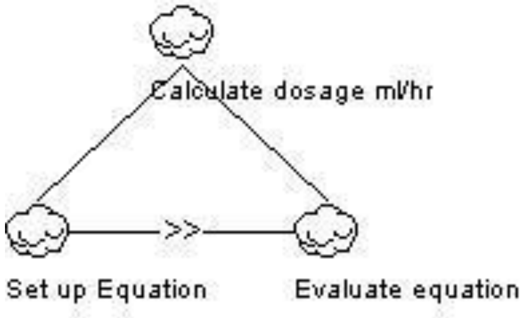

Figure 13: Task model for calculating dosage for given example

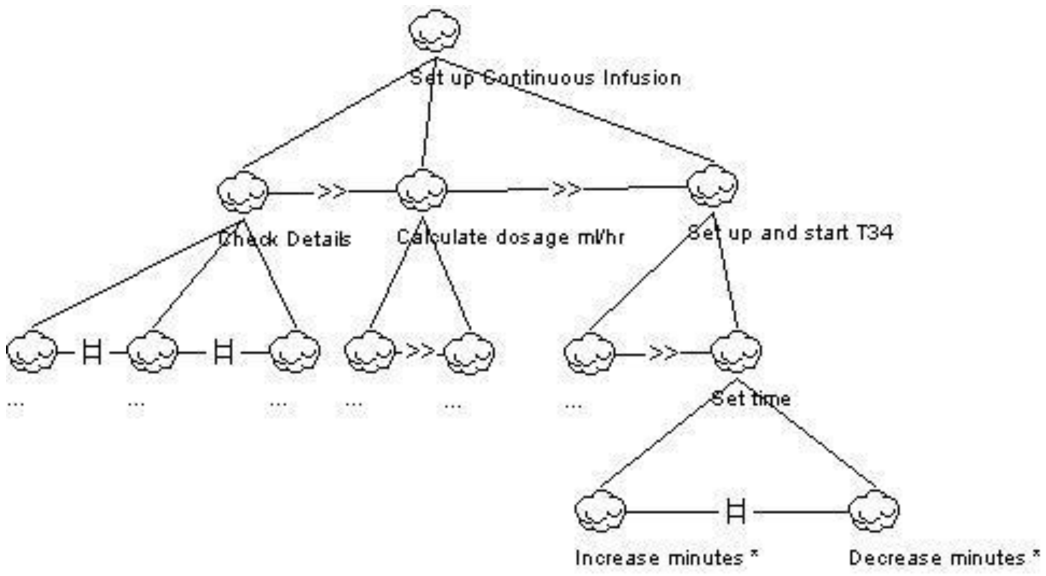

Figure 14: Combined full specialised task model

The activities described in both the task models based on the numeracy example (Figures 9 and 10) and those from the device models in the previous section (Figures 4 and 5) link directly to the numbered steps within the competence model section as follows:

- (1) is required to check the information and links to the steps under the "Check Details" subtask
- (1) is required to set up the equation with the correct values and calculation steps
- (2) is required to evaluate the equation
- (3) is required to set up and start infusion

We can, therefore, identify a relationship between the two domains, where fulfilling tasks described in device models depends on competencies outlined in the MDC-PS. The actual relationship depends on the level of analysis (how far we decompose the sub-tasks into smaller steps) but we can see how it starts to become possible to identify errors or mismatches in beliefs, behaviours and requirements as we start to bring the two domains together via the models. We discuss this further next.

Combining and comparing task models

Now that we have described how to generate task models from both interactive system models and numeracy education tasks, we can consider how these might be used together. In Figure 1 in our introduction, we state that we want to use the comparisons of these models to: inform device design; inform user education; identify potential errors. In addition, our longer term goals are to consider how mappings between models, as well as comparisons, can be informative. Here we begin this process by discussing how we might compare and combine the two sets of models in a useful manner, i.e., one that is productive of safe, efficient and effective clinical practice.

The task models of Figures 9 and 10 represent the task described in the safeMedicate® example of Figure 8. We can, of course, decompose these further into specific (or specialised) steps for the actual example, in much the same way as we considered Figure 4 as a generic model that could be specialised for a specific medical device (the T34) of Figure 6. If we expand the "Set up Equation" subtask with these specific details we may generate a task model such as that of Figure 12.

We can do the same for the "Evaluate equation" model by adding the relevant detail from the example task, although we omit the actual model here for brevity. The higher level model of Figure 10 is now specialised, as shown in Figure 13.

In order to build a complete task model to include the setting up of the infusion device (i.e., completing a specialised version of Figure 9 we can do one of three things:

- Build the sub-task model from the problem example by inferring the steps required (using the sample device picture included in Figure 8).
- Use the generic “Set up” model of Figure 4 and add this as a sub-task to “Set up Continuous Infusion” which is dependent on the completion of the “Check Details” and “Calculate Dosage” tasks.
- Most usefully, we can use the specialised “Set up” model for whichever actual device is being used to deliver the medication (which we assume here for convenience is the T34 syringe driver) as the sub-task. If we do so and create the model shown in part in Figure 14 (where the ... replace omitted sub-tasks) we are immediately able to identify a mismatch which could lead to potential user error.

The “Calculate dosage task” is defined in terms of millilitres per hour (ml/hr), but the “Set time” task is defined in terms of minutes. If the user is unaware of this difference they have the potential to enter the calculated value without first converting to the units of the medical device. With this potential problem identified we could use the information to ensure that such a possibility is highlighted within the numeracy education and device training, i.e., the knowledge identifies a potential error and we use it to inform user education.

Conversely, we could use this information to inform device design. If we were using the combined task models during the development process we could identify the mismatch that occurs and design the device such that it enables the user to enter the values as calculated but also define the units (ml/hr) of their values. The device could then perform the conversion to whichever values it requires for its own delivery calculation. Even better, if we are using the models in initial design and prototyping processes it might suggest that the device itself could be used to evaluate the equation with the user entering all of the values and units from the equation set up directly into the device. This has the added advantage of removing any additional calculation device (such as a pocket calculator or phone calculator) which might be used to evaluate the equation and which has the potential to introduce a whole new set of errors (see Thimbleby, 2000; Thimbleby, 2015 for a full discussion of this problem).

Discussion

In the introduction we introduced our wider research goals with the following questions:

- How can we use formal models of medical devices in conjunction with task models to improve the design and development process?
- How can we use task models to identify mismatches in user knowledge and education with device usage?
- How can we use task models of medication calculations and technical competence to inform user education and the design of medical devices?
- How can we use comparisons of task models of devices and medication delivery to identify potential use errors?

We now discuss how these have been addressed in this paper.

We began by describing how generic task models can be used as inputs into formal models of interactive systems through their incorporation into informal design artefacts. While the use of task models in UCD is typical, we gain additional benefits from the use of formal models which describe the informal design artefacts as this ensures the informal inputs are captured within such models. By then deriving interaction sequences from the formal models we have shown how such model-specific task models can be used to improve designs and prototypes. This satisfies Step 1 in our overview diagram of Figure 1 and answers the first question above.

Task models of specific devices, such as the Niki T34 syringe driver described above, can be compared to the more generic models which enables the identification of potential mismatches in user expectations and actual behaviour of the device. Medical devices can have a variety of different number entry types (from simple up/down arrows to full numeric keypads). They also exhibit a variety of different behaviours regarding minimum and maximum value input roll-overs (what happens at '0' if user tries to decrease the value) and positional displays which change according to whether or not decimal points exist in displayed values. These are often the cause of number entry errors, and so being able to identify the potential for some of these to occur in the manner suggested enables us to both improve design as well as provide better education by identifying to users that such differences exist. This partially answers our second research question.

We show how task models can be generated from numeracy education examples which represent authentic tasks and then describe how we can combine and compare these with generic, model-dependent and device-dependent models (Steps 2 and 3 in the overview diagram). This enables us to identify potential user errors which can be used to inform both device design and user education and partially answers our third and fourth research questions. We also show how the MDC-PS competence descriptions are related to the generation and use of the task models, which strengthens the relationship between our two domains.

Bringing together the two domains in the manner described has the potential, therefore, to provide benefits to both. The formal models and device models can suggest areas that should be included in the technical competence and numeracy education. Likewise, the task models of the authentic numeracy tasks can indicate improvements or enhancements of the medical devices. Although we have introduced the idea of using such models to support users switching between different types of device (which may have subtle differences in how they are used), we have not elaborated on this here. Similarly we have not discussed how the combined models may be used to consider the use of multiple devices for a single patient. We leave these matters for future work, however, from these initial examples we believe we have demonstrated the applicability of our methods in this area.

Conclusions and future work

In this paper we have laid the groundwork for a closer integration between software and device models used to improve design and use of safety critical medical systems with numeracy education in and for the clinical context.

We have shown how task models of user goals (which we call generic task models) can be used as inputs to formal models of interactive systems, such as medical devices. We have also shown how task models can be derived from such formal models (model-dependent task models) and also specific devices (device-dependent task models). Task models used in this way can be used to improve design and identify and mitigate the effects of potential user errors.

We have also shown how task models can be created from numeracy education examples which describe knowledge-based tasks for medication delivery to ensure nursing staff have the necessary skills to perform medication dosage calculations and administer medication. The task models can then be used in conjunction with model-dependent and device-dependent models to inform user education and identify potential user errors.

Our main contributions here are demonstrating that by expressing properties of two different domains (interactive system modelling and numeracy education) in a common language - task models - we are able to compare and integrate models. This allows us to partially answer all of the research questions posed in our introduction. It also provides the basis for our further research which will enable us to fully answer all of these questions.

In order to extend this work further we can now begin to consider deriving algorithms for traversing the task models in order to automate the process of combining and comparing the models. This may also require some ontological mapping to resolve naming differences and automate the understanding needed to identify types - such as units of time and measurement, etc.. We also wish to investigate further how different types of number entry (five key interfaces *vs*. numeric keypads, for example) may lend themselves to calculation competence better than others and whether this can be identified from the task model comparisons.

We believe task models may provide the solution to the problems outlined above since they can provide a link between the interactive system design and the numeracy demands, practices and associated education and training. Tasks are fundamental to both in terms of ensuring devices are designed to correctly support tasks in a usable fashion and ensuring healthcare personnel can complete specific tasks relating to medication calculations. In particular: formal models and device models may suggest areas that should be included in numeracy education, especially for the development of technical competence; task models of authentic numeracy tasks may suggest improvements or enhancements of medical devices.

References

Barboni, E., Ladry, J.-F., Navarre, D., Palanque, P. A., & Winckler, M. (2010). Beyond modelling: An integrated environment supporting co-execution of tasks and systems models. In *EICS '10 Proceedings of the 2nd ACM SIGCHI symposium on Engineering interactive computing systems. Berlin, Germany — June 19 - 23, 2010* (pp. 165–174). New York, NY, USA: ACM.

Berndt, E., Furniss, D., & Blandford, A. (2015). Learning contextual inquiry and distributed cognition: A case study on technology use in anaesthesia. *Cognition, Technology & Work, 17*(3), 431–449.

Blandford, A., Buchanan, G., Curzon, P., Furniss, D., & Thimbleby, H. (2010). Who's looking? Invisible problems with interactive medical devices. In G. R. Hayes, D. S. Tan, & L. Wilcox (Eds.), *Proceedings of the First International Workshop on Interactive Systems in Healthcare ACM Special Interest Group on Computer-Human Interaction* (pp. 9–12). Irvine, CA: University of California, Irvine.

Blandford, A., Butterworth, R., & Good, J. (1997). Users as rational interacting agents: Formalising assumptions about cognition and interaction. In M. D. Harrison & J. C. Torres (Eds.), *Design, Specification and Verification of Interactive Systems '97. Eurographics* (pp. 45–60). Vienna: Springer.

Blandford, A., De Pietro, G., Gallo, L., Gimblett, A., Oladimeji, P., & Thimbleby, H. (2011). Engineering interactive computer systems for medicine and healthcare (EICS4Med). *EICS*, 341–342.

Blandford, A., & Furniss, D. (2005). DiCoT: A methodology for applying distributed cognition to the design of teamworking systems. In S. W. Gilroy & M. D. Harrison (Eds.), *Interactive Systems. Design, Specification, and Verification. DSV-IS 2005. Lecture Notes in Computer Science* (Vol. 3941, pp. 26–38). Berlin, Heidelberg Springer.

Bodkin, H. (2018, 24 June 2018). Hunt orders probe into faulty opioid syringe pumps amid allegations thousands may have died. *The Telegraph*. Retrieved from https://www.telegraph.co.uk/news/2018/06/24/hunt-orders-probe-faulty-opioid-syringe-pumps-amid-allegations/

Bowen, J. (2015). Creating models of interactive systems with the support of lightweight reverse-engineering tools. In *Proceedings of the 7th ACM SIGCHI Symposium on Engineering Interactive Computing Systems, EICS 2015, Duisburg, Germany, June 23-26, 2015* (pp. 110–119). Duisburg, Germany.

Bowen, J., & Reeves, S. (2006). Formal models for informal GUI designs. In *1st International Workshop on Formal Methods for Interactive Systems, Macau SAR China, 31 October 2006* (Vol. 183, pp. 57–72). Amsterdam, NL: Elsevier.

Bowen, J., & Reeves, S. (2007). Refinement for user interface designs. In P. Curzon & A. Cerone (Eds.), *Proceedings of the 2nd International Workshop on Formal Methods for Interactive Systems (FMIS 2007). Lancaster University, UK* (Vol. 208, pp. 5–22). Amsterdam, NL: Elsevier.

Bowen, J., & Reeves, S. (2008). Formal models for user interface design artefacts. *Innovations in Systems and Software Engineering, 4*(2), 125–141.

Bowen, J., & Reeves, S. (2013). Modelling safety properties of interactive medical systems. In *Proceedings of the 5th ACM SIGCHI Symposium on Engineering Interactive Computing Systems (EICS '13)* (pp. 91–100). New York, NY: ACM.

Bowen, J., & Reeves, S. (2017). Combining models for interactive system modelling. In B. Weyers, J. Bowen, A. Dix, & P. Palanque (Eds.), *Handbook of Formal Methods in Human-Computer Interaction* (pp. 161–182). New York, NY: Springer International Publishing.

Calvary, G., Coutaz, J., & Thevenin, D. (2001). Supporting context changes for plastic user interfaces: A process and a mechanism. In A. Blandford, J. Vanderdonckt, & P. Gray (Eds.), *Joint Proceedings of HCI'2001 and IHM'2001* (pp. 349–363): Springer-Verlag.

Campos, J., & Harrison, M. (2011). Modelling and analysing the interactive behaviour of an infusion pump. *Electronic Communications of the EASST. Proceedings of the Fourth International Workshop on Formal Methods for Interactive Systems (FMIS 2011), 45*, 1-16. Retrieved from https://haslab.uminho.pt/sites/default/files/jccampos/files/641-1972-1-pb.pdf

Card, S. K., Moran, T. P., & Newell, A. (1980). Keystroke-level model for user performance time with interactive systems. *Communications of the ACM, 23*(7), 396–410. doi:http://dx.doi.org/10.1145/358886.358895

Card, S. K., Moran, T. P., & Newell, A. (1983). *Psychology of Human Computer Interaction*. Mahwah, NJ: Lawrence Erlbaum Associates.

Coben, D., & Weeks, K. W. (2014). Meeting the mathematical demands of the safety-critical workplace: Medication dosage calculation problem-solving for nursing. *Educational Studies in Mathematics. Special Issue on Vocational Education and Workplace Training, 86*(2), 253-270. doi:10.1007/s10649-014-9537-3

Dittmar, A., & Forbrig, P. (2003). Higher-order Task Models. In J. A. Jorge, N. Jardim Nunes, & J. Falcão e Cunha (Eds.), *Interactive Systems. Design, Specification, and Verification. DSV-IS 2003. Lecture Notes in Computer Science* (Vol. 2844, pp. 187–202). Berlin, Heidelberg: Springer.

European Communities Education and Culture. (2008). *The European Qualifications Framework for Lifelong Learning (EQF)*. Retrieved from Luxembourg: http://www.bmwf.gv.at/fileadmin/user_upload/.../nqr/EQF_EK_EN_2008.pdf

FDA. (2017, 13 December 2017). Examples of Reported Infusion Pump Problems. Retrieved from https://www.fda.gov/medicaldevices/productsandmedicalprocedures/generalhospitaldevicesandsupplies/infusionpumps/ucm202496.htm

Galbraith, P. L., Henn, H.-W., & Niss, M. (Eds.). (2007). *Modelling and Applications in Mathematics Education. The 14th ICMI Study*. Cham, Heidelberg, New York, Dordrecht, London: Springer.

Geiger, V., & Frejd, P. (2015). A reflection on mathematical modelling and applications as a field of research: Theoretical orientation and diversity. In G. Stillman, W. Blum, & M. S. Biembengut (Eds.), *Mathematical Modelling in Education Research and Practice: Cultural, social and cognitive influences* (pp. 161–171). Cham: Springer.

Gottlieb, L. N. (2012). Nurses: The first and last line of defence for not only patients but physicians as well. *Canadian Journal of Nursing Research (CJNR), 44*(2), 3-6.

Harrison, M. D., Masci, P., Campos, J. C., & Curzon, P. (2017). Verification of user interface software: The example of use-related safety requirements and programmable medical devices. *IEEE Trans. Human-Machine Systems, 47*(6), 834–846. doi:http://dx.doi.org/10.1109/THMS.2017.2717910

Johnson, C. W. (2011). An introduction to human error, interaction and the development of safety-critical systems. In G. A. Boy (Ed.), *A Handbook of Human-Machine Interaction: A human-centred design approach*. Ashgate Publishing Ltd: Abingdon, Oxon.

Kieran, C., Doorman, M., & Ohtani, M. (2015). Frameworks and principles for task design. In A. Watson & M. Ohtani (Eds.), *Task Design In Mathematics Education: An ICMI Study 22* (pp. 19-81). Cham, Switzerland: Springer International Publishing.

Kohn, L. T., Corrigan, J. M., & Donaldson, M. S. (Eds.). (2000). *To Err is Human: Building a safer health system*. Washington, DC: National Academy Press.

Martinie, C., Palanque, P. A., Navarre, D., Winckler, M., & Poupart, E. (2011). Model-based training: An approach supporting operability of critical interactive systems. In *EICS '11 Proceedings of the 3rd ACM SIGCHI symposium on Engineering interactive computing systems* (pp. 53–62). New York, NY: ACM.

Martinie, C., Palanque, P. A., Ragosta, M., & Fahssi, R. (2013). *Extending procedural task models by systematic explicit integration of objects, knowledge and information. Article No. 23*. Paper presented at the ECCE '13 Proceedings of the 31st European Conference on Cognitive Ergonomics, Toulouse, France.

Mori, G., Paternò, F., & Santoro, C. (2002). CTTE: Support for developing and analyzing Task Models for Interactive System Design. *IEEE Transactions on Software Engineering, 28*(8), 797–813. doi:https://doi.org/10.1109/TSE.2002.1027801

Niss, M., Blum, W., & Galbraith, P. (2007). Introduction. In W. Blum, P. Galbraith, H.-W. Henn, & M. Niss (Eds.), *Modelling and Applications in Mathematics Education* (pp. 3–32). New York, NY: Springer.

Palanque, P. A., Ladry, J.-F., Navarre, D., & Barboni, E. (2009). High-fidelity prototyping of interactive systems can be formal too. In J. A. Jacko (Ed.), *Proceedings of the 13th International Conference on Human-Computer Interaction. Part I: New Trends* (Vol. 5610, pp. 667–676). Berlin, Heidelberg: Springer-Verlag.

Palm, T. (2009). Theory of authentic task situations. In L. Verschaffel, B. Greer, W. van Dooren, & S. Mukhopadhyay (Eds.), *Words and Worlds. Modelling verbal descriptions of situations* (pp. 3-19). Rotterdam: Sense Publishers.

Paternò, F. (2001). Task Models in interactive software systems. In S. K. Chang (Ed.), *Handbook of Software Engineering and Knowledge Engineering* (pp. 817–836). London: World Scientific Publishing Company.

Paternò, F., Mancini, C., & Meniconi, S. (1997). ConcurTaskTrees: A diagrammatic notation for specifying task models. In S. Howard, J. Hammond, & G. Lindgaard (Eds.), *International Conference on Human-Computer Interaction, INTERACT '97, IFIP TC13 14th-18th July 1997, Sydney, Australia* (pp. 362–369). London: Chapman & Hall.

Paternò, F., Santoro, C., & Spano, L. D. (nd). ConcurTaskTrees. Retrieved from https://www.w3.org/wiki/images/6/62/W3c-ctt.pdf

Rajkomar, A., & Blandford, A. (2012). Understanding infusion administration in the ICU through Distributed Cognition. *Journal of Biomedical Informatics, 45*(3), 580–590.

Rukšėnas, R., Back, J., Curzon, P., & Blandford, A. (2008). Formal modelling of salience and cognitive load. *Electronic Notes in Theoretical Computer Science, 208*, 57–75. Retrieved from https://core.ac.uk/download/pdf/1685658.pdf

Shepherd, A. (1989). *Analysis and Training in Information and Technology Tasks*. Hemel Hempstead, Herts.: Ellis Horwood.

Stillman, G. (2019). State of the art on modelling in mathematics education—Lines of inquiry. In G. Stillman & J. P. Brown (Eds.), *Lines of Inquiry in Mathematical Modelling Research in Education* (pp. 1-19). Cham, Switzerland: Springer.

Thimbleby, H. (2000). Calculators are needlessly bad. *Int. J. Hum.-Comput. Stud, 52*(6), 1031–1069. doi:http://dx.doi.org/10.1006/ijhc.1999.0341

Thimbleby, H. (2015). Safer user interfaces: A case study in improving number entry. *IEEE Transactions on Software Engineering, 41*(7), 711–729. Retrieved from http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=6991548

Turner, J., Bowen, J., & Reeves, S. (2017). Supporting interactive system testing with interaction sequences. In *EICS '17 Proceedings of the ACM SIGCHI Symposium on Engineering Interactive Computing Systems* (pp. 129–132). New York, NY: ACM.

Vipond, S. (2016). Pioneering medical device training in the Digital Age. *Learning Solutions Magazine, 2027*.

Watson, A., & Ohtani, M. (Eds.). (2015). *Task Design In Mathematics Education: An ICMI Study 22*. Cham, Switzerland: Springer International Publishing.

Weeks, K. W., Hutton, B. M., Young, S., Coben, D., Clochesy, J. M., & Pontin, D. (2013). Safety in Numbers 2: Competency modelling and diagnostic error assessment in medication dosage calculation problem-solving. *Nurse Education in Practice, 13*(2), e23-32. doi:http://dx.doi.org/10.1016/j.nepr.2012.10.013