Electronic Communications of the EASST
Volume 45 (2011)

Proceedings of the
Fourth International Workshop on Formal Methods
for Interactive Systems
(FMIS 2011)

Modelling and analysing the interactive behaviour of an infusion pump

José Creissac Campos, Michael D. Harrison

16 pages

Guest Editors: Judy Bowen, Steve Reeves
Managing Editors: Tiziana Margaria, Julia Padberg, Gabriele Taentzer
ECEASST Home Page: http://www.easst.org/eceasst/ ISSN 1863-2122

Modelling and analysing the interactive behaviour of an infusion pump

José Creissac Campos[1], **Michael D. Harrison**[2]

[1] jose.campos@di.uminho.pt Departamento de Informática/CCTC, Universidade do Minho, Braga, Portugal

[2] michael.harrison@ncl.ac.uk School of Computing Science, Newcastle University, Claremont Tower Newcastle upon Tyne, NE1 7RU, UK

Abstract: This paper is concerned with the scaleable and systematic analysis of interactive systems. The motivating problem is the procurement of medical devices. In such situations several different manufacturers offer solutions that support a particular clinical activity. Apart from cost, which is a dominating factor, the variations between devices are relatively subtle and the consequences of particular design features are not clear from manufacturers' manuals, demonstrations or trial uses. Despite their subtlety these differences can be important to the safety and usability of the device. The paper argues that formal analysis of the range of offered devices can provide a systematic means of comparison. The paper also explores barriers to the use of such techniques, demonstrating how layers of specification may be used to make it possible to reuse common specification. Infusion pumps provide a motivating example. A specific model is described and analysed and comparison between competitive devices is discussed.

Keywords: MAL, IVY, medical devices, procurement, interactive systems

1 Introduction

The systematic analysis of properties of interactive behaviour using Modal Action Logic (MAL) and the IVY tool has been reported in previous papers [CH08, CH09]. This paper focuses on the modelling and analysis of a more complex device than described previously, supporting a variety of display modes. It is part of the ongoing analysis of a range of medical infusion devices with the aim of reducing procurement costs. The situation that motivates this discussion is where comparable analyses are required for a range of devices to assess which design is most appropriate for purchase given a particular context. The focus of this comparison is to identify issues which might affect *their use* in the proposed work context. Properties of the user interface provide the primary concern. A recently developed intravenous infusion pump (the Alaris GP [Car06]) is considered as a candidate for analysis. The relevance of the rigorous assessments proposed here prior to procurement are heightened by well documented concerns about safety of infusion devices in general (see for example [FA10]).

The model that is used in the analysis is based on two sources: a manual [Car06] and a simulation developed by Patrick Oladimeji at Swansea University http://cs.swan.ac.uk/~cspo/simulations based on the actual device. In addition to discussing a proposed means of product comparison, the paper demonstrates that MAL and IVY can be scaled appropriately to a suitable class of devices. The task of producing the model for the candidate device is illustrated by providing and explaining MAL fragments. Inevitably this means that, because the model is not presented completely, it can appear complex. However the benefit of this level of explanation is to indicate the scale of the task of producing a detailed model of a real device.

The paper makes two distinct contributions. It proposes a layered approach to modelling systems with the aim of supporting reuse, thereby reducing modelling costs. It demonstrates the modelling and analysis of a more compex system than previously discussed using the IVY tools which includes the specification of both soft buttons and screen areas where information is mode dependent. It proposes an approach to dealing with this complexity. The paper briefly introduces the model's specification and focuses on properties that can be used for the purposes of comparison. After a brief review of the background (Section 2) and introduction to the device (Section 3) the analysis is introduced and illustrated. Section 4 introduces the three layers of model and Section 5 provides examples of the verification. A final section (Section 6) provides some discussion and conclusions.

2 Background

The procurement process is an important stage in the introduction of new technology within healthcare. Techniques are required that will make it possible to explore candidates systematically (for discussion, see [Gin05]). A prime motivation for this analysis has been a concern with the safety of medical devices, particularly from an ergonomic perspective [MNMC08]. A number of usability engineering methods have been explored in this regard, for example Cognitive Walkthrough [BO07] and Heuristic Evaluation [ZJP$^+$03]. These techniques have been adapted to be relevant to the requirements of the medical domain. They involve the imaginative application of guidelines or heuristics either to a description of the device, in the case of heuristic evaluation, or to how the device is to be used (in terms of a task or a scenario) in the case of cognitive walkthrough. These desk-based evaluation techniques are either applied by individuals or suitably qualified teams. Human reliability analysis techniques [Kir94] have also been applied in a medical domain, for example SHERPA [LSH06], with the aim of providing realistic assessments of the safety of use of medical devices. A useful review of techniques can be found in [MNMC08]. All these techniques are systematic, they often involve teams of evaluators with a balance of skills, and require a representation of the activities that are being carried out in terms of a task or a scenario. The device model is often informal, for example the proposal in [BO07] is that the task model be annotated with screen shots to indicate how the device functions.

Formal analysis of software based systems provides the underpinning to the work described here. These techniques provide the opportunity to be both thorough and systematic by checking a battery of properties. Relevant techniques include those of Thimbleby and Gow [TG08] who have applied formal approaches to medical devices based on graph models of the interface. Thimbleby and Oladimeji use similar techniques to explore the data entry characteristics of the infusion pump [TO09]. Their process involves the manual development of a state model reflecting the design. These models are analysed as graphs. Animation and prototyping of models can also be a powerful method of visualising the consequences of a design. Palanque and others have

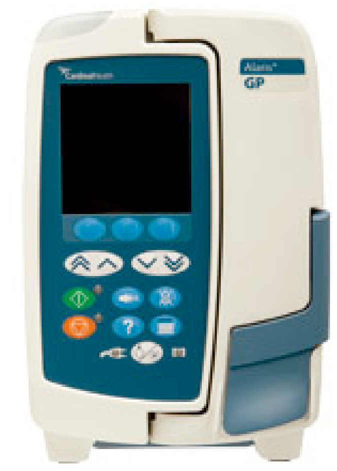


Figure 1: The Alaris GP Volumetric Pump

developed Petri net and object oriented models of interactive devices and use a tool (Petshop) [NPDB06] to animate and prototype. Their work focuses at the widget level and has included a specification of the ARINC 661 aircraft cockpit controls standard [BCNP07]. "Visual Event Grammars" (VEG) [BRRP05] have also been used to describe interface widgets. These widgets are described as a set of production rules and are translated into Promela and analysed using SPIN [Hol03]. VEG is used to explore reachability and deadlock properties but does not explore usability related properties explicitly.

An important issue in the development of such models is to deal with issues of scale, particularly dealing with the number of states that must be explored. In this respect Bolton and Bass use SAL to explore an architecture for a Baxter iPump [BB10], another variant of the type of device explored here. Their paper focuses on the architecture of their model and the measures that were taken to ensure the model was tractable. Using IVY (which maps to SMV) it was possible to explore substantially richer models of a similar device.

3 The infusion pump

The candidate device for procurement is an infusion pump. The particular focus for this discussion is one variant of the Alaris GP infusion pump [Car06] (see Figure 1). This device is safety critical, and is used in hospital settings to provide accurate and reliable intravenous infusions to patients. Failure to set up infusion pumps correctly can have severe consequences. A number of documented incidents with a range of types of infusion pump indicate how the wrong material, or the wrong volume at the wrong rate, has had disastrous consequences for the patient to whom the infusion was being administered (see, for example [ZJP^{+}03]). The manual for the particular

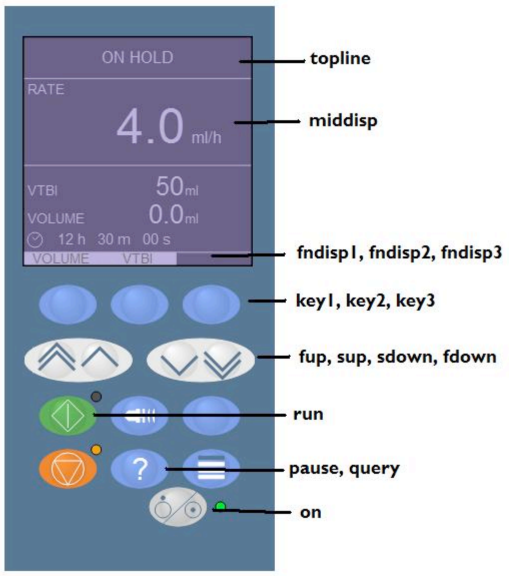


Figure 2: Alaris actions

device described in this paper [Car06] proposes that it is designed to be used in a variety of settings, from general hospital wards, critical and intensive care, operating rooms to accident and emergency rooms. The infusion pump allows input of data relating to the volume to be infused and rate of infusion prior to starting the infusion process.

The pump has two basic states: infusing and holding. In the infusing state the volume to be infused (vtbi) is pumped into the patient intravenously according to the infusion rate. While in the infusing state the vtbi can be exhausted, in which case the pump continues in KVO (Keep Vein Open) mode and sets off an alarm. The other basic state is holding. In this state values and settings can be changed via chevron buttons (for the device layout, see Figure 2). These buttons are used to increase or decrease entered numbers incrementally. Depending on mode the buttons can be used to change infusion rate and volume to be infused, or alternatively allow the user to move between options in a menu, for example in bag mode and in query mode. Bag mode allows the user to select from a set of infusion bag options, thereby setting vtbi to a predetermined value. Query mode, which is invoked by pressing the query button, generates a menu of options. These options include a number of possibilities according to how the device is configured by the manufacturer, including the possibility of locking the infusion rate, or disabling the possibility of locking it. The device also allows movement between display modes via three function keys (*key1*, *key2* and *key3*). Each function key has a display associated with it indicating its present function. The focus of the analysis presented here is to consider confusions that arise as a result of the mode structure exhibited by this device. The aim is to compare it with the structure of other similar devices. Furthermore the analysis explores potential confusions that arise through the information that is being displayed.

The focus of the paper is on the “software” aspects of the design. The relationship between the device and its environment, which is of course extremely important, will not be considered explicitly in this paper. Hence, for example physical aspects of the "giving set" used to connect the device to the vein of the patient will not be considered nor will any issues associated with electromagnetic interference. These other aspects are of course also important in the procurement of these devices.

There are a wide range of infusion pumps and syringe drivers in use in hospitals with a variety of interface styles that differ in terms of style of data entry and the ways in which modes are structured. This specific example will trigger discussion of broader issues.

4 The model

The full specification of the model discussed in this paper can be seen at the Minho HCI specification repository (http://hcispecs.di.uminho.pt). Three layers of the model are described to make it easier to reveal the differences between alternative candidate pump designs. The underlying pump model (the inner layer) is identical, or nearly identical, subject to minor controlled variations across the range of devices. The middle layer is specific to the device being modelled and describes its interface structure. The activity model (the outer layer) will assume the same activities in each device that is being considered. These activities will be derived from an understanding of the work that the clinician is to carry out. This outer layer is not dependent on characteristics of the device though it provides a mapping into the middle layer thereby grounding it in the specific details of the device. The paper introduces the MAL notation in which the model is described through a number of fragments. There is no space to provide a full description of the notation. More details of MAL can be found in [CH01, CH08, CH09].

4.1 The inner layer

The inner layer describes the temporal evolution of the infusion process. The process is captured in an invariant:

$$\begin{aligned}
&\mathit{infusionrate} > 0 \, -\!> \mathit{infusionrateaux} = \mathit{infusionrate}\\
&\mathit{infusionrate} > 0 \, -\!> \mathit{time} = (\mathit{vtbi}/\mathit{infusionrateaux})\\
&\mathit{infusionrate} = 0 \, -\!> \mathit{time} = 0
\end{aligned}$$

This invariant asserts relationship between vtbi, infusion rate and time to completion of the process. The invariant ensures that division by zero is not possible by using an additional state attribute *infusionrateaux* taking values in the range $1..\mathit{maxrate}$. The tick action (which captures the evolution of the process) describes infusion and the alarms that occur when the device has been left in a hold state for too long. To illustrate the model's specification in the simplest terms one element of the infusion process is described using a modal axiom in MAL.

$$\begin{aligned}
&(\mathit{infusionstatus} = \mathit{infuse}) \;\&\; (\mathit{infusionrate} < \mathit{vtbi}) \, -\!> [\mathit{tick}]\\
&\qquad \mathit{vtbi}' = \mathit{vtbi} - \mathit{infusionrate}\\
&\qquad \& \; \mathit{volumeinfused}' = \mathit{volumeinfused} + \mathit{infusionrate}\\
&\qquad \& \; \mathit{keep}(\mathit{infusionrate}, \ldots
\end{aligned}$$

This axiom describes *tick* as the pump is infusing ($infusionstatus = infuse$) when the infusion rate (i.e. the volume infused per tick) does not lead to exhaustion of the volume to be infused. The axiom has three elements: the action that is being described (contained in square brackets); the conditions that must be satisfied for the action to have the stated effect (left side of the implication); the result of the action under these conditions. $vtbi' = vtbi - infusionrate$ specifies that the next state (indicated by the prime symbol) of *vtbi* must be equal to its old value minus the *infusionrate*. MAL specifies that unless a state attribute is explicitly constrained in a modal axiom then it can change randomly in the next state. The *keep* function determines the list of state attributes that cannot change.

4.2 The middle layer

The difference between the competing pumps is captured in the middle layer of the specification. The pumps use modes so that the devices' limited display space can be used effectively to help the clinician see what the effects of actions will be. This layer describes the mode structure and the information that is displayed at each stage. The visibility of these variables is specified explicitly using the boolean array *middisp*. The following illustrative axiom describes the behaviour of the soft key 2 when the top line of the device shows "holding" (see Figure 2) and the pump is processing normally (vtbi has not been exhausted i.e. the pump is not in KVO mode).

$$
\begin{aligned}
&(topline = holding)\ \&\ !kvoflag\ {->}\ [key2]\\
&\quad topline' = dispvtbi\ \&\ oldvtbi' = vtbi\ \&\\
&\quad middisp[dvtbi]'\ \&\ !middisp[dvol]'\ \&\ !middisp[dtime]'\ \&\\
&\quad !middisp[dbags]'\ \&\ !middisp[dkvorate]'\ \&\ !middisp[dquery]'\ \&\\
&\quad fndisp1' = fok\ \&\ fndisp2' = fbags\ \&\ fndisp3' = fquit\ \&\\
&\quad vtmode'\ \&\ !rmode'\ \&\ !bagmode'\ \&\ !qmode'\ \&\ elapse' = 0\ \&\\
&\quad keep(infusionstatus, infusionrate, volumeinfused, vtbi,\\
&\qquad\qquad onlight, runlight, pauselight, rdisabled, rlock, kvoflag)
\end{aligned}
$$

The axiom asserts that the next top line shows "volume to be infused" (*dispvtbi*) and the value of vtbi is displayed (i.e. $middisp[dvtbi]'$ is true). The soft function keys ($fndisp1'$ etc.) in this next step show "ok", "bags" and "quit" respectively. The mode in this step is determined to be *vtmode* while other modes are false.

4.3 The outer layer

The third layer of the model describes the activities that are assumed to be typical of the clinician's use of the device. These activities are assumed to be determined through negotiation with domain and human factors experts. The examples here are intended to be indicative only. One of the activities that may have been determined as being important in the infusion process is that another clinician should confirm that the correct value of vtbi has been entered into the device based on the original prescription. This layer of the model describes activities, capturing what needs to be done in relation to the pump. The process of eliciting and understanding these activity actions and meta-state attributes is a process akin to task modelling [KA92] where the human factors or domain expert observes the infusion activity in context. These meta-state attributes do not capture any feature of the device or its interface, rather they indicate a state in the activity as assumed to be understood by the clinician who is using the device. They can be used to constrain the device actions described in the middle layer. Hence confirming that the vtbi has been entered involves completing the *entervtbi* activity (by requiring that the meta-state attribute *enteringvtbi* is set to false) and asserting that vtbi has been confirmed (using the *vtbiconfirmed* meta-state attribute).

$$[confirmvtbi]\ !enteringvtbi'\ \&\ vtbiconfirmed'\ \& \\ \quad keep(enteringrate, readyforrate, mrate, mvolume, rateconfirmed, readyforvtbi)$$

The *confirmvtbi* activity is only permitted if *enteringvtbi* is true and and the value of vtbi is equal to the required volume contained in the prescription. Note that the attribute *vtbi* defined within the *alarispump* interactor is referred to by prefixing the attribute name with the interactor name *alarispump.vtbi*. Hence

$$per(confirmvtbi) -> (mvolume = alarispump.vtbi)\ \&\ enteringvtbi$$

MAL makes it possible to describe deontic assertions using operators for permission and obligation. In this model permission axioms are used widely to restrict the possible behaviours of actions. Hence the use of a chevron key is permitted only if entering the vtbi or entering the infusion rate as part of an activity defined at the outer layer. These actions are now assumed to bear a specific relation to the prescribed value that the clinician has taken from the prescription. Hence the “fast up” chevron is only used if the current value is less than the required value by more than “big step” in the appropriate activity. This is specified using the following permission axiom.

$$\begin{array}{l} per(alarispump.fup) -> \\ \quad (enteringvtbi\ \&\ alarispump.vtmode\ \&\ ((bigstep + alarispump.vtbi) < mvolume)) \\ \quad | \\ \quad (enteringrate\ \&\ alarispump.rmode\ \&\ ((alarispump.infusionrate + bigstep) < mrate)) \end{array}$$

This outer level has some similarity with the cognitive models described by [RBCB09]. It is however simpler, making no assumptions about the cognitive process itself, simply concerning itself with observed activities.

5 Verifying the model

The aim of the verification process is that similar properties relating to user interface characteristics can be checked of each candidate device. These formal properties provide a benchmark against which each candidate can be explored. As a result, potentially unforeseen design consequences can be discovered that could not be found simply by reading the manual or through brief experiment with the device. The circumstances in which properties fail are assessed with aid of human factors or domain expertise. Thus failure acts as a trigger for the consideration of a human interface characteristic that would otherwise lie hidden. Properties checked of each candidate are of the following types.

- Checking that the process represented in the inner layer is sufficiently visible through the device interface (mirroring the process in the interface).
- Ensuring consistency of use of the display, or of action (consistency of the interface).
- Ensuring that activities described in the outer layer are supported (supporting activities).

The specific details of the properties in these three categories will differ depending on the device that is described in the middle layer of the model. In the case of the device modelled here, the properties that were checked of the inner and middle layers were checked first (reducing the time for checking) before adding a further interactor describing the activity layer.

Properties are presented for analysis using CTL (see [CGP99] for an introduction to model checking). CTL provides two kinds of temporal operator, operators over paths and operators over states. Paths represent the possible future behaviours of the system. When p is a property expressed over paths, $A(p)$ expresses that p holds for all paths and $E(p)$ that p holds for at least one path. Operators are also provided over states. When q and s are properties over states, $G(q)$ expresses that q holds for all the states of the examined path; $F(q)$ that q holds for some states over the examined path; $X(q)$ expresses that q holds for the next state of the examined path; while $[qUs]$ means that q holds until s holds in the path. CTL allows a subset of the possible formulae that might arise from the combination of these operators. $AG(q)$ means that q holds for all the states of all the paths; $AF(q)$ means that q holds for some states in all the paths; $EF(q)$ means that q holds for some states in some paths; $EG(q)$ means that q holds for all states in some paths; $AX(q)$ means that q holds in the next state of all paths; $EX(q)$ means that q might hold in the next state; $A[qUs]$ means that q holds until some other property s holds in all paths; $E[qUs]$ means there exists a path in which q holds until some property s.

The properties are analysed in the IVY tool by translating the MAL model into SMV and using the symbolic model checker NuSMV [CCG+02] to check the CTL properties. It was necessary to restrict the state space by reducing the ranges of vtbi, infusion rate and menu lengths. Checking the combination of inner layer and middle layer of the model using the whole battery of properties took less than twenty minutes on a standard laptop. Adding the outer layer increased the checking time to about 20 minutes per property. Checking the three layer model involved $1.2963e+09(2^{30.2718})$ reachable states.

5.1 Mirroring the device in the interface

Relevant properties first concern the extent to which the infusion pump process is visible in the device. This visibility is concerned with the basic variables: infusion rate, vtbi, time to infuse. It is also concerned with whether it is clear that the pump is infusing or not, including whether it is in KVO mode. The middle layer model simply records whether the attribute is displayed or not. By this means visibility is made explicit in the model. The status of the pumping process is slightly more complex. This status is captured at the inner level using the state attribute *infusionstatus*. In the Alaris pump there is no specific display attribute that indicates unequivocally whether the device is pumping or not. The top line of the display provides information that helps the clinician understand (see Figure 2) what mode the device is in. Appropriate properties associated with checking the interface are:

$$
\begin{aligned}
&AG(topline\ in\ \{infusing, dispkvo, vtbidone\} \leftrightarrow infusionstatus = infuse)\\
&AG(topline\ in\ \{holding, dispvtbi, setvtbi, attention, volume, options, dispinfo, locked, blank\}\\
&\qquad \leftrightarrow infusionstatus = hold)
\end{aligned}
$$

The first property fails. Examining the trace indicates that in KVO mode a display appears that allows function key 1 to take the device to a state where the top line display shows “volume”. This means that “volume” can appear as the top line when both infusing and holding, potentially a cause for concern, particularly if there is no audible alarm associated with the device infusing in KVO mode. On the other hand it could be argued by domain experts that this potential confusion is unlikely to be a real confusion in the context of clinical activities.

In the second property there is a substantial list of displays in the top line that indicate that the device is holding. A further top line display indicates when the infusion rate is locked and appears for one tick if the user attempts to change the infusion rate. The final value in the top line is “blank” used to indicate that the device is switched off regardless of the value of the other state attributes of the model. The second property also fails because “volume” can also appear as the top line when infusing.

5.2 Consistency of the interface

A number of properties are concerned with the consistency of the interface. These include: (1) the role and visibility of modes; (2) consistency of naming and purpose of functions; (3) consistency of behaviour of the data entry keys.

The role and visibility of modes

The device supports modes that govern whether the chevron keys (see Figure 2) can be used to enter vtbi, infusion rate or to move between menu selections in bags mode or options mode. Additional modes include whether the device is in KVO mode and whether infusion rate modification is locked. It is important to assess how the interface alerts the user to these modes. In the Alaris two modes allow entry of vtbi: direct entry (*vtmode*) and via bag mode (*bagmode*). Both modes are indicated by a display of vtbi in the top line as can be checked using:

$$AG((vtmode \mid bagmode) \leftrightarrow topline = dispvtbi)$$

Furthermore it is possible to enter the infusion rate, when the device is in *rmode* and the top line shows holding.

$$AG(topline = holding \leftrightarrow rmode)$$

There are other state attributes that act as mode discriminators. Infusion rate and vtbi are displayed in a number of situations, hence the first two properties below are false. However the options menu is visible if and only if in query mode, and likewise the bags menu is only visible in the case of bags mode.

$$
\begin{aligned}
&AG(rmode \leftrightarrow middisp[drate])\\
&AG((bagmode \mid vtmode) \leftrightarrow middisp[dvtbi])\\
&AG(qmode \leftrightarrow middisp[dquery])\\
&AG(bagmode \leftrightarrow middisp[dbags])
\end{aligned}
$$

The weaker implications, that the variables to be changed are visible in the respective modes is true except in the case of bag mode (hence the second property below is false). Whether this is an issue is a matter for the domain expert.

$$
\begin{aligned}
&AG(vtmode \rightarrow middisp[dvtbi])\\
&AG(bagmode \rightarrow middisp[dvtbi])
\end{aligned}
$$

Checking these properties indicates that display of infusion rate or vtbi does not act as a mode discriminator.

Consistency of naming and purpose of functions

A second class of properties is concerned with the consistent appearance of function keys. Typical relevant properties include: a particular type of function is always mapped to the same function key; soft functions "ok" and "quit" always return the device to a display mode in which the top line shows "holding"; any display mode (as defined by the top line) will always show the same annotations to function keys. An illustration of the first property is to check that function key 3 is always mapped to "quit" if it is used.

$$AG(fndisp3 \mathrel{!=} fnull \rightarrow fndisp3 = fquit)$$

This property is false. There are a number of other uses of function key 3. For example in bags mode function key 3 shows "back" rather than "quit", a case that should be discussed with a domain or human factors expert. If this possibility is excluded then the property continues to fail.

$$
\begin{aligned}
&AG(((fndisp3 \mathrel{!=} fnull)\&!bagmode\&!(topline \text{ in } \{attention, vtbidone, setvtbi\}))\\
&\qquad \rightarrow fndisp3 = fquit)
\end{aligned}
$$

When the top line shows "attention" or "vtbi done" the device is in an alarm state. When the top line shows "set vtbi" an attempt is being made to start an infusion when no vtbi has been entered. These special cases can be presented to domain or human factors experts so that they can assess whether these exceptions are a cause for concern.

To check that if "quit" is a function attached to a key it is associated with key 3, the following property can be used.

$$AG((\mathit{fndisp1} = \mathit{fquit} \mid \mathit{fndisp2} = \mathit{fquit} \mid \mathit{fndisp3} = \mathit{fquit}) \ \rightarrow\ \mathit{fndisp3} = \mathit{fquit})$$

Furthermore, consistency of effect of function keys (returning to a top line showing holding) can be checked using:

$$AG((\mathit{fndisp1} = \mathit{fok}\ \&\ \mathit{topline}! = \mathit{options}) \ \rightarrow\ AX(\mathit{key1} \ \rightarrow\ \mathit{topline} = \mathit{holding}))$$
$$AG(\mathit{fndisp3} = \mathit{fquit} \ \rightarrow\ AX(\mathit{key3} \ \rightarrow\ \mathit{topline} = \mathit{holding}))$$

The first property demonstrates that the ok function always returns the device to a state in which the top line is holding unless the display is showing options. In the options case, ok will show information relevant to the option. In the case of quit the device always returns to the top line of holding. The final property aims to demonstrate that there is always the same function key configuration:

$$AG(\mathit{topline} = \mathit{volume} \ \rightarrow\ (\mathit{fndisp1} = \mathit{fnull}\ \&\ \mathit{fndisp2} = \mathit{fclear}\ \&\ \mathit{fndisp3} = \mathit{fquit}))$$

Although this is true in the case of volume it is not true for all top lines, for example the following property is false:

$$AG(\mathit{topline} = \mathit{dispvtbi} \ \rightarrow\ (\mathit{fndisp1} = \mathit{fok}\ \&\ \mathit{fndisp2} = \mathit{fbags}\ \&\ \mathit{fndisp3} = \mathit{fquit}))$$

because in bags mode the top line displays vtbi but the function keys show ok, null and back respectively. A further property demonstrates consistent effect of the clear function when the top line displays volume.

$$AG((\mathit{topline} = \mathit{volume}\ \&\ \mathit{infusionstatus} = \mathit{hold}) \ \rightarrow\ AX(\mathit{key2} \ \rightarrow\ \mathit{volumeinfused} = 0))$$

In this case if top line is “volume” and key2 is the action then the effect is always to change *volumeinfused* to 0.

Consistent behaviour of the data entry keys

These properties check that (1) keys will have a similar effect regardless of the value being changed or the mode in which the change is taking place and that (2) keys will have similar properties, for example reversibility, irrespective of the value changed or the mode. These properties can be illustrated in the case of (1) by the template:

$$AG((\mathit{mode}\ \&\ (\texttt{modeattribute})) \ \rightarrow\ AX(\texttt{chevron} \ \rightarrow\ \texttt{effect}(\texttt{modeattribute})))$$

The following illustrative property ranges over all the possible values of *infusionrate* (in the IVY tool meta-variables `IVAL1` for example can be defined that range over the values of a type or some subset thereof).

$$AG((rmode \ \& \ (infusionrate = \mathtt{IVAL1})) \ \rightarrow \ AX(sup \ \rightarrow \ infusionrate = \mathtt{IVAL1} + 1))$$

The mode of concern here is *rmode*. The attribute of concern is *infusionrate* and the effect that is to be consistent is that the action will increment the value of *infusionrate*. In this particular example the properties are false because the infusion rate lock can be switched on via the options menu. The following alternative properties:

$$AG((!rlock \ \& \ rmode \ \& \ (infusionrate = \mathtt{IVAL1})) \ \rightarrow \ AX(sup \ \rightarrow \ infusionrate = \mathtt{IVAL1} + 1))$$

are true except for the extreme case when infusion rate is maximum. The final property template is concerned with reversability.

$$AG(\mathtt{attribute} = \mathtt{value} \ \rightarrow \ AX(\mathtt{action} \ \rightarrow \ EX(\mathtt{attribute} = \mathtt{value})))$$
$$AG(\mathtt{attribute} = \mathtt{value} \rightarrow AX(\mathtt{action1} \rightarrow$$
$$\quad EX(\mathtt{action2}) \ \& \ AX(\mathtt{action2} \rightarrow (\mathtt{attribute} = \mathtt{value}))))$$

As an illustration of this class of properties, the following property is true.

$$AG((infusionrate = \mathtt{IVAL1} \ \& \ rmode \ \& \ !rlock) \ \rightarrow \ AX(sup \ \rightarrow \ (EX(sdown)\&$$
$$\quad AX(sdown \ \rightarrow \ infusionrate = \mathtt{IVAL1}))))$$

Here *sdown* provides the reverse for *sup*.

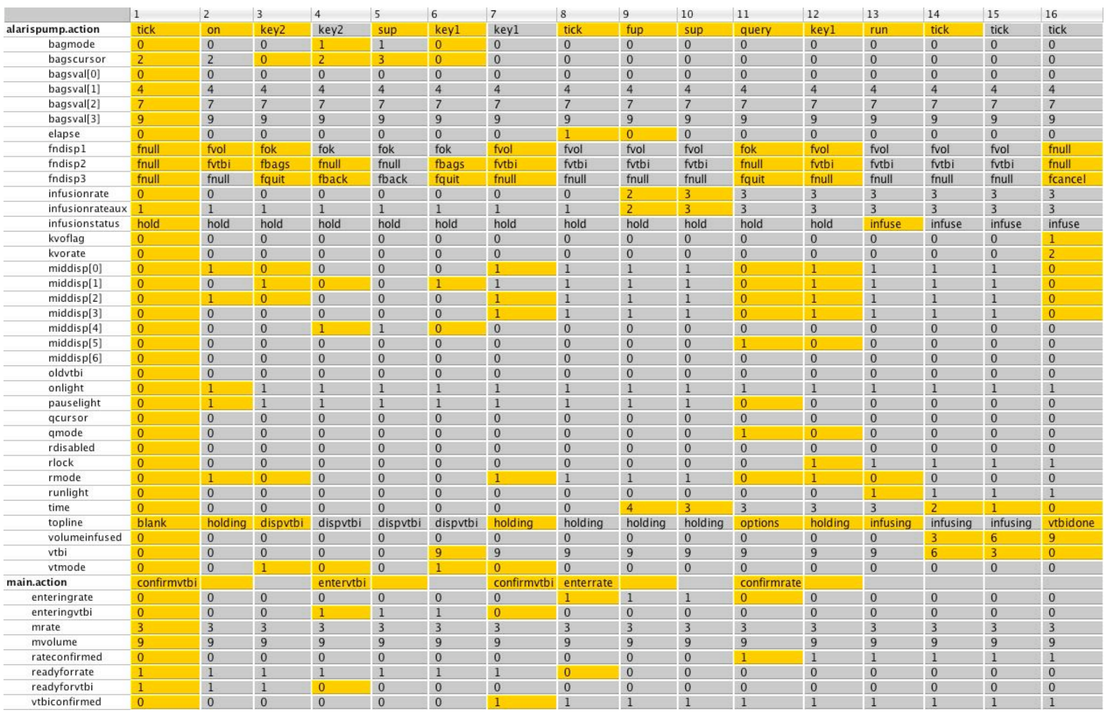
5.3 Supporting activities

The support of activities is explored by proving that specific goals related to the infusion pump can be achieved. Hence given an initial infusion rate and vtbi, the pump eventually stops with a volume infused equivalent to the original vtbi. To prove otherwise (which would provide useful counter-examples) amounts to checking:

$$AG((alarispump.infusionrate = \mathtt{IVAL1}\&alarispump.vtbi = \mathtt{IVAL2}) \ \rightarrow$$
$$\quad AG(alarispump.volumeinfused \ ! = \mathtt{IVAL2}))$$

The traces generated by the model checker as counter-examples can then be explored, consider for example $\mathtt{IVAL1} = 2$ and $\mathtt{IVAL2} = 9$ (see trace Figure 3). The activities involved include: (1) *entervtbi* (which starts at step 4); (2) *confirmvtbi* (which starts at step 7); (3) *enterrate* (which starts at step 8); and (5) *confirmrate* (which starts at step 11).

The infusion starts at step 13 and completes at step 19. A number of device actions are subsumed within these activities. Step 2 involves switching the device on, then moving to a top line of vtbi (step 3) then moving to bag mode, then going up one step in the bags menu using the chevron button (step 5, note that step 4 shows the beginning of the *entervtbi* activity) then exiting from VTBI and returning to a state where the top line is *holding* (step 7). At this point



	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
alarispump.action	tick	on	key2	key2	sup	key1	key1	tick	fup	sup	query	key1	run	tick	tick	tick
bagmode	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0
bagscursor	2	2	0	2	3	0	0	0	0	0	0	0	0	0	0	0
bagsval[0]	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
bagsval[1]	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4
bagsval[2]	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7
bagsval[3]	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9
elapse	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
fndisp1	fnull	fvol	fok	fok	fok	fok	fvol	fvol	fvol	fvol	fok	fvol	fvol	fvol	fvol	fnull
fndisp2	fnull	fvtbi	fbags	fnull	fnull	fbags	fvtbi	fvtbi	fvtbi	fvtbi	fnull	fvtbi	fvtbi	fvtbi	fvtbi	fnull
fndisp3	fnull	fnull	fquit	fback	fback	fquit	fnull	fnull	fnull	fnull	fquit	fnull	fnull	fnull	fnull	fcancel
infusionrate	0	0	0	0	0	0	0	0	2	3	3	3	3	3	3	3
infusionrateaux	1	1	1	1	1	1	1	1	2	3	3	3	3	3	3	3
infusionstatus	hold	hold	hold	hold	hold	hold	hold	hold	hold	hold	hold	hold	infuse	infuse	infuse	infuse
kvoflag	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
kvorate	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2
middisp[0]	0	1	0	0	0	0	1	1	1	1	0	1	1	1	1	0
middisp[1]	0	0	1	0	0	1	1	1	1	1	0	1	1	1	1	0
middisp[2]	0	1	0	0	0	0	1	1	1	1	0	1	1	1	1	0
middisp[3]	0	0	0	0	0	0	1	1	1	1	0	1	1	1	1	0
middisp[4]	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0
middisp[5]	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
middisp[6]	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
oldvtbi	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
onlight	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
pauselight	0	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0
qcursor	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
qmode	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
rdisabled	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
rlock	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1
rmode	0	1	0	0	0	0	1	1	1	1	0	1	0	0	0	0
runlight	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1
time	0	0	0	0	0	0	0	0	4	3	3	3	3	2	1	0
topline	blank	holding	dispvtbi	dispvtbi	dispvtbi	dispvtbi	holding	holding	holding	holding	options	holding	infusing	infusing	infusing	vtbidone
volumeinfused	0	0	0	0	0	0	0	0	0	0	0	0	0	3	6	9
vtbi	0	0	0	0	0	9	9	9	9	9	9	9	9	6	3	0
vtmode	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0
main.action	confirmvtbi			entervtbi			confirmvtbi	enterrate			confirmrate					
enteringrate	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0
enteringvtbi	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0
mrate	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
mvolume	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9
rateconfirmed	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1
readyforrate	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0
readyforvtbi	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0
vtbiconfirmed	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1

Figure 3: The activity counter-example

the vtbi is 9 and this permits *confirmvtbi* to be activated. At step 8 where *enterrate* is begun there is initially a *tick* (a random action that is not relevant to the intended process). The top line is holding — the device returns to this display mode after completing the *entervtbi* activity in step 7. The infusion rate is entered as 3 by first using the double chevron up button and then the single chevron up button. The confirm rate action is permitted in step 11. At this same step the *options* menu is selected and the rate lock option selected (because it is the first entry encountered). The next step (11) is to confirm it. This returns the device to the holding state, whereupon it is set to "infuse" using the run key. The infusion is completed after 3 *ticks* at which point the condition of the property is satisfied and the top line shows "vtbi done". It can be seen that the model checker has generated a plausible sequence of actions in finding a counter-example to the property. This sequence could be further analysed in a more qualitative way using a clinician or human factors specialist to explore the implications of this sequence.

6 Discussion and Conclusions

Analysing the infusion pump has produced promising results. The properties used in the analysis are largely based on standardisable patterns either generated by the IVY tool or easily generated from the requirements of the device and situation. Analogous properties can be systematically applied to other candidate pumps. Indeed this analysis has also been performed on the BBraun Infusomat pump [BBr05]. The checking of the properties raises a number of issues associated with the relation of the display to the underlying pump process and the mode structure of the particular device. For example, the analysis indicates that there are:

- possible confusions relating to the use of the display of vtbi in the top line;
- ambiguities about whether the pump is infusing or not using the top line of the display as a guide;
- inconsistent use of function keys.

The analysis says nothing about the significance of these issues. The method is to be used as part of a process including the active participation of human factors and domain specialists. In the context of use these discrepancies in the device may not be issues at all. An important aspect of the method is that it is systematic and that it has the potential to be reused for every candidate infusion pump. The inner level of the specification can be reused for every candidate pump. The outer activity layer can also be partially reused, guiding the analyst to create the appropriate constraints on the device model. The middle layer will be created afresh for each new device. In practice this part of the model continues to be substantial (perhaps a week's work for the second author). Further work is required to develop methods of reusing models that have similar structure or drive particular mode structures.

There is still some distance to go before procurers will find the techniques described here cost-effective, however the cost of using an infusion device extends beyond making a good deal on price and ongoing maintenance. It is also important to develop systematic techniques that will make it possible to explore the issues associated with the use of the device.

The work described here is part of the ongoing analysis of a range of infusion devices with the aim of reducing these costs. Part of this process should include a comparison with empirical techniques and with usability evaluation methods in terms of the range problems that can be uncovered by these different methods.

Acknowledgements: We acknowledge enthusiastically the help of Patrick Oladimeji of Swansea in particular and meetings with members of the EPSRC funded CHI+MED project (EP/G059063/1) in Queen Mary University, UCL Interaction Centre and Swansea. Ann Blandford, Andy Gimblett and Paolo Masci provided helpful comments. Trips to the sites were funded by the project.

Bibliography

[BB10] M. L. Bolton, E. J. Bass. Formally verifying human-automation interaction as part of a system model: limitations and tradeoffs. *Innovations in System and Software Engineering* 6(3):219–231, 2010.

[BBr05] BBraun. BBraun Infusomat Space: instructions for use. Technical report, B. Braun Melsungen AG, 2005.
www.bbraun.com

[BCNP07] E. Barboni, S. Conversy, D. Navarre, P. Palanque. Model-based engineering of widgets, user applications and servers compliant with ARINC 661 Specification. In Doherty and Blandford (eds.), *Interactive Systems: Design, Specification and Verification (DSVIS 2006)*. Springer Lecture Notes in Computer Science 4323, pp. 25–38. Springer-Verlag, 2007.

[BO07] L.-A. Bligard, A.-L. Osvalder. An analytical approach for predicting and identifying use error and usability problem. In Holzinger (ed.), *Proceedings of the 3rd Human-computer interaction and usability engineering of the Austrian computer society conference on HCI and usability for medicine and health care*. Springer Lecture Notes in Computer Science 4799, pp. 427–440. Springer-Verlag, 2007.

[BRRP05] J. Berstel, S. Reghizzi, G. Rouseel, P. Pietro. A scalable formal method for the design and automatic checking of user interfaces. *ACM Transactions on Software Engineering and Methodology* 14(2):124–167, 2005.

[Car06] Cardinal Health Inc. Alaris GP Volumetric Pump: directions for use. Technical report, Cardinal Health, 1180 Rolle, Switzerland, 2006.

[CCG$^+$02] A. Cimatti, E. Clarke, E. Giunchiglia, F. Giunchiglia, M. Pistore, M. Roveri, R. Sebastiani, A. Tacchella. NuSMV 2: An Open Source Tool for Symbolic Model Checking. In Larsen and Brinksma (eds.), *Computer-Aided Verification (CAV '02)*. Lecture Notes in Computer Science 2404. Springer-Verlag, 2002.

[CGP99] E. Clarke, O. Grumberg, D. Peled. *Model Checking*. MIT Press, 1999.

[CH01] J. Campos, M. Harrison. Model checking interactor specifications. *Automated Software Engineering* 8:275–310, 2001.

[CH08] J. Campos, M. Harrison. Systematic analysis of control panel interfaces using formal tools. In Graham and Palanque (eds.), *Interactive systems: Design, Specification and Verification, DSVIS'08*. Springer Lecture Notes in Computer Science 5136, pp. 72–85. Springer-Verlag, 2008.

[CH09] J. Campos, M. Harrison. Interaction engineering using the IVY tool. In Calvary et al. (eds.), *Proceedings of the ACM SIGCHI Symposium on Engineering Interactive Computing Systems*. Pp. 35–44. ACM Press, 2009.

[FA10] U. Food, Drug Administration. Infusion Pump Improvement Initiative. Technical report, Center for Devices and Radiological Health, April 2010.
http://www.fda.gov/MedicalDevices/ProductsandMedicalProcedures/GeneralHospitalDevicesandSupplies/InfusionPumps/ucm205424.htm

[Gin05] G. Ginsburg. Human factors engineering: A tool for medical device evaluation in hospital procurement decision-making. *Journal of Biomedical Informatics* 38:213219, 2005.

[Hol03] G. Holzmann. *The SPIN Model Checker, Primer and Reference Manual*. Addison Wesley, 2003.

[KA92] B. Kirwan, L. Ainsworth. *A Guide to Task Analysis*. Taylor and Francis, 1992.

[Kir94] B. Kirwan. *A Guide to Practical Human Reliability Assessment*. Taylor and Francis, 1994.

[LSH06] R. Lane, N. A. Stanton, D. Harrison. Applying hierarchical task analysis to medication administration errors. *Applied Ergonomics* 37:669679, 2006.

[MNMC08] J. L. Martin, B. J. Norris, E. Murphy, J. A. Crowe. Medical device development: The challenge for ergonomics. *Applied Ergonomics* 39:271283, 2008.

[NPDB06] D. Navarre, P. Palanque, P. Dragicevic, R. Bastide. An approach integrating two complementary model-based environments for the construction of multimodal interactive applications. *Interact. Comput.* 18(5):910–941, 2006.

[RBCB09] R. Ruksenas, J. Back, P. Curzon, A. Blandford. Verification-Guided Modelling of Salience and Cognitive Load. *Formal Aspects of Computing*, 2009. DOI: 10.1007/s00165-008-0102-7.

[TG08] H. Thimbleby, J. Gow. Applying Graph Theory to Interaction Design. In *Engineering Interactive Systems*. Springer Lecture Notes in Computer Science 4940, pp. 501–519. Springer-Verlag, 2008.

[TO09] H. Thimbleby, P. Oladimeji. Social Network Analysis and Interactive Device Design Analysis. In Calvary et al. (eds.), *Proceedings of the ACM SIGCHI Symposium on Engineering Interactive Computing Systems*. Pp. 91–100. ACM Press, 2009.

[ZJP$^+$03] J. Zhang, T. R. Johnson, V. L. Patel, D. L. Paige, T. Kuboseb. Using usability heuristics to evaluate patient safety of medical devices. *Journal of Biomedical Informatics* 36:2330, 2003.