Modelling information resources and their salience in medical device design

Michael D. Harrison
School of Computing Science,
Newcastle University,
Newcastle upon Tyne, NE1
7RU, UK &
michael.harrison@ncl.ac.uk

José C. Campos
Universidade do Minho, Dep.
Informática
& HASLab / INESC TEC
Braga, Portugal
jose.campos@di.uminho.pt

Rimvydas Rukšėnas
Queen Mary University of
London,
London E1 4NS, UK

Paul Curzon
Queen Mary University of
London,
London E1 4NS, UK
p.curzon@qmul.ac.uk

ABSTRACT

The paper describes a model that includes an explicit description of the information resources that are assumed to guide use, enabling a focus on properties of "plausible interactions". The information resources supported by an interactive system should be designed to encourage the correct use of the system. These resources signpost a user's interaction, helping to achieve desired goals. Analysing assumptions about information resource support is particularly relevant when a system is *safety critical* that is when interaction failure consequences could be dangerous, or *walk-up-and-use* where interaction failure may lead to reluctance to use with expensive consequences. The paper shows that expressing these resource constraints still provides a wider set of behaviours than would occur in practice. A resource may be more or less salient at a particular stage of the interaction and as a result potentially overlooked. For example, the resource may be accessible but not used because it does not seem relevant to the current goal. The paper describes how the resource framework can be augmented with additional information about the salience of the assumed resources. A medical device that is in common use in many hospitals is used as illustration.

Author Keywords

Human error, formal verification, performance, medical devices, model checking, user interaction error

ACM Classification Keywords

D.2.4. Software/Program Verification: D.2.2.Design Tools and Techniques: User Interfaces

INTRODUCTION

When analysing the usability of an interactive device it is often difficult to predict how the device could be used even when the tasks and activities that the design was intended for are relatively well understood. To explore the usability of a device, or the vulnerability of a device to user interaction error, a systematic exploration of "plausible" behaviours would provide a good baseline for analysis. This would enable analysis of properties of the plausible behaviours, for example consistency. The problem however is to identify those behaviours that can be discounted and ignored in such an analysis.

Previous work [5] has identified that the notion of information resource can be used systematically to explore the paths that a user will take in interacting with the device. However the resource approach, while restricting the set of behaviours for consideration, does not solve the problem because too many false positives are included (these behaviours while possible are not reasonable). This paper recognises that the salience of the resources available during an interaction are very significant in helping the user make reasonable decisions. Salience can arise as a result of factors such as visual significance or the amount of training that the user has had. The problem of [5] is solved by describing how resources *as well as* their salience can be specified as constraints over the model of a device. The systematic analysis of behaviour is then based on the possible interactions with the device that are governed by these constraints.

A model-based design process for interactive systems is adopted as a basis for this analysis. The models, that will be developed in the paper, include assumptions about how information resources afforded by the device are to be used. This contrasts with task modelling approaches [13], where such assumptions are separated from the user interface design. It has the advantage that these constraints can allow potential behaviours that were not envisaged in a task model and yet are plausible. It will be illustrated that this approach has value in the early stages of design, when working with design ideas, as

well as later stages [5] when a full specification of the system has been decided. A medical example is used to illustrate the method. This paper aims to focus analysis, limiting to more specific behaviours than are possible with the resource model by adding assumptions, not just about the presence and use of information resources, but also about their salience in the context of interaction.

BACKGROUND

Usability Engineering Techniques

All interactive systems encourage or discourage use in a variety of ways. The problem for the designer and developer of an interactive system is to understand what aspects of the design will affect use. While techniques such as cognitive walkthrough [15], and more rigorous techniques such as GOMS and their supporting tools [10, 20], are valuable in the development of interactive systems from this perspective, the advantage of the approach described in the paper is that it is both systematic and exhaustive. Our approach assumes *explicitly* that resources place constraints on user action. It assumes that resources help users to take the next step in the interaction to achieve the work that the system is designed to support. The model so developed can be analysed to assess the consequences of these assumptions. The example introduced in the paper is to determine whether the modes of a medical infusion pump device are unambiguous and effectively signposted. A scenario is developed where mode confusion can be particularly problematic for users when programming identical pumps — patients often require multiple infusions in intensive care, and setting up these multiple pumps is often a task that is done under time constraint.

The BBraun infusomat [3] is a medical infusion pump which is used in many contexts in hospitals to infuse medication intravenously. This device will be used to illustrate the approach. These pumps are designed to infuse a volume (vtbi – **v**olume **t**o **b**e **i**nfused) of medication intravenously, at a prescribed infusion rate or over a prescribed time. The device supports two basic modes. The prescription values can be set up in one mode (*paused*) and the infusion process takes place in the other mode (*infusing*). Infusion is an error prone process for reasons including the programming of the infusion device (there are other issues including wrong drug, wrong prescription as is outlined in [11]). This situation is exacerbated when clinicians are under pressure, for example when multiple pumps are being programmed at the same time. The criticality of the situations in which these devices are used justifies a rigorous approach to their analysis.

The present paper extends [5] by adding salience to the assumptions about the information resources that guide interaction. It takes the model of salience, the scenario and salience assumptions (about the scenario) from previous work described by Rukšėnas et al. in [18]. A scenario (multiple device infusions) is considered in which the resource approach used alone generates more paths than are plausible. Some are not relevant to the analysis and can safely be omitted. For example, as seen in Figure 2, multiple uses of the same action are included because the actions are simple and subsequent invocations have no effect. An abstract model of the BBraun device is used in the analysis that emphasises the mode structure of the device and abstracts other features such as the details of number entry. While this analysis is performed on an actual device design it is envisaged that such analysis could be done during the early stages of design. In such an analysis of a preliminary design the mode structure of the proposed device might be conjectured rather than fleshed out. The full model of salience is taken from work first presented in [17]. Salience assumptions are based on the user's interaction with the device and include the effects of learned procedures, the last action performed or how physically salient the resource is. The paper argues that this added detail can be a valuable analysis tool when considering the design of an interface. The scenario, as described in [18], is concerned with establishing whether any feature of the design of the system may be made less prone to error when programming multiple infusion pumps. It has been observed by [2], for example, that the location of prescription forms can have an effect on the likelihood of errors when transcribing from multiple prescriptions. A step towards exploring these design issues is to include, as additional resources, where the prescription is in relation to the position of the infusion device and whether an interleaving strategy has been part of the user training.

To analyse the suitability of the device for performing the given task a model of the device is first produced. This is then augmented, first by modelling the activities and resources that are involved and then by focussing on the salience of the resources at different stages of the activity. Before describing the model and its augmentations, the modelling approach is introduced and the BBraun device described at an appropriate level of abstraction.

Resources

Information needs, and the resources afforded by a designed system, provide constraints on, and triggers for, user behaviour. These constraints and triggers *shape* user behaviour. They lie behind the *situated actions* described by Suchman [19] and the *resources* described in Distributed Cognition [9]. If an appropriate resource is not available to fulfil an information need, then the user's ability to achieve a desired goal may be compromised.

Campos et al. [5], using work first presented in [21], describe how resources, as constraints on user action, can be used to analyse (and contrast) plausible user behaviours in different devices. Assumptions about resource constraints are expressed as formal properties over a model of the device. Action availability is resourced when the relevant option is currently on the screen, and the action is appropriate for the activity being carried out. Action effect information is available (afforded) when an appropriate label is provided. This information shapes what the user does in a typical interaction.

Two more elements contribute to the resource analysis: activities, that capture how the device is used, and activity related attributes to capture the stages of the activities of which the users are aware.

2

A model checker is then used to generate paths that satisfy the constraints. These paths, because they are governed by the resource assumptions, are considered to be "plausible". Paths generated are only those that satisfy the constraints presumed to influence use. The plausible paths can be analysed to explore the implications of these assumptions. This analysis can involve simple inspection of the sequence of actions, creating scenarios based on the sequence or by proving properties of actions in relation to the context in which they can occur, for example proving their consistency

The effectiveness of the resources depends on the expertise of the user and the salience of the information.

Salience

Simple availability is not always enough to explore plausible interactive behaviour. A richer model of how the device is being used is needed to provide additional insight about the effect of a potential design. Further detail can be achieved by analysing the salience of these resources, exploring the relevance and timeliness of the resource at a particular stage in the interaction in relation to the prescribed activities.

The notion of salience used in the framework is informed by the ideas of activation theory [1]. Four distinct notions of salience have been identified [17] : specificity, cognitive salience, procedural salience and sensory salience. *Specificity* defines the dynamic relevance of an information resource to an activity. It captures the "just in time" trigger provided by the information. *Cognitive salience* captures the task knowledge assumed of the user to decide which action to take next. This information is relevant irrespective of how the particular device is implemented. In the case of the example device, setting up an infusion involves (in any order) entering the volume to be infused (vtbi) and *either* entering the prescribed infusion rate or the time over which the infusion is to be completed. Which of these combinations is required depends on what information is provided by the prescription. The cognitive salience of entering time or rate is increased if the vtbi has already been entered irrespective of how the device is designed. *Procedural salience* is salience that results from training in the use of the particular device's programmed sequences because of the way the prescription is organised. Finally *sensory salience* is triggered by available visual or auditory cues that remind the user of the next action. These notions of salience may be thought of as determining the user significance of the triggers intended in original definition of information resource [21].

Assumptions about the salience of a resource can be used to give strength to the constraints on action defined by the resources. The remainder of the paper demonstrates how salience can be integrated into the resource based analysis in [5]. The overall modelling and analysis approach is briefly discussed, as well as the modelling language used. Resources and their salience are modelled as constraints on the actions or activities within a system involving devices and people. They are then used for the analysis of plausible interactions with a given system. Resource modelling and analysis, and salience modelling and analysis are then described. Limitations of the resource based approach are noted for the present example.
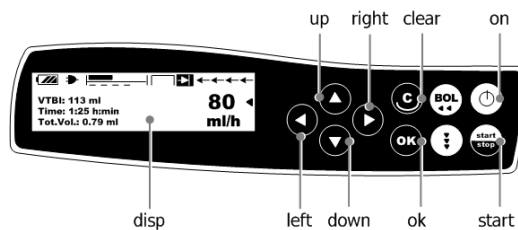

Figure 1. The B Pump (relevant attributes and actions)

As a final stage in the analysis of the example, salience assumptions are added to enrich the resource assumptions to further focus the plausible interactions between users and devices. This richer analysis enables the discovery of issues that have previously been identified by experiment [2]. The paper ends with a discussion of related work.

MODELLING

The model has elements relating to the device itself and its interface. Activities are then added to the device model that aim to capture the intended "work" for which the device is to be used. The device is intended for a purpose and part of the analysis involves making these activities that reflect the purpose explicit. Activities may be inferred by observing the existing system or may be envisaged by considering the role for which the new design is intended. Once the actions relating to the device and the activities relating to the work have been modelled the next step is to describe how action and activities are facilitated by information resources. These facilities are expressed as constraints in the model so that it is possible to explore those actions that are governed by these information resources.

Device Modelling

The design of the BBraun infusion pump [3] forms the basis for the analysis. The device is illustrated in Figure 1. It has a small display and supports a number of functions through ten multi-use keys. When the infusion device is in a "paused" mode, the basic pump variables: infusion rate, vtbi, time and volume infused, can be modified. Other settings can also be changed as is described in the fuller model discussed in [5]. These settings will not be the focus of the analysis of the present paper and are therefore omitted from the model under consideration. Entering vtbi is achieved in the BBraun pump when the device is paused by selecting the vtbi option in the device's main menu and then entering the value of vtbi using four keys (up, down, left and right). When the value to be infused has been entered the user must press the *ok* key to confirm that the value has been entered.

At the first level of the model is the device: its state and available actions. The device model describes the actions that can be performed with the device, when they are permitted to occur and the effects that they have on the device. Modal Action Logic (MAL) is used because it describes the state transitions in a similar form to that required in graphical notations such as Simulink statecharts. These notations are increasingly being adopted by industry [16] and there is some evidence that this style of specification notation is preferred by developers

3

[12]. The IVY environment for the development of interactive systems that supports MAL is described in [6]. An example of the form of the specification used is as follows. The specifications used in the paper can be found at our HCI specifications repository[1]. The aim of the device model is to describe features of the device, faithfully and precisely, that capture how mode transitions occur in the device. The analysis is not concerned with the details of how numbers are entered. For this reason a simple action "enter" specifies the entry of a number. The individual actions that are required for such an action in the case of the BBraun are abstracted. The result of entering a number in the model is simply that the number becomes "entered" (the variables are therefore specified as boolean). The device model therefore has the following form:

interactor $device$
  types
    $mode = \{off, hold, infuse\}$
    $dispmode = \{dblank, mainmenu, dvtbi, drate,$
                $dtime, dinfusing\}$
  attributes
    $m : mode$
    [vis] $dm : dispmode$
  actions
    [vis] $enter$
  axioms
    $per(enter) \rightarrow dm\ in\{dvtbi, dtime, drate\}$
    $dm = dvtbi \rightarrow [enter]\ vtbi'\ \&\ keep(m, dm)$
    $dm = drate \rightarrow [enter]\ rate'\ \&\ keep(m, dm)$
    $dm = dtime \rightarrow [enter]\ time'\ \&\ keep(m, dm)$

Two interaction modes are introduced by this specification. m is a mode used to specify whether the device is off, holding or infusing. dm specifies the display mode, describing whether the device is showing the main menu, or is in a mode in which the device will accept entry of a particular pump variable. Display modes indicate that the user can enter vtbi ($dvtbi$), infusion rate ($drate$) or time ($dtime$) for example. Priming is used to specify the new value of an attribute after the action has occurred. $vtbi'$ specifies that a new value of $vtbi$ after the action will be $true$. $keep$ indicates that the listed attributes keep their original values. In the event of there being no explicit change to an attribute or use of $keep$ the attribute will take a random value. The specification also describes when the action is permitted. Hence $per(enter) \rightarrow dm\ in\ \{dvtbi, dtime, drate\}$ specifies that $enter$ is only permitted if in a number entry mode dm is $dvtbi$, $dtime$ or $drate$.

As already discussed entering vtbi, infusion rate or time in reality involves cursor movement using the up and $down$ keys (see Figure 1) and, when the appropriate entry in the menu is highlighted the ok key is used to enter the mode that allows entry of the relevant pump variable. In addition to this the model used in the present analysis simply assumes there is an action that changes the mode. This action, in the case of infusion rate is $chooserate$, that combines the actions that are

---

supported by the BBraun pump. Since the details of the implementation are not relevant at this stage of the analysis, and may be implemented in a variety of ways including the actual implementation, they are not modelled in detail. A fully detailed model of the interface to the same pump and its associated resources is described in [5]. The focus of the current simplifying abstraction is to focus on the modes associated with entering these different values.

ACTIVITY MODELLING
Activity modelling is concerned with identifying assumed activities for which the device will be used. This is done by specifying the activities of actions intended to describe aspects of user activities. Hence $entervtbi$ is an activity that involves entering the vtbi and is specified as a MAL action. This activity assumes also that the user is aware of the stage that they are at within the activity. Hence $phasevtbi$ is a state attribute that is included in the model to represent what the user knows in terms of entering vtbi: they are ready to do it ($ready$); they are doing it ($entering$); they have done it and it has been confirmed ($confirmed$). The activity attributes can be considered to be information resources, as is discussed in the next section, that are internal to the user. They "know" where they have got to. The assumption is that their use of the device will reinforce their progress within the activity.

Since the work to be explored concerns the entry of infusion rate and vtbi on two infusion devices, the activities are duplicated to include two instances of the infusion device and associated activities. The actions describing activities should be independent of the particular details of the device.

RESOURCE MODELLING
The resource modelling stage involves modelling the information that is assumed to be significant in triggering an action. Analysis of properties of the device is then restricted to those paths that the device allows as constrained by the resource assumptions.

The device action $choosevtbi$ selects the mode $dvtbi$ when the device is in the main menu mode. The behaviour of the action for the device is described using MAL as:

$per(choosevtbi) \rightarrow dm = mainmenu$
    $[choosevtbi]\ dm' = dvtbi\ \&$
        $keep(m, vtbi, time, rate, clamp)$

This specifies that the action $choosevtbi$ will only be enabled if the device is in the main menu mode. The effect of $choosevtbi$ is to change the mode to $dvtbi$.

The information resource constraints leading to this action of the device now requires an understanding that two devices are involved. The constraints for $choosevtbi$ are illustrated for the second device ($device2.choosevtbi$). The resource constraint is expressed as a permission. The action is only permitted if the value for vtbi has not been entered ($!vtbi$), there is a prescribed value for vtbi written on the prescription form from the pharmacy ($mvtbi2$) and the user understands that this is the stage of the activity in which vtbi is being entered in device 2 ($phasevtbi2 = entering$). In this way the

4

only actions that can appear in traces generated by the model are ones in which this resource constraint holds for the action.

$$per(device2.choosevtbi) \rightarrow$$
$$!device2.vtbi \& mvtbi2 \& (phasevtbi2 = entering)$$

A further permission relates to the confirmation action. This suggests that confirmation will happen whenever the user realises that an entry phase has been completed (e.g., $phasevtbi2 = confirmed$) and the device is in the mode relevant to the value being entered (e.g. $device2.dm = dvtbi$).

$$per(device2.confirm) \rightarrow$$
$$((phasevtbi2 = confirmed) \& (device2.dm = dvtbi)) |$$
$$((phaserate2 = confirmed) \& (device2.dm = drate)) |$$
$$(phasetime2 = confirmed) \& (device2.dm = dtime))$$

The activity of reaching the relevant stage in the activity is specified as:

$$[confirmvtbi2]phasevtbi2' = confirmed \&$$
$$keep(phasevtbi1, phasetime1, phasetime2,$$
$$phaserate1, phaserate2, phaseopenclamp1,$$
$$phaseopenclamp2, phaseinfuse1, phaseinfuse2)$$

The key to understanding this transition is to understand the resources that trigger it.

$$per(confirmvtbi2) \rightarrow device2.m ~! = ~off \&$$
$$(phasevtbi2 = entering) \&$$
$$(mvolume2 = device2.vtbi) \&$$
$$(device2.dm = dvtbi)$$

The user's engagement in the $confirmvtbi2$ activity is triggered when the current phase of the activity is that it is entering, the value of vtbi as set in the second pump is the same as the prescribed valuer and the current mode of the device allows vtbi to be entered.

An important stage in any modelling process is to check the plausibility of the model. One aspect of plausibility is to check that the transition between confirming one activity and starting another is always marked by the confirm action.

$$AG(device2.confirm \rightarrow$$
$$!AX(phasevtbi2 = entering |$$
$$phasetime2 = entering |$$
$$phaserate2 = entering |$$
$$phaseinfuse2 = entering |$$
$$phaseopenclamp2 = entering))$$

This property asserts that it is never the case that immediately after invoking the confirm action any of the activities for the same device are in an entering phase. The property is true.

RESOURCE-BASED ANALYSIS

Once the information resource assumptions have been established as demonstrated in the previous section, the resulting model can be analysed. The analysis typically involves investigating a number of questions. The first question is concerned with the paths that are generated that satisfy the resource assumptions and at the same time achieve the goals of the various activities. This analysis explores plausible sequences that achieve the goals of infusing the prescribed rate

and vtbi, while at the same time setting up the infusion device appropriately. The IVY tool [6] is used to check properties by translating the specification into NuSMV [8]. This generates a representation of the model in which all possible paths can be checked subject to the conditions specified by the property. For example, Figure 2 illustrates a path where the device has started infusing and the volume infused is as specified by the prescription. The following property can be used to check whether the device can start infusing subject to resource constraints before both clamps have been opened:

$$AG(!(phaseinfuse1 = entering \&$$
$$phaseinfuse2 = entering \&$$
$$device1.clamp \& device2.clamp))$$

Checking the property reveals a counter-example that describes a sequence of actions that completes the process subject to the given constraints (see Figure 2 for example). The figure describes one path: a sequence of states that demonstrates a situation in which the property is false. Typically there are many such sequences. The counter-example generated by the model checker is likely to be the shortest in this particular case. Analysis usually proceeds by exploring the counter-examples by adding conditions that exclude already found counter-examples until a property holds true.

The figure describes a sequence starting from the initial state (column 1), ending at a state where the property fails to be true (column 16 in this case). Columns indicate values held by attributes. These are named in the left hand column (i.e., column 0). For example, the attribute m has value $hold$ in column 6. The colour yellow is used to indicate that a state attribute has changed value between successive states.

The problem with this counter-example is that while it achieves the required goals for the two devices it indicates a trace that is not likely in practice. The sequence indicated in Figure 2, see rows marked $device1.action$, $device2.action$ and $main.action$, indicates that:

1. the two devices are set up (columns 1 and 2)

2. the enter rate phase started (column 3) and rate entered for device 2 (column 4)

3. the clamp is opened and rate confirmed for device 1 (column 5)

4. the rate is entered for device and the clamp is opened for device 2 and confirmed (column 9)

5. vtbi is entered for device 2 twice and is confirmed (column 11) then the clamp is closed for both devices prior to starting to infuse (column 16).

One weakness of the resource framework is that resources do not necessarily prevent duplication of actions. The abstraction of number entry action makes the action trivial and therefore in this particular path it is repeated. Also no idea of how clinicians are trained is folded into the analysis and therefore the order in which the rate and vtbi are entered is not guaranteed.

Figure 2. Constrained interaction achieving activity goals

	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
device1.action	setup	mainmenu	mainmenu	openclamp	confirm		chooserate	enter	enter	enter	confirm	choosevtbi	enter	confirm	closeclamp
clamp	TRUE	TRUE	TRUE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	TRUE
dm	dblank	mainmenu	mainmenu	mainmenu	mainmenu	mainmenu	mainmenu	drate	drate	drate	mainmenu	dvtbi	dvtbi	mainmenu	mainmenu
m	off	hold	hold	hold	hold	hold	hold	hold	hold	hold	hold	hold	hold	hold	hold
rate	FALSE	FALSE	FALSE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE
time	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	TRUE	TRUE	TRUE
vtbi	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	TRUE	TRUE	TRUE
device2.action	setup	chooserate	enter	enter	confirm		openclamp		choosevtbi	enter	enter	confirm	closeclamp	start	
clamp	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	TRUE	TRUE	TRUE
dm	mainmenu	drate	drate	drate	mainmenu	mainmenu	mainmenu	mainmenu	dvtbi	dvtbi	dvtbi	mainmenu	mainmenu	dinfusing	dinfusing
m	hold	hold	hold	hold	hold	hold	hold	hold	hold	hold	hold	hold	hold	infuse	infuse
rate	FALSE	FALSE	TRUE	TRUE	TRUE	TRUE	TRUE	FALSE	FALSE	FALSE	FALSE	TRUE	TRUE	TRUE	TRUE
time	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	TRUE	TRUE	TRUE	TRUE	TRUE
vtbi	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE
main.action		enterrate2	enteropenclamp1	confirmrate2	confirmopenclamp1	enteropenclamp2	enterrate1	confirmopenclamp2	entervtbi2	confirmrate1	confirmvtbi2	entervtbi1	startinfuse2	confirmvtbi1	startinfuse1
phaseinfuse1	ready	ready	ready	ready	ready	ready	ready	ready	ready	ready	ready	ready	ready	ready	entering
phaseinfuse2	ready	ready	ready	ready	ready	ready	confirmed	confirmed	confirmed	confirmed	confirmed	confirmed	entering	entering	entering
phaseopenclamp1	ready	ready	entering	entering	confirmed	confirmed	confirmed	confirmed	confirmed	confirmed	confirmed	confirmed	confirmed	confirmed	confirmed
phaseopenclamp2	ready	ready	ready	ready	ready	entering	entering	confirmed	entering	confirmed	confirmed	confirmed	confirmed	confirmed	confirmed
phaserate1	ready	ready	ready	ready	ready	ready	entering	entering	confirmed	confirmed	confirmed	confirmed	confirmed	confirmed	confirmed
phaserate2	ready	entering	entering	confirmed	ready	ready	ready	ready	ready	ready	confirmed	confirmed	confirmed	confirmed	confirmed
phasetime1	ready	ready	ready	ready	ready	ready	ready	ready	ready	ready	ready	ready	ready	ready	ready
phasetime2	ready	ready	ready	ready	ready	ready	ready	ready	ready	ready	ready	ready	ready	ready	ready
phasevtbi1	ready	ready	ready	ready	ready	ready	ready	ready	ready	ready	ready	entering	entering	confirmed	confirmed
phasevtbi2	ready	ready	ready	ready	ready	ready	ready	ready	entering	entering	confirmed	confirmed	confirmed	confirmed	confirmed

The model also supports a more detailed assessment of the actions and their role in the interaction. For example, action consistency is an important consideration when using a device. In the case of the present model, because many of the actions have been abstracted, consistency is more difficult to analyse but it is possible to explore for example that whenever a pump variable is entered for a particular device then it is confirmed before attempting to enter another pump variable for that device. One instance of this requirement is expressed as:

$$phasevtbi1 \ = \ entering \ \rightarrow \ phaserate1 \ ! = \ entering$$

A further development of this property would be to show that the confirm action for device 1 must occur before $phasetime1$ can become entering.

$$AG(phasevtbi1 = entering \ \rightarrow$$
$$AX(phasevtbi1 = confirmed \ \rightarrow$$
$$!E[!device1.confirm \ U$$
$$(phasetime1 = entering \ |$$
$$phaserate1 = entering)]))$$

This property is true for both devices.

SALIENCE MODELLING

The example trace of Figure 2 illustrates that a focus on resource availability alone is not sufficient to generate relevant actions in a way that is consistent with typical practice. It does however provide useful feedback about the implications of resource assumptions. A more detailed analysis of action salience and its effect in biasing user activity can provide more insight into the plausibility of possible paths. In the example scenario a clinician sets up infusion on two identical devices, taking the information from a paper description of two prescriptions. Information may be available on the prescription form but it may not be sufficiently salient for the clinician to use it effectively. When therapy involves multiple infusion pumps this situation is relevant to design.

A third level of modelling and analysis addresses problems such as these. The focus of this addition to the model is how information resources combine to bias choice of action. The availability of these resources depend on a number of factors including additional resources that are not included in the previous model. For example, the proximity of the prescription form to the pumps will be a considered element in the process. Depending on how close the prescription is, the clinician user can choose either:

1. to remember both prescribed values for pump 1 and then enter them before moving to pump 2, or

2. to adopt a strategy in which the vtbi is entered for each pump before moving to enter infusion rate (an interleaving strategy).

These additional resources are combined with the resources specified in the earlier model to produce a new model that includes salience assumptions. The new model specifies the same activities as in the resources case. However $Tconfirmvtbi1$ differs from $confirmvtbi1$ by adding salience assumptions associated with the resources.

Hence

$$per(Tconfirmvtbi1) \rightarrow (totalcv1 \geq total) \& !done$$

This activity is only permitted if the salience associated with the activity ($totalcv1$) exceeds a threshold ($total$). Salience of resources is relative and dynamic, depending on the stage of the interaction. At any stage in the interaction two resources may be more or less salient relative to each other. The model defines salience numerically to achieve such comparison. This value will vary during the interaction and must therefore be calculated at each interaction. The highest value for currently enabled activities leads to the choice of the next activity. When there are several such activities, the choice is non-deterministic in the model. $totalcv1$ specifies the current salience value for $Tconfirmvtbi1$. The activity is enabled if the salience exceeds a threshold. This threshold is successively reduced if no activity currently has a salience that exceeds it. The approach follows that described in [17]. The total salience for each activity is calculated based on its specificity as well as its cognitive, procedural and sensory salience.

Specificity inherits the resource constraints already discussed in the earlier analysis. It captures the dynamic aspect of resource availability and relevance. Confirmation of vtbi is specific only if the device is in the vtbi entry mode and indicates this on its display ($device1.dm = dvtbi$) and the vtbi has been entered ($device1.vtbi$ is true). Furthermore, the user is either focused on device 1 ($pump = one$) or programs both devices simultaneously ($interleave$ is true). Thus, specificity is characterised by the resource constraints already specified in the more general resource approach described in the previous sections with additional elements that are considered to be critical to the salience of these resources. These additional elements are associated with whether an interleaving strategy is being used and which of the two pumps is the current focus.

$$speccv1 = (device1.vtbi \wedge$$
$$(device1.dm = dvtbi) \wedge$$
$$(interleave \,|\, (pump = one)))$$

The need for the confirmation step in the context of number entry task depends on the device design and is not a feature of the device independent task. For this reason cognitive salience has no meaning for the confirmation action and therefore $cogcv1$ is false in the model.

Procedural salience depends on whether the last activity (defined by $lastactivity$) the user engaged in involved entering vtbi for device 1 which assumes knowledge of how the device is used.

$$proccv1 = (lastactivity = Xentervtbi1)$$

Sensory salience for the confirmation of vtbi is assumed to be true only when the activity is specifically supported by the device. This is so since the ok key is next to the number entry keys and, therefore, the action of pressing it is assumed to be sensorily salient provided it is dynamically relevant.

The overall salience combines the cognitive, procedural and sensory salience values. However, an activity must be specific (resourced) for its sensory and cognitive salience to make an impact. If it is, then its degree of overall salience depends on whether the activity is cognitively, sensorily or procedurally salient — the degree of salience ($totalcv1$) is described by a set of rules below. Note that when an activity is not specific, its overall salience is only taken to be significant if the activity is procedurally salient. It is assumed that because the user knows how the device works information resources are not required to assist in taking the next action.

$$((specAct \& cogAct \& sensAct \& procAct) \rightarrow$$
$$totalAct' = salient + salient + salient) \&$$
$$((specAct \& cogAct \& sensAct \& !procAct) \rightarrow$$
$$totalAct' = salient + salient) \&$$
$$((specAct \& (cogAct != sensAct) \& procAct) \rightarrow$$
$$totalAct' = salient + salient) \&$$
$$((specAct \& !cogAct \& !sensAct \& procAct) \rightarrow$$
$$totalAct' = salient) \&$$
$$((specAct \& (cogAct != sensAct) \& !procAct) \rightarrow$$
$$totalAct' = salient) \&$$
$$((!specAct \& procAct) \rightarrow$$
$$totalAct' = notsalient) \&$$
$$((specAct \& !cogAct \& !sensAct \& !procAct) \rightarrow$$
$$totalAct' = notsalient) \&$$
$$((!specAct \& !procAct) \rightarrow$$
$$totalAct' = notsalient) \&$$

Salience values are either *salient* (taken to be 1) or *notsalient* (taken to be 0). Since $cogcv1$ and $senscv1$ are false and true, respectively, the above pattern for the degree of salience $totalcv1$ reduces to the following cases:

$$((speccv1 \wedge proccv1) \rightarrow$$
$$totalcv1' = salient + salient) \wedge$$
$$((speccv1 != proccv1) \rightarrow$$
$$totalcv1' = salient) \wedge$$
$$((!speccv1 \wedge !proccv1) \rightarrow$$
$$totalcv1' = notsalient)$$

Intuitively, $totalcv1$ depends on the procedural and sensory (subject to specificity) salience of the activity.

The confirm vtbi activity invokes the relevant device action $device1.confirm$.

$$[Tconfirmvtbi1] \; effect(device1.confirm) \wedge$$
$$pump = one \wedge done \wedge cv1completed \wedge$$
$$lastactivity = Xconfirmvtbi1 \wedge$$
$$total = ceilingsalience$$

It determines that the last pump used is the one denoted *one* (and it is assumed that this is memorised by the user). Likewise it is assumed that the fact that this is the last activity is also memorised $lastactivity = Xconfirmvtbi1$. *done* set to true indicates that saliences must be recomputed, and the total salience threshold is returned to its maximum value ($ceilingsalience$).

SALIENCE-BASED ANALYSIS
Salience enables a consideration of paths that are conditioned by training as well as other cognitive effects. These factors further restrict the paths considered over those generated by

the resource analysis. An example of the role that these additional attributes play can be found in the definition of specificity and sensory salience associated with memorising the infusion variables.

$$specmem1 = (!mem1completed \ \& \ !close)$$
$$sensmem1 = close$$

Specificity in this case depends on whether the information has been memorised before and whether it is worth memorising it (if the prescription is close then it is not worth it). Sensory salience indicates that, on the contrary, memorising is more likely to happen if the prescription cue is close. The closeness of these resources is also relevant in other activities, for example entering vtbi and entering time where the values of vtbi and time are critical for completing the activity. Cognitive salience only contributes if the information has been memorised or the prescription is close, for example:
$$cogev1 = (mem1completed \ | \ close)$$

These salience assumptions provide constraints that focus the plausibility as illustrated in Figure 4. It allows exploration of whether actions can be omitted in achieving the goals of setting up both infusions. In the resource analysis actions could repeated as was seen in the example trace (Figure 2) because their repetition had no effect on the result of an activity. In this example it is possible that key actions are also omitted as was noted in [2]. Consider, for example, the step of opening the roller clamp. It would be helpful if the user is biased towards this action rather than ignoring it when starting the infusion. In the analysis, the relevant property to check is whether it is *always* the case for both pumps that their roller clamp is open when the infusion key is pressed. In the case where the prescription form is located close to the pumps, model checking the property produces a scenario where this condition is not satisfied for the first pump. This suggests that appropriate user bias may not be sufficiently strong in some cases. It turns out that this occurs when some aspects of the programming of the two pumps is interleaved (i.e., both vtbi values are first entered, then both time values, etc.) as opposed to sequential programming of one pump after the other. In such a situation, the salience of the correct action (opening the roller clamp) is at most as high as that of the erroneous action (pressing the infusion key). Interleaving is plausible, particularly if the prescription is organised so that entering both vtbis before entering both times or rates seems an efficient strategy to the user. Since observation leads to a suspicion that the process can lead to failure to open the roller clamp it makes sense to consider:

$$AG((inf1completed \ \& \ inf2completed) \rightarrow$$
$$(oc1completed \ \& \ oc2completed))$$

These suspicions are confirmed by the trace generated (Figure 3).

1. The two devices are switched on (the order in which this happens is not constrained) (not shown).

2. vtbi is entered for device 1 and confirmed, then time is chosen for device 1 and entered and confirmed (not shown).

	31	32	33	34	35	36	37	38	39	40	41	42	43	44
device1.action														
clamp	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE
dm	mainmenu	mainmenu	mainmenu	mainmenu	mainmenu	mainmenu	mainmenu	mainmenu	mainmenu	mainmenu	start	mainmenu	mainmenu	mainmenu
m	hold	hold	hold	hold	hold	hold	hold	hold	hold	hold	dinfusing	infuse	infuse	infuse
rate	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE
time	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE
vtbi	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE
device2.action														
clamp	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
dm	dtime	enter	dtime	dtime	confirm	mainmenu	mainmenu	openclamp	mainmenu	mainmenu	start	mainmenu	mainmenu	mainmenu
m	hold	hold	hold	hold	hold	hold	hold	hold	hold	hold	hold	hold	hold	dinfusing
rate	FALSE	FALSE	FALSE	FALSE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE
time	FALSE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE
vtbi	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE
main.action	dropthroughroller	Tentertime2	setupround	dropthroughroller	Tconfirmtime2	setupround	dropthroughroller	Topenclamp	setupround	dropthroughroller	Tstartinfuse	setupround	dropthroughroller	Tstartinfuse
cht1completed	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE
cht2completed	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE
ct1completed	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE
ct2completed	FALSE	FALSE	FALSE	FALSE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE
cv1completed	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE
cv2completed	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE
done	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	TRUE
et1completed	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE
et2completed	FALSE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE
ev1completed	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE
ev2completed	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE
inf1completed	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	TRUE	TRUE	TRUE	TRUE
inf2completed	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	TRUE
interleave	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE
lastactivity	Xchoosetime	Xentertime2	Xentertime2	Xentertime2	Xconfirmtime2	Xconfirmtime2	Xconfirmtime2	Xopenclamp	Xopenclamp	Xopenclamp	Xstartinfuse	Xstartinfuse	Xstartinfuse	Xstartinfuse

Figure 3. Are the clamps always dealt with?

3. vtbi is entered for device 2 (not shown) and confirmed (not shown), then time is chose for device 2 and entered (column 32) and confirmed (column 35).

4. The clamp is opened for device 2 (column 38).

5. Infusion is started for device 2 (column 41) and then for device 1 (column 44).

The trace has two interesting features. Firstly, because the prescription is close to the pumps, entry is sequential rather than interleaved. In this particular case device 1 is entered then device 2 is entered. The sequence is determined by the salience of the resources in contrast to the earlier resource analysis. Secondly, open clamp 1 is omitted. When infusion is started for device 1, then the attribute clamp is true which means it is closed. This is because the open clamp is deferred until before starting the infusion. This is the interleaving that actually occurs in this particular scenario. Procedural salience for open clamp 1 depends on whether confirming time for 1 was the last activity which at this point is not the case. It is however the case that there are situations where infusion may commence with both clamps open with the same configuration as demonstrated by considering the following property that checks whether a path exists in which the processes can be completed for both pumps with the clamps open.

$$EF((inf1completed \ \& \ inf2completed) \ \rightarrow \ (oc1completed \ \& \ oc2completed))$$

One such sequence can be seen in Figure 4.

This raises the question as to what can be changed in the interactive system to foster the appropriate bias effecting a change in the salience of the relevant activities. One possible solution is to change the placement of prescription forms [2]. To capture this change the salience values would change making use of the attribute *close*, as mentioned above, that simply indicates whether or not the prescription form is close to the device. In the case where the form is located further away from the pumps (close is false), model checking shows that the relevant property (roller clamp is open when the infusion key is pressed) is satisfied for both pumps in *all* paths . This suggests that user bias towards the correct action (opening the roller clamp) is sufficiently strong in the modified interactive system.

CONCLUSIONS

Cognitive biases are clearly an important consideration in the design of an interactive system. Predictable failures arise in many interactive systems because the user takes a wrong choice or is otherwise confused by a misleading display. While there are many efforts to analyse interactive behaviour to design these aspects out before it is too late (Cognitive Walkthrough [15] and Usability Heuristics [14] are well known examples) the approach described in this paper allows the developer or analyst to be explicit about the assumptions that will affect use. It also supports exhaustive and repeatable analysis, something which is relevant in safety critical domains, but not guaranteed by techniques such as Heuristic Evaluation.

	40	41	42	43	44	45	46	47	48	49	50	51	52	53
device1.action														start
clamp	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
dm	mainmenu	mainmenu	mainmenu	mainmenu	mainmenu	mainmenu	mainmenu	mainmenu	mainmenu	mainmenu	mainmenu	mainmenu	mainmenu	dinfusing
m	hold	hold	hold	hold	hold	hold	hold	hold	hold	hold	hold	hold	hold	infuse
rate	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE
time	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE
vtbi	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE
device2.action		enter			confirm			openclamp			start			
clamp	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
dm	dtime	dtime	dtime	dtime	mainmenu	mainmenu	mainmenu	mainmenu	mainmenu	mainmenu	dinfusing	dinfusing	dinfusing	dinfusing
m	hold	hold	hold	hold	hold	hold	hold	hold	hold	hold	infuse	infuse	infuse	infuse
rate	FALSE	FALSE	FALSE	FALSE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE
time	FALSE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE
vtbi	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE
main.action	dropthrough	Tentertime2	setupround	dropthrough	Tconfirmtim	setupround	dropthrough	Topenclamp	setupround	dropthrough	Tstartinfuse	setupround	dropthrough	Tstartinfuse
cht1completed	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE
cht2completed	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE
ct1completed	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE
ct2completed	FALSE	FALSE	FALSE	FALSE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE
cv1completed	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE
cv2completed	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE
done	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	TRUE
et1completed	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE
et2completed	FALSE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE
ev1completed	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE
ev2completed	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE
inf1completed	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	TRUE
inf2completed	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	TRUE	TRUE	TRUE	TRUE
interleave	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
lastactivity	Xchoosetim	Xentertime2	Xentertime2	Xentertime2	Xconfirmtim	Xconfirmtim	Xconfirmtim	Xopenclamp	Xopenclamp	Xopenclamp	Xstartinfuse	Xstartinfuse	Xstartinfuse	Xstartinfuse

Figure 4. Paths where salience is taken account of

The approach differs from task orientated techniques (see for example [4, 10]) because it aims not to describe idealised interactions with the designed system but rather to enable exploration of how the system will be used. A similar goal is pursued by approaches based on psychological theories of human behaviour (see for example [7, 20]) which work by using a simulation of how users will interact with the system to make predictions about usability problems. Our approach can be used to explore *systematically* all possible behaviours of the system, and while it does not use a cognitive architecture, it captures a richer set of factors based on cognitive principles that might influence user behaviour (e.g. salience). How that influence is manifested, instead of being *calculated* by the cognitive architecture, must be considered during the modelling process, opening up space for discussion.

This work is in its early stages. However tools that make explicit use of models of cognition in the systematic analysis of interactive systems have the potential to be useful in safety critical domains. In such domains a guarantee is required that all interaction possibilities have been considered. Formal techniques, such as the techniques used here, are already in use. It is crucial however in these situations that the assumptions made in the underlying cognitive theory are explicit. The cognitive model is to be seen for what it is, a formalisation of a hypothesis about how the system might be used. This analysis may then be one of several analyses based on alternative models of cognition.

Acknowledgements

REFERENCES

1. Altmann, E., and Trafton, J. Memory for goals: an activation-based model. *Cognitive Science 26*, 1 (2002), 39–83.

2. Back, J., Cox, A., and Brumby, D. Choosing to interleave: human error and information access cost. In *Proceedings of the 2012 ACM annual conference on Human Factors in Computing Systems. CHI 12*, ACM Press (2012), 16511654.

3. BBraun. B.Braun infusomat space: instructions for use. Tech. rep., B. Braun Melsungen AG, 2005.

4. Bolton, M. L., Bass, E. J., and Siminiceanu, R. I. Generating phenotypical erroneous human behavior to evaluate human-automation interaction using model checking. *International Journal of Human-Computer Studies 70* (2012), 888–906.

5. Campos, J. C., Doherty, G., and Harrison, M. D. Analysing interactive devices based on information resource constraints. *International Journal of Human-Computer Studies 72* (2014), 284–297.

6. Campos, J. C., and Harrison, M. D. Interaction engineering using the IVY tool. In *Proceedings of the ACM SIGCHI Symposium on Engineering Interactive Computing Systems*, G. Calvary, T. Graham, and P. Gray, Eds., ACM Press (2009), 35–44.

7. Chi, E. H., Rosien, A., Supattanasiri, G., Williams, A., Royer, C., Chow, C., Robles, E., Dalal, B., Chen, J., and Cousins, S. The Bloodhound Project: Automating discovery of web usability issues using the Infoscent™ simulator. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '03, ACM (New York, NY, USA, 2003), 505–512.

8. Cimatti, A., Roveri, M., Olivetti, E., Keighren, G., Pistore, M., Roveri, M., Semprini, S., and Tchaltsev, A. NuSMV 2.3 user manual. Tech. rep., ITC-IRST, Trento, Italy, 2007.

9. Hutchins, E. *Cognition in the Wild*. MIT Press, 1994.

10. John, B. E., Prevas, K., Salvucci, D. D., and Koedinger, K. Predictive human performance modeling made easy. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '04, ACM (New York, NY, USA, 2004), 455–462.

11. Masci, P., Zhang, Y., Jones, P., Thimbleby, H., and Curzon, P. A Generic User Interface Architecture for Analyzing Use Hazards in Infusion Pump Software. In *5th Workshop on Medical Cyber-Physical Systems*, V. Turau, M. Kwiatkowska, R. Mangharam, and C. Weyer, Eds., vol. 36 of *OpenAccess Series in Informatics (OASIcs)*, Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik (Dagstuhl, Germany, 2014), 1–14.

12. Monk, A., Curry, M., and Wright, P. Why industry doesn't use the wonderful notations we researchers have given them to reason about their designs. In *User-centred requirements for software engineering*, D. Gilmore, R. Winder, and F. Detienne, Eds. Springer, 1991, 185–189.

13. Mori, G., Paternò, F., and Santoro, C. CTTE: Support for developing and analyzing task models for interactive system design. *IEEE Transactions on Software Engineering 28*, 8 (2002), 797–813.

14. Nielsen, J. Heuristic Evaluation. In *Usability Inspection Methods*, J. Nielsen and R. Mack, Eds. John Wiley & Sons, Inc., 1994, ch. 2.

15. Polson, P. G., Lewis, C., Rieman, J., and Wharton, C. Cognitive walkthroughs: a method for theory-based evaluation of user interfaces. *International Journal of Man-Machine Studies 36*, 5 (1992), 741–773.

16. Ramos-Hernandez, D., Fleming, P., and Bass, J. A novel object-oriented environment for distributed process control systems. *Control Engineering Practice 13*, 2 (2005), 213 – 230.

17. Rukšėnas, R., Curzon, P., Blandford, A., and Back, J. Combining human error verification and timing

analysis: a case study on an infusion pump. *Formal Aspects of Computing* (2013), 1–44.

18. Rukšėnas, R., Curzon, P., and Harrison, M. D. Integrating formal predictions of interactive system behaviour with user evaluation. In *Integrated Formal Methods*, E. Johnsen and L. Petre, Eds., vol. 7940 of *Springer Lecture Notes in Computer Science*, Springer-Verlag (2013), 238–252.

19. Suchman, L. *Plans and Situated Actions: The Problem of Human Machine Interaction*. Cambridge University Press, 1987.

20. Teo, L.-H., John, B., and Blackmon, M. Cogtool-explorer: A model of goal-directed user exploration that considers information layout. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '12, ACM (New York, NY, USA, 2012), 2479–2488.

21. Wright, P., Fields, R., and Harrison, M. Analyzing human-computer interaction as distributed cognition: the resources model. *Human-Computer Interaction 15*, 1 (2000), 1–42.