UNIVERSITY OF
NEWCASTLE

University of Newcastle upon Tyne

COMPUTING
SCIENCE

Time as a dimension in the design and analysis of interactive systems

M. D. Harrison and K. Loer

TECHNICAL REPORT SERIES

No. CS-TR-980 **August, 2006**

TECHNICAL REPORT SERIES

No. CS-TR-980 **August, 2006**

Time as a dimension in the design and analysis of interactive systems

Michael D. Harrison and K. Loer

Abstract

This paper discusses the relevance of timing to the design of interactive systems. It introduces a set of dimensions to assist the process of making appropriate time design decisions in interactive systems. Timing issues are associated with the dynamic behaviour of the design and with the information resources that may serve to control the pace of the interaction. Finally the paper considers two examples of system for which time design is appropriate and considers how a model of the system can provide assistance in design. The analysis of interactive properties of the systems uses a specification based on uppaal which enables the exploration of real-time properties.

Printed and published by the University of Newcastle upon Tyne,
Computing Science, Claremont Tower, Claremont Road,
Newcastle upon Tyne, NE1 7RU, England.

Bibliographical details

HARRISON, M. D., LOER, K..

Time as a dimension in the design and analysis of interactive systems
[By] M. D. Harrison, K. Loer.

Newcastle upon Tyne: University of Newcastle upon Tyne: Computing Science, 2006.

(University of Newcastle upon Tyne, Computing Science, Technical Report Series, No. CS-TR-980)

Added entries

UNIVERSITY OF NEWCASTLE UPON TYNE
Computing Science. Technical Report Series. CS-TR-980

Abstract

This paper discusses the relevance of timing to the design of interactive systems. It introduces a set of dimensions to assist the process of making appropriate time design decisions in interactive systems. Timing issues are associated with the dynamic behaviour of the design and with the information resources that may serve to control the pace of the interaction. Finally the paper considers two examples of system for which time design is appropriate and considers how a model of the system can provide assistance in design. The analysis of interactive properties of the systems uses a specification based on uppaal which enables the exploration of real-time properties.

About the author

Michael Harrison researches the analysis of dependable interactive systems, techniques for their assessment and to aid their design at different levels of rigour. He is also interested in the nature and structure of dependability arguments, particularly but not exlusively, in relation to interactive systems. His research is now focussed on ambient mobile intelligence, extending analysis techniques to systems that will dominate our future. In this regard he works within an interdisciplinary framework.

Suggested keywords

INTERACTIVE SYSTEMS,
TIME DIMENSIONS,
SPECIFYING INTERACTION

Time as a dimension in the design and analysis of interactive systems

M. D. Harrison[1] and K. Loer[2]

[1] Informatics Research Institute, University of Newcastle upon Tyne, NE1 7RU, UK
Michael.Harrison@ncl.ac.uk
[2] Department Strategic Development, Germanischer Lloyd AG - Head Office, Vosetzen 35, 20459 Hamburg, Germany
loe@gl-group.com

Abstract. This paper discusses the relevance of timing to the design of interactive systems. It introduces a set of dimensions to assist the process of making appropriate time design decisions in interactive systems. Timing issues are associated with the dynamic behaviour of the design and with the information resources that may serve to control the pace of the interaction. Finally the paper considers two examples of system for which time design is appropriate and considers how a model of the system can provide assistance in design. The analysis of interactive properties of the systems uses a specification based on uppaal which enables the exploration of real-time properties.

1 Introduction

Timing is an important and yet neglected feature of the design and implementation of interactive systems and in understanding their usability. People use technology to complete activities, to achieve goals, to judge whether to delay goals, and to do things imperfectly under time constraint. People's experience within an environment involving computer based systems may be affected by temporal concerns. For example, in the context of an airport information system, travellers may become anxious if the status of a flight is not updated often enough, or may become annoyed if they are reminded too often. Time requirements may relate to experience and may be subjective or requirements may be critical to the safety or security of a system and therefore objective. This paper highlights the importance of time in design and suggests ways that devices within systems may be designed to support timing aspects of the user's interaction more effectively. Finally the paper briefly illustrates the role that modelling techniques might play in helping designers to explore the timing implications of a design.

Time is an element of the *context* in which a device is placed that may affect its activity in a number of ways. It may be a factor to be taken into account in the performance of the device itself. Properly designed systems may help people juggle activities to use time most effectively, may enable them to predict episodes that are likely to involve high workload and deal with them accordingly, may keep the user in control while taking away those aspects of the activity that can be dependably automated. Systems may use timely feedback, or timely recommendations to allay the stress of not being sure whether some future event will happen at the appropriate moment.

The paper has two objectives. The first is to discuss how design is affected by time and to explore dimensions that help designers to account for time in design. Design of the whole interactive system may be improved by a proper consideration of time. There are a number of dimensions that may be important in a consideration of how time might impact a design. In sections 2 and 3 two notions are explored that are important to an understanding of the temporal capabilities of the system. In the first place dimensions relating to dynamic control are considered, then how information resources might affect the temporal progress of the system. Finally Section 4 explores the role that modelling may play in helping designers to identify timing issues in the design of a particular system.

2 Time dimensions for design

The success of an interactive system design is sensitive to its dynamic behaviour and how the user or users manage this behaviour. There are a number of issues or dimensions that are important to making the interactive system effectively These dimensions may be more or less important depending on time granularities, the nature of the context in which the device is situated, the training and capabilities of the user and may be categorised as: internal / external; subjective / objective; sporadic / continuous; user initiative / system initiative; extreme / normal.

Internal or external behaviour effects The pace or timing of an interaction may be driven by the activity of the device itself. As a result user actions may take time and affect the pace at which the user can achieve goals. Alternatively the environment of the device may drive the pace of the interaction. For example, incoming email messages or telephone calls can affect work-rate. These characteristics of the system may be important to how the task is carried out and therefore device design may need to take this into account. For example, the device might alert the user to the consequences of the external environment or help the user decide how a goal should be achieved given some external circumstance.

An internal / external timing dimension might lead to a number of design decisions. For example:

- Designing a text reader so that the first few pages are displayed to mask the delay involved in the computer reading the whole document. This is a design decision that is made in response to an internal timing effect. How necessary or effective such a strategy is depends on the activities of the user. If the user normally starts reading at the beginning of the document then this might be a good strategy leading users to a perception that there has been no delay. Hence an appropriate design strategy involves an understanding of delays in the system, generation of information and the way that the user operates.
- Providing a public display in an airport terminal that continually updates to show when a flight is to leave or is guaranteed to provide flight information with some specific delay after arriving in the hall. Here timing considerations in the external environment are important to the tasks being carried out by the user of the display.

> The currency of this information is important to the effective performance of the user's (passenger's) task. A similar but more direct purpose is achieved by a pedestrian traffic light that counts down to indicate how long before the light goes green. Both examples provide information about how much "slack time" is available in which the user can carry out other activity.

Subjective or objective effects Timing affects the *experience* the user has of the system by generating a sense of satisfaction or well being or by causing anxiety or a sense of hurry. Such experiences might be created without otherwise affecting the behaviour of the system. Such experiences are subjective. Those aspects of the system for which timing is externally important — for example, the system times out if the user fails to achieve an objective (entering a pin number) within a certain interval — may be said to be objective, although it is clear that this objective property will have subjective effects.

The subjective / objective dimension is more difficult to assess. Consider again the airline display. An objective requirement of this system would be that the information is updated within some maximum time interval, whereas a subjective requirement would be that the display should be updated sufficiently regularly for the particular passenger so that the anxious traveller may be sure that the information presented on the display is a sufficiently accurate picture of the system.

Sporadic or continuous effects Timing characteristics may be continuous. They affect the ongoing pace of the interaction. Alternatively effects might be sporadic — they happen in bursts where users have deadlines to meet with differing degrees of warning, and as a result may have periods of high workload. For example, the broadband connection may suffer high loadings that might be exhibited as a temporary cessation to the user's progress.

Here the design might enable items arriving sporadically to be buffered so that the user can deal with them in a more continuous way, or enable the user to batch items so that when the system is able to to deal with the actions they can be carried out. Another example is a control system such as an aircraft where automation automatically takes over from the pilot when the risk of the automation performing the actions are less than the pilot carrying out the actions — for example high performance manoeuvers in which an aircraft is inherently unstable.

Whether the system is driven by user initiative or system initiative The pace of interaction might be driven through the initiative of the user or through the activity of the device, or the surrounding context.

A system might be designed to enable the user to see what progress is being made toward the goal so that even though the user has the initiative, the system will provide information about the consequences of the user's pace. An example of such a system is a car trip computer that predicts time to destination at current pace. Alternatively the system may calculate that, according to current progress, the user will not meet a deadline and therefore carry out some of the processing even though the quality of the output may be more prone to error as a result.

Extreme or normal behaviour Timing effects may be normal and dealt with in the everyday activity of the user or something that only happens in extreme situations. In extreme situations unusual decisive action may need to be taken leading to changes in the pace of the interaction. These changes may be hard to deal with and may require new strategies or plans that are unfamiliar, and the device interface might provide assistance to the user.

Here an infrequently practiced procedure might be supported more directly by the interface than a routine procedure. Consider the support for standard operating procedures for normal operation in an aircraft cockpit versus recovery procedures that are rarely carried out.

Thinking about these design dimensions can help to understand how the system should be designed, and they may lead to a rationale for design decision.

Providing timely displays Conn [1] comments that time design should be considered in terms of the tasks to be carried out. Hence there may be temporal constructs that should be understood in the context of tasks.

delays the design may help the user to manage the delay by indicating its extent, by counting down, by visualising how much of the delay has been spent. As well as providing information about the delay itself, the device might help the user manage available slack time. For example an airport based information system might inform a passenger of the existence of a restaurant if a flight delay has occurred and there is sufficient time to have a meal.

deadlines in an interaction. It may not be enough to say when the deadline is but be more appropriate to help people manage the deadlines. Hence it might be appropriate to:
- give a prediction of whether the objective will be achieved at the present rate of progress;
- indicate how pace may need to be changed in order to achieve the deadline;
- offer to carry some of the workload to achieve it faster;
- indicate what kind of result will be achieved if current progress is continued.

pace of the interaction where the device may take over some of this activity if the pace is too slow.

deadlines the system may indicate explicitly to the user how to defer the inessential so that the essential can be carried out before the deadlines. Alternatively it may indicate how good the result is as the system continues to refine the result as the dealine approaches.

Conn's recommendations also relate to status information about the task's scope (how big is it), progress, how much of the execution is left. In relation to events and status information, Conn discusses the role that a "time tolerance window" plays in enabling an operator to assess for example the length of time an operator allows before deciding that a task is not making progress or that something must have gone wrong. Implicitly the information about tasks, status and events leads to support for decision making. To complete things in time appropriate human decision making may be required. The ability to predict how long something will take, to alert people to the likelihood of delay, to compromise or make realistic trade-offs, in order to be timely is critical to effective and satisfactory work practice. When activity supported by the device is task related, the appropriateness of different techniques for visualising and presenting this information depends on the kind of task that is being supported. Design depends on a number of effects, for example:

– Activity arrival rates and how predictable these are and the deadlines associated with carrying them out
– The user's or users' awareness of activity arrival, their control mode (see for example [2]) which gives an indication of how much time there is to reason about options and their awareness of activities
– How the activities that are external relate to system objectives, what resources are available and what their service rates are
– The pre-emptability of activities that have to be performed, whether it is feasible to combine activities, interleave them, postpone or drop them. The discretion that there is for satisficing and trading off between system objectives.

3 Resourcing time

People perform actions by using resources that are available to them in their environment. These resources might be information about the actions that are possible, information about goals, about plans, about the current state or about the effects that action might have [3]. Time design may be facilitated by identifying and analysing resources that have an impact on the temporal behaviour of the system. These resources play a role in shaping the interaction. The timing and availability of these resources may be a critical factor in the user's pace and understanding of deadlines and delays. In the following list, the impact of time is considered from a resource perspective.

– plans specifying actions to be performed and the order in which they are to be performed. It may be necessary to know whether the order of actions can be changed, whether it might be possible to drop actions and what the effect on the overall goal would be of dropping the action, whether there are alternative strategies that will produce more or less reliable answers depending on time pressure. While all these timing properties relate to performance against goal it may be reassuring to know about the progress that is being made in relation to the achievement of the goal.
– goals and sub-goals to be achieved: the effects of the states of the system that are to be attained. In terms of goals it may be appropriate to know whether a sub-goal is essential to the achievement of the top-level goal so that a decision can be made to drop it if appropriate. It may be helpful to know how long it will take to recover a sub-goal.
– knowledge of the current state of the world or interactive system to be used as a means of comparison with the goal state that is to be achieved. The current state may in fact differ from the state as represented by the device configuration because updates have not been sufficiently recent. As a result it may be difficult to anticipate what still has to be done and how long it will take to do it.

– historical information about process, actions and properties that have held of the state in the past; this may be in the form of a script of the last few commands as is the case of Unix or some description of previous landmarks or profile of how the current state was reached
– the effect that actions may have on the system or the world and how long these actions take, and how long they take to undo
– action possibilities that the system currently supports (including constraints on the interaction that limit what can or cannot be done) including indications of best strategy relating to the possibilities in the face of temporal concerns.

Resources may be used to control the plans of the user and to support the best strategy under time constraint. Resources may be used to indicate the immediacy of a deadline and the impact that actions may or may not have in achieving the deadline with an appropriate level of accuracy. Hence a plan following interface where the device is forcing the user down a path can take control away from the user and may affect the mode of control through the pace of the interaction. The operator may no longer have time to consider the activity nor the freedom to choose alternative courses of action.

4 Modelling time for design

4.1 Introduction

Modelling aims to allow designers to assess the implications of a particular set of design decisions. The concern in this paper is that models should be used to assess ways in which users of the system can make better use of timing aspects of the system. The paper has so far described and illustrated features that may be important from a modelling perspective: how time relates to the control of the system; how information resources relate to temporal properties of the system; how the system affords delay, pace, deadline etc. These features can be designed in a variety of ways and for a variety of purposes. Different kinds of model are most appropriate for the modelling of different features of the design. In this paper a particular modelling technique is used. Other more quantitative styles of modelling would be appropriate for assessing typical queueing behaviour for example, the identification of appropriate strategies for dealing with steady state behaviour.

To illustrate possible techniques, two examples are explored in more detail. The first is a system that involves operator control of a dynamic process (a paint shop). This activity is taking place at a time granularity that involves seconds rather than minutes. The system offers the user opportunity to automate some of the activities that have to be carried out. The system can break down from time to time, the paint guns wear out and require replacement. Options are provided for the operator to replace or repair parts. These two options have different costs associated with them, replacement is immediate but costs money while repair is free but costs time. Modelling is used here to explore what would be the optimal strategy in normal circumstances, what would be most robust to variation and what alternative strategies would be appropriate in different extreme circumstances. This analysis can be used to assess whether the interface to the operator is appropriate. The second analysis is concerned with the deployment of service information within a built environment, in this example an airport. The important activity from the user perspective works at a time granularity that is in terms of minutes rather than seconds. The system uses public display and hand-held device. Here issues associated with the timeliness and freshness of deployed information are important. An example property that may be checked of the model is that relevant flight information should be available to a passenger within one minute of arriving in a new space.

Using modelling techniques, processes can be specified to describe the physical characteristics of the environment, assumptions about the user and features of the interactions with the device that is being designed. These processes can, in particular, be used to capture temporal characteristics of the external environment, strategies including temporal strategies relating to the user's behaviour, features of the artefact that are important from the perspective of interaction with it.

The uppaal tool [4] is used here. It allows the analysis of networks of linear hybrid automata with clocks whose rates may vary within a certain interval. Thus it is possible to take different temporal reference systems into account – for instance, the real-world frequency of items on the belt and the operator's perception of the frequency under varying workload. Automata may communicate either by means of variables (which are global) or by using binary communication channels. Messages can be passed and synchronised correctly by employing patterns that provide the basic discipline for using channels and shared variables to synchronise and communicate. Some of these patterns are described in the uppaal tutorial [5]. Communication occurs as a result of two process synchronisations using receiving actions `a?` and sending actions `a!`. Guards are used to describe the circumstances in which communications can take place. Automata may be guarded by conditions involving clocks that can be used to represent delays or time invariants. It is not within the scope of this paper to describe the syntax and semantics of uppaal in detail, however the examples given below should be sufficiently clear to give the spirit of the approach. Although the expressive power of the notation has some limitations, the system has the advantage of easy availability using a graphical interface and this makes the model more accessible to non specialists. Uppaal provides tools for the simulation of systems — the state transition diagrams are animated, and the inter-process communication is displayed as an animated message sequence chart. The tool also supports analysis by state exploration. Models used in model checking can only label input and output resources. A key issue in developing these models is to minimise the number of states by making appropriate abstractions so that analysis can be performed.

Simulation or checking of the model generates state sequences. They can be generated through a simulation facility. By means of user intervention, sequences can be created and explored which form the sequence of actions that provided the bones of the richer narrative description that had previously been gathered through a process of user elicitation. Scenarios may be gathered by interviewing users of the system to be replaced or may be produced as a result of some kind of experimental evaluation of typical or emergent user strategies. Alternatively sequences may be generated by asserting properties that can be checked by the model checker. By this means it could be possible to discover the path that satisfies or contradicts properties such as the sequence that would take the least time or the strategy that would minimise loss given failure of a particular component of the system — here loss might result from delay in carrying out some action. Once paths are generated further analysis can be carried out.

This approach is discussed in more detail in Campos and Harrison [6] and Loer and Harrison [7]. It is intended that domain experts or human factors experts consider the implications of a sequence generated through the modelling and checking process. They use their expertise to envisage a situation, a context, in which the sequence might occur thereby creating a narrative based on the sequence. Loer and Harrison [7] explore property templates or patterns of properties aimed at making the process of generating appropriate properties to create these sequences more intuitive. Their tool provides an interface that allows the instantiation of CTL properties to a model based on usability heuristics [8]. Thus the process of analysis may be made easier for designers. Reachability is of particular interest in the context of timing properties. Thus the system can check whether significant states (states that have been designated as goal states) may be reached within a timescale subject to given constraints. In the finite state model of the environment, end states can be associated with reaching significant points in the process, for example if a physical model of the process is involved then relate a state to the completion of that process.

Alternative paths may be explored by adding additional constraints to the property to be checked. Further analysis might involve the manual annotation of action sequences with the information resources that have been described as part of the design but can only be hinted at in the state labelling captured in the model (see [9]).

A key to effective modelling is to find appropriate abstractions without biasing the analysis. Device models at different levels of abstraction capture key characteristics of the interactive system. For example a flight management system in [6] is explored in the context of a simple model of airspace trivially capturing notions of ascent and descent. A particularly important concern here is to produce small models of the salient features of the physical environment which is itself a complex and continuous environment. The particular concern here was to provide sufficient detail to explore mode issues associated with the design of the device. On the other hand, in [10], two context models constrain the behaviour of a handheld device. Here a sewage plant model characterises the behaviour of tanks, pipes, valves and effluent contained in the tanks and transported by the pipes. An additional model captures the spatial position of the hand-held device.

A human factors expert can consider alternatives to assess the implications for the design of the system and, possibly as a result, work with the designer to produce an interface that has better timing characteristics.

4.2 The paint shop study

The paint shop study is an example of a system that involves the control of dynamic processes. Indeed it has been used as a “microworld” experiment to explore how users devise strategies in the face of real-time constraint [11]. The design issue here would be to explore what device interface would provide most support for the user in the face of varying levels of workload as well as to support the decision processes required when choosing between repair and replacement strategies in the face of paint station failure.

These strategies might include decisions in relation to postponement, interleaving, synchronisation, speeding up or slowing down of function servicing and whether control should be automatic or manual.

A number of questions then relate to the design of the artefact — how an appropriate strategy can be communicated appropriately in the interface, whether appropriate strategies or next actions can be adequately resourced [3].

The system involves a conveyer belt that transports boxes to two parallel paint stations (Figure 1) for painting. Boxes may enter the paint system at different rates and a financial reward is given according to the number of boxes painted. When boxes ar-

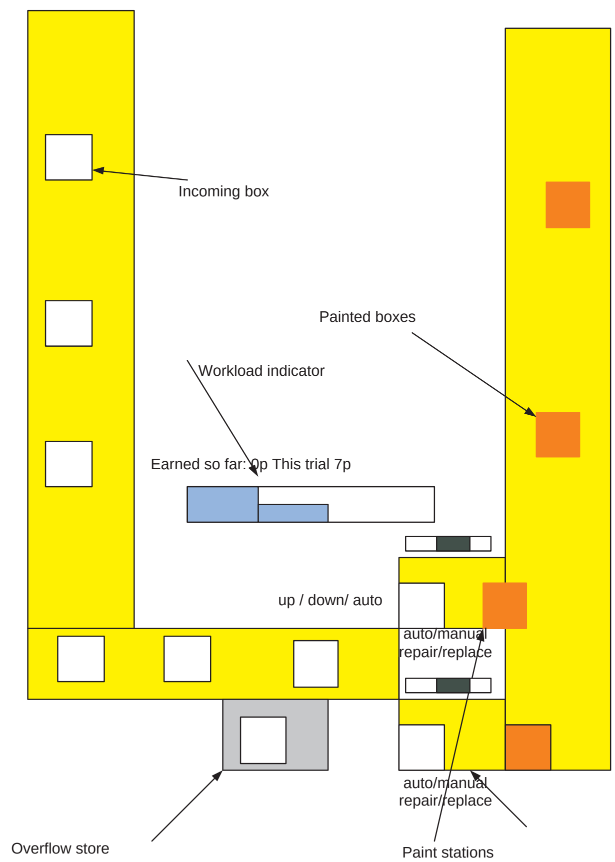

Fig. 1. The paintshop system

rive at a distribution lift, the user can then choose to make allocation to paint stations automatic or can press the “up” and “down” manual buttons as appropriate. The painting process can either be set to automatic mode (which is the default) or to the manual mode. In automatic mode, the paint station will automatically specify the number of coats to be painted, carry out painting and wait for it to dry. The rate of paint flowing through the nozzles is displayed just above each production line. The flow rate may decrease if nozzles become blocked or increase if the nozzle leaks thereby providing information about the future potential for replacement or repair. To paint an item manually, the operator has to click on a box and keep the mouse button pressed for a specified period of time to complete the process. If the mouse button is released before the minimum paint time the box is not painted and a spoiled box is released.

In the model, painting takes five time units in the automatic case and two time units in the manual case. When a nozzle ceases to function properly it can be repaired or replaced. Replacing a nozzle incurs no time cost but does incur a certain monetary cost. Repairing the nozzle incurs no monetary cost but causes a delay before the nozzle can be used again. In the micro-world experiments the cost and time variables were manipulated and indeed in the model also these values can be manipulated. Depending on the rate at which boxes arrive at the station and the state of the nozzles and the strategy used to employ the paint stations a certain proportion of the possible boxes will be painted. Boxes can fail to be painted either because the appropriate procedure has not been carried out inside the paint station or because the queue of boxes waiting to be painted exceeds a certain number. When the queue waiting to be processed exceeds some number, boxes are lost down a reject shute.

4.3 Modelling paintshop

The paintshop is modelled as seven concurrent processes with the aim of identifying what the optimal strategies under different constraints are and how the design may be changed to make it easier for the operator to adopt a better strategy. Thus it is possible to express and check for reachability properties such as:

1. "Is it possible to reach a state where the clock x is greater than 20"
2. "Is it possible to reach a state where all boxes have been painted?"

The result of checking the property is a path that can then be explored in more detail. Two of the models (Figures 2 and 3) describe the physical environment of the system. A dispatcher automaton (Figure 2) captures the regular distribution of boxes defined by a constant (`workload`) that is used to describe the workload level. This can be changed to explore different workloads. This process dispatches objects to the incoming queue. Figure 3 specifies the behaviour of a box being channelled through the two paint stations. This automaton models the part of the system containing the queue of boxes waiting to be serviced by the paint stations, as well as the lift that causes the boxes to be moved to one paint station or the other. It also models a repository for boxes that have fallen off the end of the queue and therefore lost. The final physical element (not illustrated) models the belt of finished items. When `workload=2` a new box arrives on the belt every two units (which is high workload). Values representing a medium and low workload are 3 and 4 respectively. In order to reduce the complexity of the analysis, the number of boxes in the model is limited to 10. While it is acknowledged that this is a great simplification in comparison to the continuous flows a real-world operator is likely to have to deal with, for the purpose of this analysis the simplified model is sufficient.

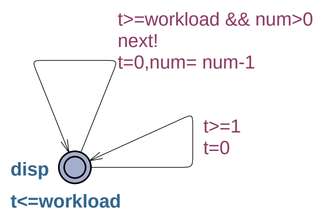

Fig. 2. High workload incoming belt

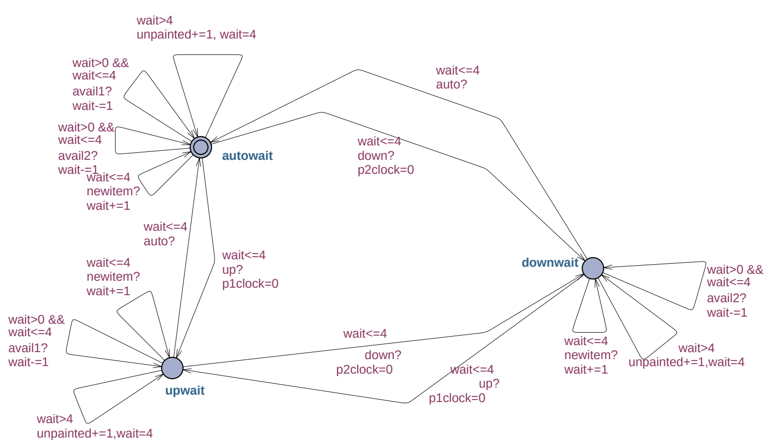

Fig. 3. Boxes waiting to be channelled

The device design is captured by the process in Figure 4. This describes the behaviour of the button that can be used to change from manual to automatic delivery to paint stations, as well as the feature that enables automatic or manual paint delivery and the mechanisms for repairing or replacing. There are two instances (station1 and station2) of this process that describe the behaviours of the two paint stations. The description of automatic and manual operation is contained in the top and bottom part of the automaton respectively. The automaton also captures fault occurrence and repair and replace costs. The severity of faults increases over time. A nozzle may break as soon as two items are painted but it will break for sure once four items are painted. Repairs cost 24 time units (see locations repairingA and repairingM), replacing a nozzle costs four tokens.

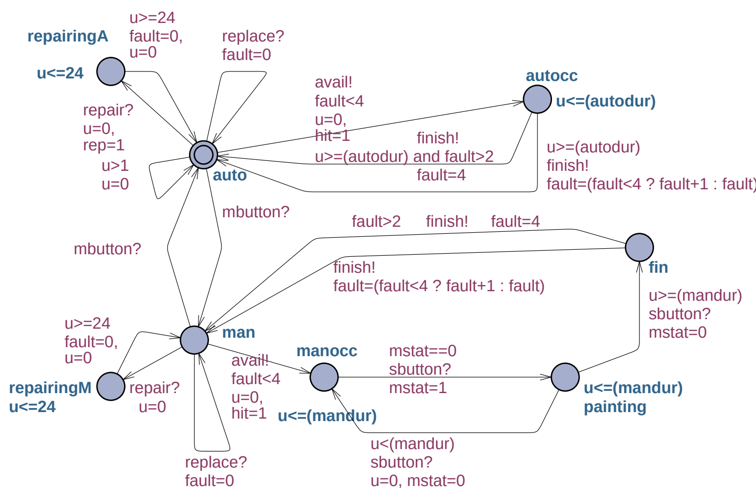

Fig. 4. The paint station automaton

The remaining modelling problem is to describe the alternative assumptions that are being made about users Two processes are designed to reflect what the user does. The first model (Figure 5) carries out a number of actions. It dispatches conditional user inputs and models simple repair/replace decisions: if the fault (variables p1fault and p2fault) is bigger than 3 and sufficient funds (variable win) are available, replace a nozzle, otherwise perform a repair.

The second user process (Figure 6 implements a random strategy. This process dispatches unconditional user inputs that are consumed by other processes ("monkey at the keyboard" style) but only generated when no internal synchronisations can be performed. Using the two models that have been described it can be used to explore a variety of properties that provide constructive information about the appropriateness of the design of the device.

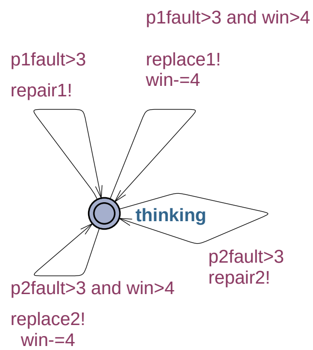

Fig. 5. Process implements a simple repair strategy

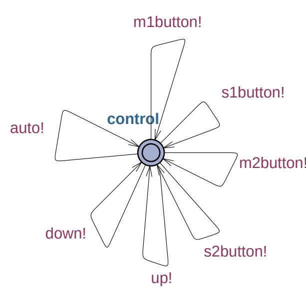

Fig. 6. Process which implements a random user

Reachability of system goals Analysis proceeds experimentally by exploring a number of properties:

P1: Can all n items be painted?

The property ("`E<> painted==n`") is true for $0 \leq n \leq 10$. When the negated property (here, the never-claim `A[] painted!=n`: in other words "n items can never be painted") is checked, the model checker produces a trace that can be simulated. Stepping through that trace, the analyst is guided through a scenario where both manual and automatic mode of painting are applied. The simulation and the sequence chart provided by uppaal can point to simple flaws or instances of unexpected behaviour of the model. In order to obtain a broader understanding of the reasons behind flaws, additional traces of similar instances are required. However, the tool only produces a single trace for each property. Additional traces, focussing on different aspects that may be considered contributing factors to a discovered problem, require a refinement of the property. For instance:

P2: Can all n items be painted, using only a single paint station?

The verifier only explores paths that involve a single instance of the paint station process. This is achieved by temporarily modifying the paint station specification so that only one of them used.

Finding minimal durations under different conditions Reachability properties may be further elaborated by considering how long it takes to reach a given state. In these cases it is necessary to explore alternative possible times to find the actual duration associated with an activity.

P3: Can all n items be painted in m time units, using only a single paint station?

This can be expressed as "`E<>(painted==n and stationUsed==0 and gtime==m)`" This property was checked for different values `m` of a global clock `gtime`. By repeated analysis the propery is satisfied for 22 units for the ten items, but the nozzle needs to be replaced at least twice, so the win is only two units. In the same way the following property can be explored:

P4: Can all items be painted in m time units, using both paint stations?

Again, the minimal duration is 22 time units. However, while the execution time is the same, in this case only one of the nozzles needs to be replaced, so the monetary win is six units.

All the traces captured by these properties confirm that the fastest way to perform the work is to opt to paint manually. To consider the design of the automation, the temporal effect of an automatic strategy was considered.

P5: What is the minimal time required to paint all items automatically?

A similar temporary modification to the one described above preventing the manual mode was performed in order to explore this property. The minimal time required to paint all items without manual intervention and by using both stations is 29 units.

Analysis of this kind has yielded the following observations that can be made use of in the design of the interface to the device:

1. Painting items manually is faster than automatic painting.
2. Using both stations does not necessarily gain a time advantage over using a single station only.

3. However, using both stations can save repair costs if the operator is prepared to take the risk and leave one station broken.

The analysis is described more fully in [12]. The temporal properties of this stage could have been calculated simply by using a numeric model of the processes. However, the additional effort of creating the uppaal model pays off when multi-valued decisions are considered, as a focus on monetary costs demonstrates.

Focussing on monetary costs So far the analysis has only been concerned with temporal costs and effects. Further properties can be used to check temporal and monetary costs associated with replacing faulty nozzles.

P6: Can all boxes be painted without losing money?

This property forces a search strategy where nozzle replacements are avoided. The resulting trace demonstrates that the task can be completed in 50 time units. The simulation demonstrates that both stations are used to paint in automatic mode until they break; then one station is repaired.

P7: What is the shortest time for painting everything without losing money?

The analysis yields that best performance (finishing the task in 44 time units) can be achieved, and the new trace suggests that this performance can only be achieved if manual control is opted. Again, both stations break, but the trace indicates that only one station needs to be repaired.

P8: Can all items be painted without losing money, using only one paint station?

This analysis is dual to P6, but focussing on a modified specification so that there is a single paint station only. This property is concerned with the robustness of the system and the additional temporal costs. The strategy exhibited by the model-checking trace could be used by an operator who does not have time pressure and therefore aims at maximising the win.

Analysing the durations under the assumption that temporal costs are secondary to monetary costs reveals again that the best possible performance can be achieved by using both stations in manual mode, but the required duration increases to 44 units. The results produced provide an indication of what a good operation strategy might be under extreme conditions with respect to temporal and monetary costs. However, it remains the task of the system designer to resolve if any of these strategies are suitable, and if they should be implemented as part of the system or as part of the operator training. For an informed decision it also remains necessary to draw on human-factors experience. A crucial additional factor that will influence this decision is the operator workload.

Variable workload The analysis of performance has assumed a constantly high workload, given by the dispatcher model in Figure 2. The analysis can be repeated using increasing, decreasing or alternating workloads in order to collect insights about further strategies.

4.4 An airport ambient system

By way of contrast the second example involves the exploration of an ambient and mobile system as it would appear in a built environment. Here the timescale is at the minutes rather than seconds level because it concerns what a passenger would perceive as urgent or immediate in the context of the environment. This system allows access to services, either global services or services that are specific to the environment (for example, passenger information about travel delays or the status of a flight or retail service information). This information might be invoked and deployed to hand-held devices as well as being displayed in a suitable format on public displays. To get a sense of the style of the system that is to be modelled and analysed consider the following scenario based on an airport.

On entry to the check-in hall, a sensor recognises the passenger's electronic ticket and therefore subscribes her to the flight service for which she is booked. Her context will be updated with current location, namely the check-in hall. The flight service immediately sends information about flight progress to her hand-held that contains queue, gate and delay information. The queue that she is notified about ensures the shortest waiting time. Queue messages are also sent to the public display in the check-in hall so that passengers have an alternative source of information. When the passenger enters the queue a sensor detects her presence and adds this locational information to her context. As a result of this change of location, information about queues are no longer sent to her hand-held device but instead she receives messages that relate to the length of queue and the predicted waiting time. The passenger is meanwhile subscribed to a seat booking service specific to the flight that enables her to book the seat she likes while waiting. The service helps the passenger choose based on her preference information.

Properties of concern would depend on a requirements elicitation but would feasibly provide the following: "The system should ensure that flight information relevant to all passengers contained within a space is supplied by posting that information on the displays of that space". In general properties would concern: the resilience of the whole system, including temporal characteristics; the usability of the hand held devices and public displays; how effectively and immediately information is provided that enhances the experience of users of the system. In the last case timing issues may be critical to achieving this experience and a sense of place in the built environment. Properties of concern here will include "however many services a user subscribes or is subscribed to, the flight information service will be dispatched both to the user's device and to the local display within a defined time interval" and "any facility that is offered to a subscriber will only be offered if there is a high probability that there is enough time to do something about the facility offered."

In reality the development of ambient mobile intelligence applications might be carried out without complete foreknowledge of the platforms and software that will be running on these platforms. The physical characteristics of alternative platforms may be important in contributing to the experience of place — frequent flyers may use smart phones, large plasma screens may be placed in the space in a number of different ways. The advantage of using walkthrough techniques is that early exploration may be carried out before the platform is decided and may assist an understanding of whether a particular configuration is appropriate.

4.5 Modelling the airport

To model this system some gross simplifications are made to indicate the style of analysis. The description captures the characteristics of the airport system by modelling sensors for each space in the built environment, the single dispatcher that is designed to distribute messages and a token passenger that captures the properties of all the passengers in the environment.

The dispatcher (Figure 7) simply distributes messages with tags associated with flight number and location. In the version described here it distributes these at random at different time intervals depending on the value of workload. The rate of distribution can be adjusted to assess the properties of different rates of distribution.

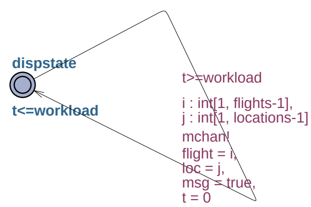

Fig. 7. The dispatcher process

The distributed characteristics of the airport system are captured by three further models. Two of these represent different types of sensor, one relates to the queue in the checkin area while the other is a more generic sensor (Figure 8) that captures the location of the different areas of the airport while at the same time receiving messages from the distributor. These messages are filtered according to location. They are marked as displayed on the public display if the sensor has received requests from passengers with flight details that match the flight tag. In other words the public display is modelled so that it only displays those messages that are relevant to the passengers in the area.

The other sensor is designed to update its display with information about its queue based on the longest wait. The non-queueing sensor checks the location and status of a passenger that communicates with it. Depending on the flight number and whether the passenger has just entered the location or is about to exit it, the count for the flight is decremented or incremented. When a message is received from the dispatcher for a particular flight it is only dispatched if it is valid for the location and there are passengers in the location that relate to the flight — this is done by the `read()` function.

The passenger (Figure 9) checks its location by communicating its flight number to the sensor. When the passenger initialises it absorbs a strategy for navigating through the building by simply updating an array `path`. It receives messages from the dispatcher and displays them on the hand-held only if the location and flight match the passenger in the distributed message.

In this example rather than check properties, analysis was carried out using the simulation facilities associated with the uppaal model checker — exploring when infor-

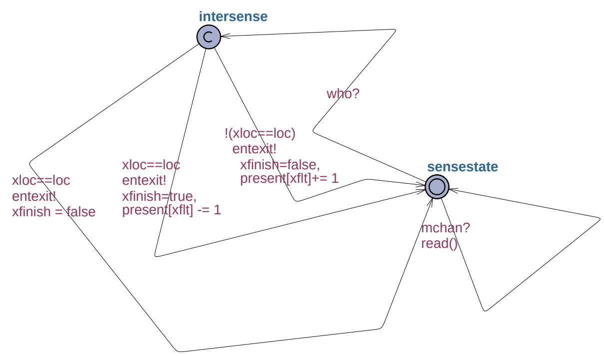

Fig. 8. The non-queueing sensor

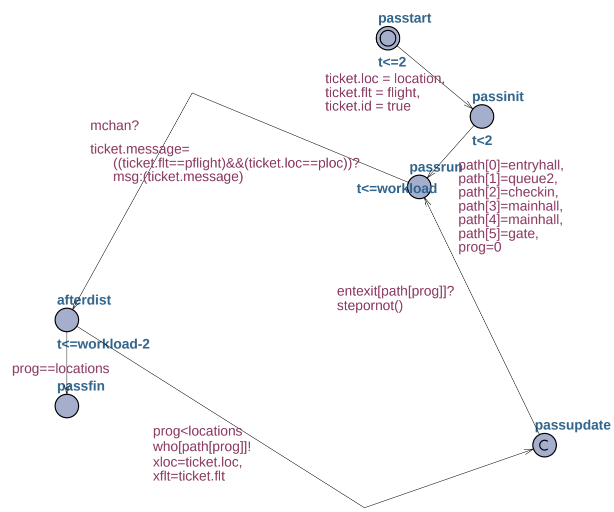

Fig. 9. The passenger

mation would be deployed that was relevant to the passenger either directly to the hand held or to the public display.

5 Conclusion

This paper has identified features in the design of an interactive system that may be influenced by timing characteristics and has provided a preliminary illustration of how modelling techniques might assist the exploration of the design. It contains a challenge — features of a design in response to timing considerations are rarely considered and yet have an important impact on the usability and the experience of the system design. Within the context of time orientated design the paper makes a number of observations, for example the fact that temporal issues may be external, a feature of the environment, or may be internal and perceived rather than actual. It may therefore be necessary to explore external cognitive resources which the user can offload to so that perceived workload may be reduced.

While modelling techniques such as uppaal that are relatively familiar to computer scientists can be used to explore the consequences of a design there are a number of challenges that must be overcome before these techniques become practically feasible.

1. dealing with the state explosion that is associated with adding temporal constraints
2. in the case of the ambient and mobile systems, dealing with the fact that there may be many processes with similar properties (such as passengers) that must be incorporated within the model without exploding the specification so that it is impractical
3. finding appropriate languages or patterns of use of the modelling techniques to capture notions of location and context in a way that makes it easier for implementers to use these techniques without serious and unnecessary overhead.

References

1. Conn, A.: Time affordances: the time factor in diagnostic usability heuristics. In: Proceedings of the SIGCHI conference on human factors in computing systems, ACM Press (1995) 186–193
2. Hollnagel, E.: Cognitive Reliability and Error Analysis Method – CREAM. Elsevier (1998)
3. Wright, P., Fields, R., Harrison, M.: Analyzing human-computer interaction as distributed cognition: the resources model. Human-Computer Interaction **15** (2000) 1–42
4. Larsen, K.G., Pettersson, P., Yi, W.: UPPAAL in a Nutshell. Int. Journal on Software Tools for Technology Transfer **1** (1997) 134–152
5. Behrmann, G., David, A., Larsen, K.: A tutorial on uppaal. In Bernardo, M., Corradini, F., eds.: Formal methods for the design of real-time systems. Number 3185 in Springer Lecture Notes in Computer Science. Springer-Verlag (2004) 200
6. Campos, J., Harrison, M.: Model checking interactor specifications. Automated Software Engineering **8** (2001) 275–310
7. Loer, K., Harrison, M.: An integrated framework for the analysis of dependable interactive systems (ifadis): its tool support and evaluation. Automated Software Engineering (2006) in press.

8. Nielsen, J., Mack, R., eds.: Usability Inspection Methods. John Wiley & Sons, Inc. (1994)
9. Campos, J., Doherty, G.: Supporting resource-based analysis of task information needs. In Harrison, M., Gilroy, S., eds.: Proceedings 12th International Workshop on the Design, Specification and Verification of Interactive Systems, Springer Lecture Notes in Computer Science (2006) in press.
10. Loer, K., Harrison, M.: Analysing user confusion in context aware mobile applications. In Constabile, M., Paternò, F., eds.: INTERACT 2005. Number 3585, Springer Lecture Notes in Computer Science (2005) 184–197
11. Hildebrandt, M., Harrison, M.: Putting time (back) into dynamic function allocation. In: Proceedings of the 47th Annual Meeting of the Human Factors and Ergonomics Society, Denver, CO (2003) 488–492
12. Loer, K., Hildebrandt, M., Harrison, M.: Analysing dynamic function scheduling decisions. In: Human Error, Safety and Systems Development, Kluwer Academic (2004) 45–60