Reasoning about Time in Dynamic Information Displays*

J.C. Campos (1) and G.J. Doherty (2)

(1) Departamento de Informática, Universidade do Minho, Campus de Gualtar, 4710-057 Braga, Portugal. e-mail: Jose.Campos@di.uminho.pt

(2) CLRC Rutherford Appleton Laboratory, Chilton Didcot, Oxfordshire, OX11 0QX, U.K. e-mail: G.J.Doherty@rl.ac.uk

Abstract

With increasing use of computing systems while on the move and in constantly changing conditions, whether it is via mobile devices, wearable computers or embedded systems in the environment, time plays an increasingly important role in interaction. The way in which information is represented in an interface is fundamental to interaction with it, and how we use the information in the users tasks and activities. Dynamic representations where the user must perceive changes in the information displayed over time pose a further challenge to the designer. The diminutive size and limited display capabilities of many ubiquitous computing devices further motivate careful design of these displays. In this paper we look at how time can be taken into account when reasoning about representational issues from the early stages of design. We look at a model which can be used to reason about these issues in a structured fashion, and apply it to an example.

1 Introduction

In this paper we look at how time can be taken into account when reasoning about issues of representation from the early stages of design. Time plays an important role in the "interaction experience" [14], and as such should be considered when designing interactive systems. Issues such as mapping actions to effects, causal reasoning, prediction of future states of a system where continuous dynamic information is involved, all must be addressed with time considerations in mind.

Reasoning about design, in the context of interactive systems, implies reasoning about how useful and easy to use those systems will be. In this context, assessing the quality of a design is no easy task. Although there are a number

*Published in G.J. Doherty, M. Massink and M.D. Wilson, editors, *Continuity in Future Computing Systems*, volume RAL-CONF-2001-001 of Conference Proceedings, pages 80-85. Council for the Central Laboratory of the Research Councils. 2001.

of human-factors oriented studies which have resulted in design guidelines and rules, these cannot be turned directly into a set of properties that all systems must obey. In a specific design context, whether a guideline is applicable or not is always debatable. It might even be the case that a guideline is wrong [21]. This is especially true when designing for novel interaction techniques and paradigms.

Issues of representation are fundamental in what we perceive and the way we think and solve problems [12]. The increasing use of novel physical form factors is likely to increase the importance of external representations in information technology applications [24]. In this paper we concentrate on issues of representation and time. We propose a model to reason about representational issues where time is involved. We then show how this model can be used to reason about a dynamic information display representing a (variable) information transfer rate. This work follows from previous work on representational reasoning in [7] and [6].

2 Information Representations and Time

The user interface of a system should help users understand and control the system's state and behaviour, or that of a process manipulated through the system. In this context, the perception the users have of both the state the system is in and of how the system behaves is vital. The user interface must provide an accurate and understandable representation of the system's state, and provide clues towards what the effect of the various actions that are possible will be.

Time plays a relevant role in both how systems are used and perceived. In [9] it is discussed how small differences in the time it takes for a user to perform different actions can influence the interaction strategies adopted. The time the system takes to react to user input can also have a great impact on the usability of the system [5, 4]. If the system's response takes too long, it will be difficult for the user to establish a causal relation between actions and their effects.

Time plays an even more relevant role in emerging interaction techniques and paradigms. Mobile and wearable devices operate in environments where conditions are continuously changing, but sensing and adapting to changing context, and accessing remote resources may take time [13]. In mobile applications, parameters such as quality of service, or even intermittent provision of service, have a great impact on interaction, and the time dimension is involved when studying how to design such interfaces. In addition to the use of novel input devices, multimedia and augmented reality systems usually involve continuous dynamic information being presented to the users.

Another relevant area is that of supervisory and manual control, where users operate or control a system in real time. Examples of analysis of this kind can be found in [22] and [3]. In both cases the pilot interaction with the automation of an aircraft is analysed regarding the control of the aircraft's climbing behaviour. Both approaches use model checking-style tools which are discrete in nature. [3] discusses how continuous systems can be analysed by a discrete tool using abstraction. In [8] the use of hybrid automata for the specification and analysis of continuous interaction is discussed. This type of approach allows for a more direct representation of the continuous aspects of interactive systems. All the approaches above tend to concentrate on behavioural aspects of the system, rather than the representational aspects.

Reasoning about design, in the context of interactive systems, implies reasoning about how useful and easy to use those systems will be. Usability evaluation methods can be divided into two groups (see [16, Part III]):

- Empirical — these rely on building prototypes of the system being developed and testing them using real users under controlled conditions.
- Analytic — these rely on confronting models of the system with how users are expected to behave.

Factors influencing the usability of a system range from pure software engineering concerns to psychological, sociological, or organisational concerns. Hence, in most situations, only actual deployment of the system will give a final answer towards its usability. It is somewhat paradoxical, therefore, that in some situations, namely where safety-critical systems are involved, empirical evidence might not be enough to support claims about the usability (and safety) of a system. Moreover, empirical methods are expensive to apply, and require a lot of organising and time. Additionally they are difficult to apply in the early stages of design when most decisions have to be made. Analytic methods fill this gap. By being based on models of the system, and not requiring a prototype to be built and placed in a plausible interaction context, analytic methods have the potential to allow reasoning about the usability of a system to be carried out early in the design process. Their drawback is that assumptions have to be made about user behaviour, and the results can only be as good as the assumptions that are made.

Traditional analytic methods tend to be carried out manually and more or less informally [16]. This can work in areas where the technology is well known, and the design of the system is not too complex. However, as new interaction techniques and paradigms emerge, or as systems become more complex, this type of approach becomes less likely to deliver.

The use of structured models which are amenable to rigorous analysis during development can impact system design at two levels:

- the use of rigorous mathematical concepts and notations can help in the organisation and communication of ideas;
- mathematical models can allow rigorous reasoning about properties of the system being designed.

Although the modelling process itself can provide valuable insight into the system being designed, the possibility of formally reasoning about the models can bring their use to full fruition. Hence, in recent years formal verification of interactive systems has been an active area of research (see, for example, [15, 2]).

Traditionally, quality will be measured against a set of properties that the system or artifact must exhibit. Trying to devise a meaningful set of properties, that should be true of an interactive system in order to guarantee its quality, is no easy task: there is no magic recipe for easy interactive systems building. Guidelines are (or, at least, should be) of a qualitative and high level nature, which means that they are not easy to verify in a rigorous way. They must first be turned into concrete properties. A typical guideline might be: "*Reduce* *cognitive load*" [20]. Design rules, on the other hand, are about very specific interface features, which means that formal verification might not be, in many cases, the best approach. A typical design rule might be: "*A menu should not have more than seven entries*". In a specific design context, whether a guideline is applicable or not is always debatable. It might even be the case that, given a specific design context, trying to make the system comply to some general purpose guideline might be detrimental [21]. This is especially true when designing for novel interaction techniques and paradigms, which were not considered when devising currently available guidelines.

In the field of software engineering, lists of properties have also emerged (see, for example, [23, 11]). However, these tend to be governed mainly by the specific style of specification being used, and what that style allows to be expressed. Their relevance towards usability is not always completely clear. In [1] and in [19], sets of properties of interactive systems that can be checked automatically are proposed. Again these seem driven more by the capabilities of the verification formalism used than by pure methodological concerns.

Contributing to this problem is that fact that interactive systems form an increasingly heterogeneous class of systems. In fact, the only common requirement is that the system interacts with a human user effectively. This accounts for systems from airplane cockpits and control rooms of nuclear power plants, to mobile phones and set-top boxes. The starting point is simply that the system will have a user interface. Additionally, the presence of the human user means that assumptions must be made about the capabilities (physical, cognitive, and perceptual) of the particular users the system is intended to serve. So, instead of trying to establish a list of properties, we should try to identify some specific issues related to interactive systems design, about which the designer might wish to be assured that the system is satisfactory. We do this by paying attention to what is generally true of all systems and all users.

We start with a a model proposed in [7] for reasoning about representational issues. This model provides a means for integrating representational reasoning into the design process. In [6] we show how the model can be used to reason about the properties of the interaction between a pilot and a cockpit instrument during the landing procedure. The analysis was done using a theorem prover (PVS [18]), and allowed us to uncover a number of assumptions about the systems that were embedded in the representation of information in the interface. By taking such a rigorous approach to the analysis of representational issues, our goal is to provide both a framework for reasoning about representation, and confidence that reasoning over the abstract design of the system holds at the presentation level.

[2] expands the model by introducing a more explicit account for actions. This enables a more explicit consideration of issues such as the relation between user goals and the actions provided by the interface. Considerations about time, however, are still not present. In the present paper we go a step forward and see how time can be incorporated into representational reasoning in order to enable analysis of the impact of time related issues in the perception the users have of a system.

3 A Framework for Analysis

Rather than conjecture about users internal representations of time, we wish to provide a framework which illustrates qualitative differences between design alternatives. We focus on representations of information in the interface which change over time, where this change is relevant to the users tasks and activities.

We start with a reworked version of the model proposed by [7] (see Figure 1). From the diagram in Figure 1 it is possible to identify the three basic entities

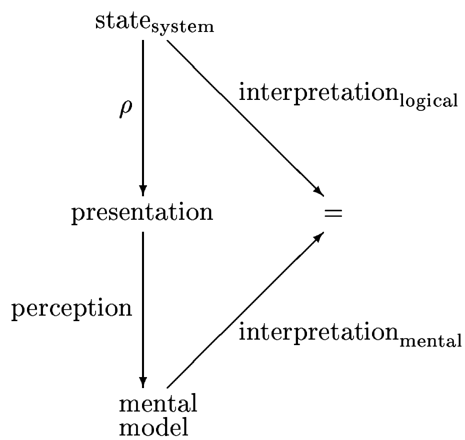

Figure 1: Representational reasoning (basic model)

involved in the interaction:

- functional core — the system state and its operations;
- user interface — the presentation and possible user actions;
- user — the person using the interactive system (building a mental model of the system and trying to fulfill goals by issuing commands).

The diagram also introduces two basic mappings:

- ρ — this map expresses the fact that the presentation of the system can be seen as a reification of the system's state. In most cases it will be this mapping (or its outcome) which will be the subject of analysis. Note that this mapping can be to any combination of modalities.
- perception — this mapping captures what is assumed the user will perceive of the user interface presentation. It can be seen as a filter which the user applies to the information presented at the interface, in order to construct a mental model of the system state. How users will perceive the user interface will depend on the actual system being analysed. The type of property and user being considered also has a bearing on what perception relation to consider. The perceptual capabilities of the user are very much a human-factors issue. Hence, defining the appropriate perception relation, for a specific system, becomes part of a process of discussion between software engineers and human-factors experts, over the design proposed for that system.

The diagram introduces two additional mappings: $\text{interpretation}_{\text{logical}}$ and $\text{interpretation}_{\text{mental}}$. What these mappings express is that, in order for the presentation to be found adequate, it must allow the user to build a mental model which is sufficient for carrying out the required tasks and activities. Because we are dealing with cognitive issues, assumptions must be made about how the users interprets the presentation. This is captured by the $\text{interpretation}_{\text{mental}}$ mapping. Both mappings will be dependent of the specific aspect/property under consideration.

By developing adequate models and mappings for the different aspects identified by the model it becomes possible to reason about whether the presentation is an adequate representation of the system in terms of the users' (assumed) cognitive capabilities. In purely representational terms, an interface is said to be correct if its presentation enables the user to build an accurate model of the underlying system state. This can be expressed as:

$$\text{interpretation}_{\text{mental}}(\text{perception} \circ \rho(\text{state}_{\text{system}})) = \text{interpretation}_{\text{logical}}(\text{state}_{\text{system}}) \quad (1)$$

That is, the user's mental model of the system matches the system state, in so far as the model relates to assumptions about what is considered relevant of the interaction process. The equation above relates to what Norman calls the Gulf of Evaluation: "the amount of effort that the person must exert to interpret the physical state of the system" [17].

This basic model is static. No notion of change is present in either the system state, the presentation, or the mental model. Nevertheless change is an important factor in the perception process. Interestingly the example used in [7] deals with perceived change in a progress indicator. In order to perform the analysis, logical and perceptual operators are used that relate two successive system states/user mental models, respectively. To better deal with change, [2] extends the model to accommodate the notions of system operations, interface actions, and user goals (see Figure 2).

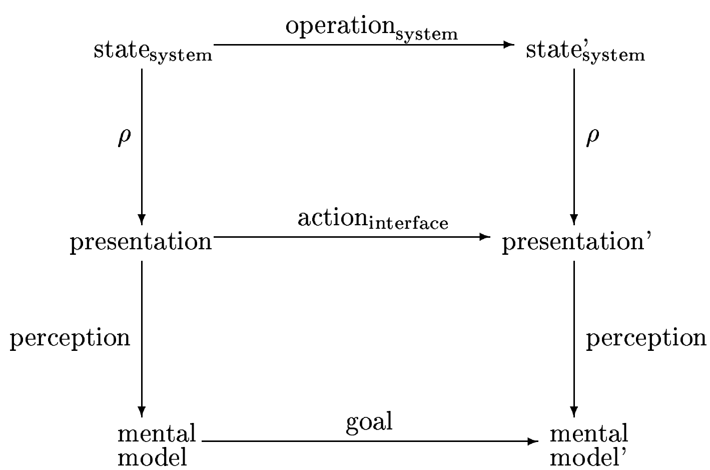

Figure 2: Representational reasoning (actions)

The new version of the model introduces three additional mappings:

- $\text{operation}_{\text{system}}$ — system operations map system states to system states. They are intended to represent the basic functionality of the system.
- $\text{action}_{\text{interface}}$ — interface actions map presentations to presentations. Each interface action will typically be associated with one or a sequence of system operations. Interface actions can occur as a consequence of user activity, or as a consequence of a system operation.
- goal — goals map mental models to mental models. They are used to capture the intentions of the users when using the system. Typically a user goal will give raise to a number of interface actions being executed. Additionally, this mapping can be used to capture changes in the user's mental model which are caused by changes in the presentation.

In this model change is represented explicitly. It allows reasoning about Norman's Gulf of Execution [17]: the *distance* between what the user wants to achieve (the goal), and what is possible at the interface.

$$\text{goal} \circ \text{perception} \circ \rho(\text{state}_{\text{system}}) = \text{perception} \circ \text{action}_{\text{interface}} \circ \rho(\text{state}_{\text{system}}) \quad (2)$$

The equation compares the result of executing an interface action with the goal the user had in mind. It is also possible to reason about whether the effect of specific actions or operations matches the user's goal.

$$\text{perception} \circ \rho(\text{state}_{\text{system}}) \bowtie \text{perception} \circ \rho \circ \text{operation}_{\text{system}}(\text{state}_{\text{system}}) \quad (3)$$

Where $\bowtie$ is a relation on user mental models, capturing the change in presentation which the user should perceive as result of operation $\text{operation}_{\text{system}}$. In fact, this model can be used to characterise not only perceptual properties, but also behavioural ones. Reasoning about whether interface actions and functional core operations are consistent can be performed by proving that:

$$\text{perception} \circ \text{action}_{\text{interface}} \circ \rho(\text{state}_{\text{system}}) = \text{perception} \circ \rho \circ \text{operation}_{\text{system}}(\text{state}_{\text{system}}) \quad (4)$$

This equation expresses the need to verify that related system operations and interface actions are equivalent in terms of the user's perception of their effect on the system.

The model in Figure 2 is biased towards event based systems and it still does not explicitly account for time. Nevertheless, time can have a vital role in user perception, especially when we are thinking of systems involving continuous flows of information between user and system. The analysis in [7] concerns determining whether the representation used in an information display (a progress bar) enables the user to perform a task (to detect progress). It does this by analysing the relationships between two different progress states at the logical level, and their perceptual representations. This is done using the following theorem:

$$\text{Compare-Progress}(x, y) = \text{Compare-Bar}(\rho(x), \rho(y))$$

where x and y are two system states, Compare-Progress is the system interpretation for the task at hand, and Compare-Bar captures both perception and mental interpretation.

We note that this property is conceptually different from those above. In fact, what is being compared here is not simply the effect of operations and goals (on otherwise static states), but the relations between system states and user mental models that change over time (not only because of actions). This is because the progress indicator is a continuous system. Even if there are no actions affecting it state its state will be changing progressively. Hence, change should not be directly connected to the concept of action, but to time. In order to better express this we change the model to that in Figure 3. What the model

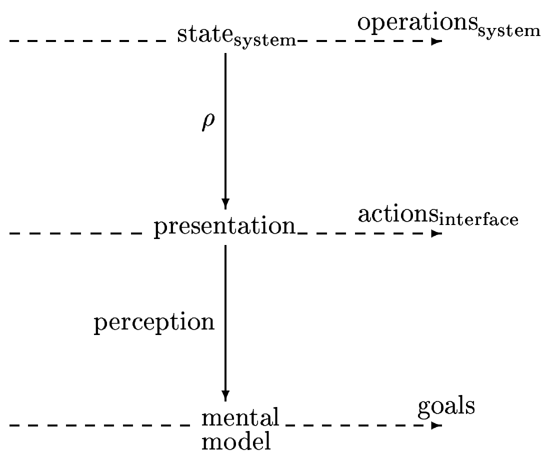

Figure 3: Representational reasoning (time)

intends to show is that all three levels change continuously with time. Note that if we define a number of discrete time slices we will be in the situation of the previous model, since each time slice can then be seen as an action.

Using this model, the relation between system state and user's mental model can be expressed using the following type of equation:

$$\forall_{t_1,t_2} \cdot \text{interpretation}_{\text{logical}}(\text{System}, t_1, t_2) = \text{interpretation}_{\text{mental}}(\text{MentalModel}, t_1, t_2) \quad (5)$$

What the equation expresses is that the evolution of the user's mental model over time must match the evolution of the state of the system. Similar equations can be written to express relations between the presentation and the system state or the user's mental model.

4 A small example

In this section we present an example of how the model introduced in the previous section can be used to reason about the perceptual properties of a user interface. We will show that we can use this reasoning to validate design decision during the early stages of development.

We will consider as an example a number of variations on a small dynamic information display whose purpose is to keep users informed of the progress made during the download of information. In many contexts, such as when using a mobile device, the quality of the connection to the service provider, and hence the rate of transfer can be expected to vary continuously, particularly if the user is on the move. Hence, it will be considered as a design goal that the presentation should enable users to have sufficient perception of the transfer rate being achieved, and especially of variations in the transfer rate. The question of what is "sufficient" depends on the tasks and activities in which the user is engaged - a requirement might be that the user have a reasonable chance of estimating when the transfer will complete, or whether it is likely to succeed at all.

A typical interface component used in this type of situation is the use of a progress bar. Hence, the question to be asked is: how well will a given progress display design reflect the underlying process (information transfer). In order to answer that question, and keeping with the model developed in the previous section, we will develop abstractions for the system performing the download, for the presentation proposed, and for a user's mental model capturing what will be relevant in the context of the task at hand (observing the download progress).

The system model is presented in Figure 4. The notation used is that of the

$$
\begin{aligned}
&\text{Mesg} : \text{TYPE} = [\#\text{Info} : \text{Data} \\
&\qquad\qquad\qquad\quad \text{Size} : \text{Nat}\#] \\
&\text{DL} : \text{TYPE} = [\#\text{Done} : \text{Mesg-list} \\
&\qquad\qquad\qquad \text{Current} : \text{Mesg} \times \text{Nat} \\
&\qquad\qquad\qquad \text{ToDo} : \text{Mesg-list}\#] \\
&\text{State} : \text{TYPE} = \text{Time} \to \text{DL}
\end{aligned}
$$

$$
\begin{aligned}
&\text{rate}_{\text{logical}}((s : \text{State}), (t_1, t_2 : \text{Time})) : \text{Real} = \\
&\quad (\text{progress}(s, t_2) - \text{progress}(s, t_1))/(t_2 - t_1)
\end{aligned}
$$

$$
\begin{aligned}
&\text{progress}((s : \text{State}), (t : \text{Time})) : \text{Nat} = \\
&\quad \left(\sum_{m \text{ in } \text{Done}(s(t))} \text{size}(m) \right) + \pi_2(\text{Current}(s(t)))
\end{aligned}
$$

Figure 4: System model

PVS tool. First, messages (units of information to be transferred) are defined as having information and size. DL represents the state of the system at a given moment in time. Information kept is the list of messages already downloaded, the download state of the current message, and a list of the messages still to be downloaded. The state of the system is defined as a mapping from time to DL.

The interpretation for the download rate at the logical level is also provided. As would be expected, the download rate between any two time instants t_1 and t_2 is calculated by the division of the amount of information downloaded in the period t_1 to t_2 by the elapsed time between the two instants.

As stated previously, we will use a progress bar for the presentation. The model for that presentation is introduced in Figure 5. The progress bar is modelled by two natural numbers. One number representing its size, and another number representing the amount of the progress bar that has been filled. The ρ mapping is also defined. The size of the progress bar will be some predefined

$$\begin{aligned}&\text{ProgressBar} : \text{TYPE} = [\#\text{Size} : \text{Nat}\\ &\qquad\qquad\qquad\qquad\quad \text{Filled} : \text{Nat}\#]\\ &\text{Presentation} : \text{TYPE} = \text{Time} \rightarrow \text{ProgressBar}\end{aligned}$$

$$\begin{aligned}&\rho((s : \text{State})) : \text{Presentation} = \lambda t : \text{Time},\\ &(\#\text{pbsize},\\ &\quad \left(\text{len}(\text{Done}(s(t))) + \tfrac{\pi_2(\text{Current}(s(t)))}{\text{Size}(\pi_1(\text{Current}(s(t))))}\right) \times \tfrac{\text{pbsize}}{\text{len}(\text{Done}(s(t)))+1+\text{len}(\text{ToDo}(s(t)))} \#)\end{aligned}$$

Figure 5: Presentation

constant pbsize. To calculate the amount of the progress bar filled a strategy is used which (by empirical observation) is in use by a number of current applications. The percentage of the progress bar filled will be proportional to the sum of the percentages of the files downloaded.

Finally, Figure 6 introduces the mental model abstraction that will be used.

$$\begin{aligned}&\text{DLp} : \text{TYPE} = [\#\text{Done} : \text{Real}\#]\\ &\text{MentalModel} : \text{TYPE} = \text{Time} \rightarrow \text{DLp}\end{aligned}$$

$$\begin{aligned}&\text{perception}((p : \text{Presentation})) : \text{MentalModel} =\\ &\qquad \lambda t : \text{Time}.(\#\text{Done}(p(t))/\text{Size}(p(t))\#)\end{aligned}$$

$$\begin{aligned}&\text{rate}_{\text{mental}}((mm : \text{MentalModel}), (t_1, t_2 : \text{Time}) : \text{Real} =\\ &\qquad (\text{Done}(mm(t_2)) - \text{Done}(mm(t_1)))/(t_2 - t_1)\end{aligned}$$

Figure 6: Mental model

In this case we are considering only the amount the user will perceive as being the percentage of download accomplished at each moment. The interpretation for download rate at this level is also provided. The definition is similar to that of the system level. The download rate between every two instants will be the amount of progress done in the time interval, divided by the time interval.

Having defined the three levels, we now go on to analyse the proposed design. In order for the interface to provide a sense of continuity, the presentation must provide the user with a sense of change that matches the download rate at the system level. This condition can be expressed by the following equation:

$$\forall_{t_1,t_2,s} \cdot \text{rate}_{\text{logical}}(s, t_1, t_2) = \text{rate}_{\text{mental}}(\text{perception}(\rho(s)), t_1, t_2)$$

If we proceed to attempt a proof of the above theorem, it is easy to see that, even for the case when the same message is being downloaded at both t_1 and t_2, and when the total number of messages does not change between t_1 and t_2 (!), the above is true only when:

$$\forall_{t:\text{Time},s:\text{State}} \cdot \text{Size}(\pi_1(\text{Current}(s(t)))) = \frac{1}{\text{len}(\text{Done}(s(t))) + 1 + \text{len}(\text{ToDo}(s(t)))}$$

What this shows is that all messages must have the same size. This happens because the logical model works on a message by message basis, while the interface attempts to give the users a sense of the download as a unit. Since from the system's perspective the progress bar is divided in as many equally sized chunks as the number of messages being downloaded, smaller messages will give the user the perception of faster download rates, while larger messages will give the users the perception of slower download rates (see Figure 7). This reasoning

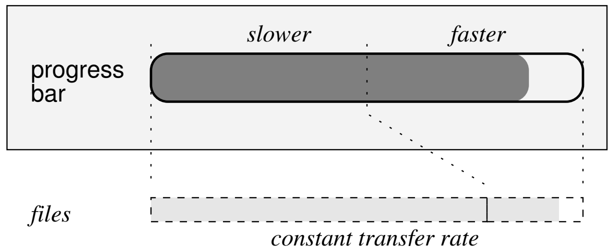

Figure 7: Progress bar vs. file sizes

can be easily validated by proving the following theorem:

$$\text{rate}_{\text{mental}}(\text{perception}(\rho(s)), t_1, t_2) \neq \text{rate}_{\text{mental}}(\text{perception}(\rho(s)), t_3, t_4)$$

for all t_1, t_2 and t_3, t_4 such that t_1 are t_2 are two instants in the download of a message, and t_3 and t_4 are two instants in the download of another message, and the sizes of both messages differ.

One solution to the problem above would be to compute the size of all messages in advance and allocate chunks of the progress bar proportionally to the sizes. This implies knowing in advance the size of all messages that will be downloaded. This is clearly an unreasonable assumption in most settings, but particularly in the context of a user interface where the data is likely to be mixed-media and multimedia documents and information resources. If redesign of the underlying system is not an option, then the solution must be found by redesign of the presentation level alone. So let us consider two alternative designs. Both will be based around the idea of a presentation split into two percepts. One possibility would be to have one component display how many files have been downloaded out of the total number of files, the other stating how much of the current file has been downloaded. An alternative would be to have one percept to give a continuous indication of transfer rate or signal quality, since this is the basic variable indicating performance, along with other to for the overall progress.

In comparison with the initial design, the presentation for the first alternative (see Figure 8) will more accurately represent the system's state (namely, the current transfer rate), and it will not mislead the user into thinking that the state of the progress bar reflects the amount of data still to be downloaded. It will allow users to have a more accurate notion of variations in the transfer rate, while keeping an overall notion of the amount of progress done.

In the typical usage scenario of a mobile or wearable device, however, the design in Figure 8 will still present some problems. In a mobile usage scenario the transfer rate will vary according to the quality of the connection established with the central system. As the connection quality increases, so will the transfer rate that is attainable. It is therefore common for users to *be aware of* for areas of greater connection quality/higher transfer rate. The problem with the design

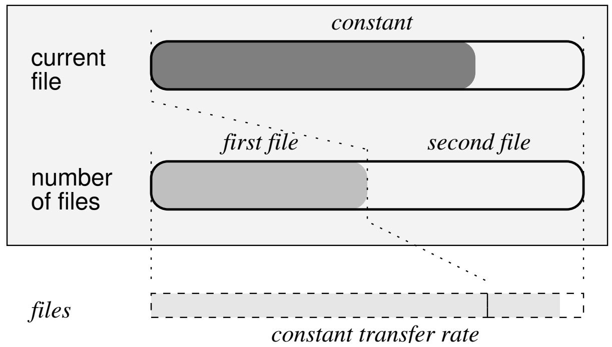

Figure 8: An alternative design

in Figure 8 is that it reflects the transfer rate on a message by message basis only. For each message, the change in the "current message" progress bar will match the transfer rate. However, for the same transfer rate, the progress bar will move faster for smaller messages than for longer messages. This does not support the long term task of looking for higher transfer rates, particularly when a large number of messages must be downloaded.

Since the real goal of the users will be to achieve higher transfer rates, and this is related to the quality of the connection, an alternative design would be to give at the interface, not at indication of the transfer rate, but of the connection quality that determines it. This is shown in Figure 9. In this case

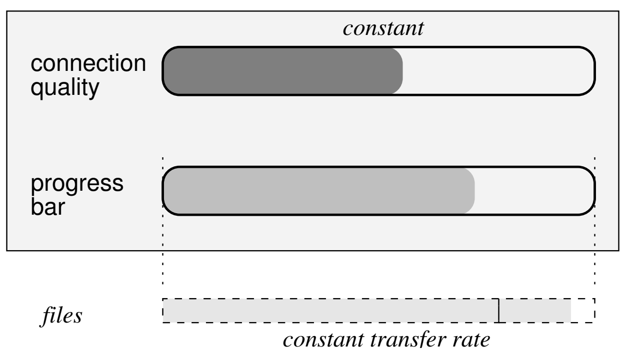

Figure 9: Another alternative design

the design uses a bar graphic percept indicating the strength of the connection being established, and a progress bar to give a general idea of the progress of the download process.

This last design alternative will give users a better qualitative indication of the quality of the download process, while still giving some indication of the overall progress. In particular it gives good indication of instantaneous progress (at each instant is is possible to have an idea of the transfer rate at that moment). The design, however, is still not without problems. As discussed above, the progress bar is not an accurate representation of the overall progress of the download process. This is due to limitations at the level of the system's design. Since this design was motivated mainly by the users' need to be aware of higher transfer rates, the limitation might be found acceptable. It does mean, however, that the user might need to resort to the connection quality information presented to more accurately assess the amount of transfer being achieved within a period. Since this cannot be achieved by means of a simple perceptual operation, this must be done by (mentally) integrating quality over time. This is demanding both in terms of mental effort, and attention required to adequately sample the display.

In general this type of operation can be expected to work if quality varies slowly, but problems will arise if quality varies very quickly (what "slow" and "quick" mean in this context is a matter for discussion with human-factors experts). In that case users will be placed under high attentional and cognitive demands, and it can happen that correct assessment of the amount of transfer cannot be achieved. This, in turn, means that correct assessment of whether the overall quality of the transfer is increasing or decreasing will not be possible when quality varies very quickly.

In summary, this second alternative design procides a straightforward representation of instantaneous progress, and gives good indication of overall progress (or progress over a time interval) only when changes in the connection quality are slow. The previous alternative design gives no easy indication of instantaneous progress, but gives good indication of overall progress, unless a large number of messages with varying sizes is being downloaded. In which case overall progress might also be difficult to achieve.

This discussion gives an idea of the sort of tradeoffs and problems that must be considered and addressed when designing system with a continuous component. It also shows how the model proposed in Section 3 can be used to reason about representational aspects in the presence of time considerations. Additional design alternatives could of course be considered and analysed in similar manner. While it is always tempting to simply add more information (percepts), the limited "real estate" in many ubiquitous devices motivates careful consideration of the prepresentational aspects of a display with respect to the user's tasks and activities.

5 Conclusions

The manner in which information is presented in an interactive system has a profound affect on our ability to perceive and reason about that information. This issue is even more vital in future computing systems, where the technology and form factor have an effect on available modalities. Furthermore, the variety of environments and situations of use (which may not always be amenable to a given form of interaction) add further constraints and challenges, many of which are time-related. Given that dynamically sensed information is a central part of many ubiquitous computing applications, dynamic information displays will be increasingly common. The constraints imposed by the physical form factors motivate careful design of these displays.

In previous work we have examined the issue of representational reasoning [7, 6]. Such work, however, did not explicitly take time into consideration. Nevertheless time is be an important factor in the interaction process between the user and the system. In this paper we have taken a first step towards the inclusion of time into representational reasoning.

We have shown with an example how the model proposed can be used to reason about representational aspects where time considerations are at stake. By using rigorous analysis, it is possible to uncover assumptions concerning interaction and perception, which are implicitly made during the design of the interface. In the example, the problems concerned a mismatch between system level progress, defined on a message by message basis, and an interface which attempts to give users a sense of the download as a unit. Having found a problem with the initial design, alternative designs were considered. The example has shown how considerations about the users' goals, and scenarios of usage, can help in analysing alternative design options. The discussion of the alternatives also shows that while the use of rigorous proofs can help in reasoning about design, it does not tie the designers to its use. In this case the consideration of the two alternatives was informed by the results of the proof performed previously, but did not require a repetition of the process. Nevertheless, rigorous analysis can be used whenever deemed useful to validate specific aspects of the rationale behind a design decision.

One important aspect of the use of rigorous approaches is the possibility of providing automated support for the reasoning process. In this case we have performed all the reasoning in the context of first order propositional logic, which allows us to use readily available tools. This also removes a significant barrier to understanding by non-specialists,

This paper has concentrated in issues of perception and time. One area of work which is not addressed is reasoning about behavioural properties when time is involved. Initial work in this area was done in [8], we plan to address this issue in future work. We would also like to investigate application areas such as mobile support for navigation (as might be found in a modern car).

Acknowledgments

The authors wish to thank Michael D. Harrison for his useful comments on previous versions of the work presented.

References

[1] Gregory D. Abowd, Hung-Ming Wang, and Andrew F. Monk. A formal technique for automated dialogue development. In *Proceedings of the First Symposium of Designing Interactive Systems - DIS'95*, pages 219–226. ACM Press, August 1995.

[2] J. C. Campos. *Automated Deduction and Usability Reasoning.* DPhil thesis, Department of Computer Science, University of York, 1999. Also available as Technical Report YCST 2000/9, Department of Computer Science, University of York.

[3] J. C. Campos and M. D. Harrison. Model checking interactor specifications. *Automated Software Engineering*, (in press), 2001.

[4] A. Dix and G. Abowd. Delays and temporal incoherence due to mediated status-status mappings. *SIGCHI Bulletin*, 28(2):47–49, 1996.

[5] Alan Dix. The myth of the infinitely fast machine. In D. Diaper and R. Winder, editors, *People and Computers III — Proceedings of HCI'87*, pages 215–228. Cambridge University Press, 1987.

[6] G. Doherty, J. C. Campos, and M. D. Harrison. Representational reasoning and verification. *Formal Aspects of Computing*, 12:260–277, 2000.

[7] Gavin Doherty and Michael D. Harrison. A representational approach to the specification of presentations. In M. D. Harrison and J. C. Torres, editors, *Design, Specification and Verification of Interactive Systems '97*, Springer Computer Science, pages 273–290. Springer-Verlag/Wien, June 1997.

[8] Gavin Doherty, Mieke Massink, and Giorgio Faconti. Using hybrid automata to support human factors analysis in a critical system. In *Proceedings of ERCIM workshop on Formal Methods in Industrial Critical Systems*, 1999. (also Formal Methods in System Design, in press).

[9] Wayne D. Gray and Deborah A. Boehm-Davis. Milliseconds matter: An introduction to microstrategies and to their use in describing and predicting interactive behaviour. *Journal of Experimental Psychology: Applied*, 6(4):322–335, 2000.

[10] M. Harrison and H. Thimbleby, editors. *Formal Methods in Human-Computer Interaction*. Cambridge Series on Human-Computer Interaction. Cambridge University Press, 1990.

[11] M. D. Harrison and D.J. Duke. A review of formalisms for describing interactive behaviour. In R. Taylor and J. Coutaz, editors, *Software Engineering and Human Computer Interaction*, number 896 in Lecture Notes in Computer Science, pages 49–75. Springer-Verlag, 1995.

[12] E. Hutchins. How a cockpit remembers its speed. *Cognitive Science*, 19:265–288, 1995.

[13] C. Johnson. The impact of time and place on the operation of mobile computing devices. In B. O'Conaill H. Theimbleby and P. Thomas, editors, *Proceedings of HCI 97*. Springer-Verlag, 1997.

[14] Chris Johnson and Phil Gray. Temporal aspects of usability (workshop report). *SIGCHI Bulletin*, 28(2), 1996.

[15] P. Markopoulos and P. Johnson, editors. *Design, Specification and Verification of Interactive Systems '98*, Springer Computer Science. Eurographics, Springer-Verlag/Wien, 1998.

[16] William M. Newman and Michael G. Lamming. *Interactive System Design*. Addison-Wesley, 1995.

[17] Donald E. Norman. *The Psychology of Everyday Things*. Basic Book Inc., 1988.

[18] S. Owre, J. M. Rushby, and N. Shankar. PVS: A prototype verification system. In D. Kapur, editor, *Automated Deduction — CADE-11*, number 607 in Lecture Notes in Artificial Intelligence (subseries of Lecture Notes in Computer Science), pages 748–752. Springer-Verlag, 1992.

[19] Fabio D. Paternò. *A Method for Formal Specification and Verification of Interactive Systems*. PhD thesis, Department of Computer Science, University of York, 1995.

[20] J. Preece et al. *Human-Computer Interaction*. Addison-Wesley, 1994.

[21] Jef Raskin. *The Humane Interface*. ACM press, 2000.

[22] John Rushby. Using model checking to help discover mode confusions and other automation surprises. In *(Pre-) Proceedings of the Workshop on Human Error, Safety, and System Development (HESSD) 1999*, Liège, Belgium, June 1999.

[23] Bernard Sufrin and Jifeng He. Specification, analysis and refinment of interactive processes. In Harrison and Thimbleby [10], chapter 6, pages 154–200.

[24] B. Ullmer and H. Ishii. Emerging frameworks for tangible user interfaces. *IBM Systems Journal*, 39(3&4), 2000.