Empirical Modelling For The Conceptual Design And Use Of Engineering Products

Chris Roe†, Meurig Beynon†, Carlos. N. Fischer*

† Department of Computer Science, University of Warwick, UK.
*** Departmento de Estatistica, Mathematica Aplicada e Computacional,**
Universidade Estaudal Paulista – UNESP, Brazil.
{croe,wmb}@dcs.warwick.ac.uk, cfischer@rc.unesp.br

Keywords : Empirical Modelling, interaction, statecharts, conceptual models, product design.

Abstract

The process of designing an engineering product usually involves only superficial interaction on the part of the user during the design. This often leads to the product being unsuitable for its target community. In this paper, we describe an approach called Empirical Modelling that emphasises interaction and experiment throughout the construction of a model that we believe has benefits in respect of usability. We use a case study in digital watch design to illustrate our approach and our ideas.

1. Introduction

Developing an engineering product usually involves building a prototype of the required system. This serves to ensure that the product is physically realisable and that it can offer the functionality for which it was designed. However a product that is feasible and functional is not guaranteed to be satisfying to use. Interaction with the product may be ill-matched to a user's conception of the task being performed, so that the use of the product is complex and error-prone even for regular users [9]. A possible reason for this is that all too often the user is only brought in to test the final design. At this stage, the modifications that can be made are limited unless the design is fundamentally altered.

A prototype developed during the design process can often be assessed only superficially, since the user cannot interact with it as they would with the final product. This means that a product developed to fit the specification may fail to satisfy the intended users because they cannot benefit from testing the early prototypes. In effect, too much weight is given to the design and functional specifications, and the quality of the user interface design is often substandard.

Computer-based modelling in principle allows the specification of a prototype that the user can interact with in a realistic manner. The user can then test the design of the interface to identify where possible misuse and inconvenience is likely to arise. The concept of testing the user together with the artefact at such an early stage of the design process is atypical of current engineering practice but will yield usability benefits. This early testing cannot be undertaken without an environment in which a fairly rich animation of the prototype can be created.

As Horrocks [7,p3] remarks, there is a widespread belief in the software industry that user interfaces are easy to develop. User interaction design is usually concerned with look and feel issues and the response of the software to the user's interactions. This can result in an interface that appears simple, yet is ill-suited to the user's conceptual way of interacting with the domain. If the user was more involved throughout the development process, and their experience of the domain could be incorporated into the interface design to make the interaction more natural, the resulting interface would be more satisfying to use.

Horrocks proposes that one way of overcoming the problems of user interface design is to use statecharts, first introduced by Harel [6]. A statechart is an interconnected network of states and events that move between states. Statecharts provide a very rich and expressive notation that allows complex systems to be specified concisely and at different levels of abstraction. Statecharts are much richer than traditional state transition diagrams because they exploit the notions of depth and orthogonality. An example of a statechart for a digital watch is shown overleaf.

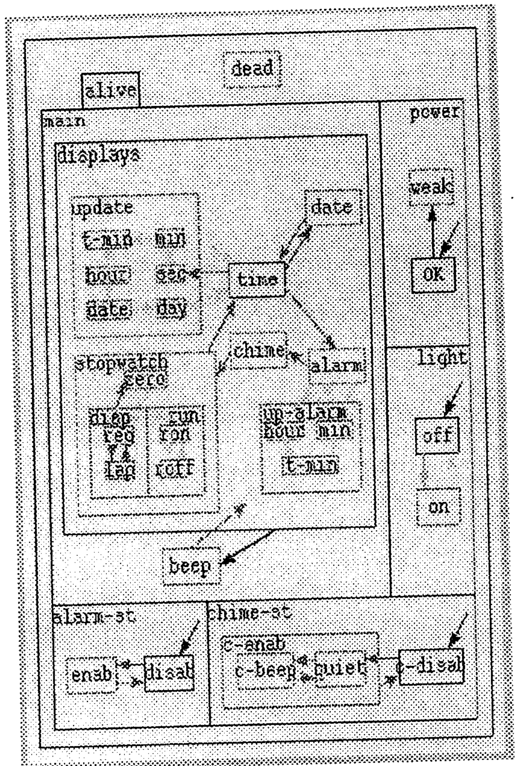

Fig 1 : A statechart for a digital watch

Modelling with statecharts has disadvantages. Creating a statechart relies on our being able to circumscribe the behaviour of the system. This reduces the flexibility of the model the statechart is describing. A statechart does not allow for context-dependent actions. In some situations, the effect of pressing of a button on a watch may depend on previous presses. This information cannot be encoded into a statechart. The statechart is a rigid formal structure suitable for recording reliable and comprehensive system knowledge; it is not so useful when we are still learning about the system that it describes.

In this paper we introduce a novel approach to modelling systems, called Empirical Modelling (EM), developed at the University of Warwick. We discuss cognitive aspects relating to the design of user interfaces and how our models could be used in educational environments. In section 4, we describe a model of a digital watch through which we have illustrated these ideas.

2. Empirical Modelling (EM)

Empirical Modelling (EM) is an approach to constructing models that emphasises interaction and experimentation with the model in all aspects of the design activity [1]. The close association between the model and its intended use aims to reduce the likelihood of the resulting product being unsuitable for the user. Construction of a model is an experimental process with revision occurring as we perform experiments both on the model and in the environment of its real world counterpart. The emphasis on experimentation, observation and interaction gives our modelling approach its distinctive flavour. One of the major advantages of an EM model is its flexibility. We can alter its functionality very easily. This has been demonstrated by the quick revision of the original digital watch model that we have undertaken.

EM is based on three key concepts, namely *observable*, *agency* and *dependency*, but these are conceived initially as personal constructs drawn from our own experience of the domain that we are modelling. An observable is a feature of a system to which we can attribute a value. The assignment of a value is an empirical procedure that involves interaction and experimentation with the real world system we are attempting to model. An agent is deemed to be responsible for changing the value of an observable. Each agent in a model can change a certain family of observables that are deemed to be under their control. A dependency is an indivisible relationship between observables that pertains in the view of a particular agent. In applying EM principles it is the quality of the modeller's construal that determines the extent to which the resulting model can be refined and extended.

An ISM (Interactive Situation Model) models the relationship between a real world situation and an observer of that situation. A change in the model can then be due to either a change in the situation or a new insight on the part of the modeller.

We create an ISM using specially designed tools. The most commonly used is an interpreter called TkEden that allows the specification and execution of definitive scripts representing observables, dependencies and actions. This environment allows us to run a model and make any changes to the script whilst the model is running. We can modify the value of an existing observable, remove one or more observables or add observables to the current situation. This activity corresponds to the modeller refining their understanding of the domain they are modelling.

Our approach to developing models differs from conventional modelling techniques and tries to address shortcomings that we believe traditional modelling techniques possess. Incorporating user-interaction into preconceived patterns of behaviour is problematic because there is no provision for *ad hoc* user intervention. Abstract mathematical models of behaviour are not well-adapted for incremental or evolutionary development because there is a high degree of preconception in their design. The design of an engineering product requires a different kind of justification from pure deduction from laws: there is a role for experimental evidence and for particular knowledge. This is because users differ and cannot be completely described by a simple law.

The power of our development method derives from the idea of systematically refining our system model by using experimentation to identify more and more precisely which factors serve to determine the behaviour. The power to model immediate experience distinguishes our approach from most computational frameworks that are based solely upon abstractions for describing circumscribed behaviour [1].

A recent extension to our modelling tools has been the development of a distributed interpreter called DTkEden [11]. This allows the distribution and execution of definitive scripts across a network of workstations. Different views of a situation can be placed at different workstations and the roles of actors simulated at each workstation. For example, in the simulation of a historic railway accident, the views of the signalmen and drivers can be at different workstations and the interactions between the agents simulated by the communication of definitive scripts or definitions through the network [4]. In this particular example, distributed modelling illustrates the problems that caused the accident and gives indications as to how they could have been avoided.

3. Cognitive aspects of interface design

There are physical and cognitive aspects to good interface design for any product. The former relate to aesthetic and engineering issues such as "Is it attractive to look at?" and "Can you operate the device?" The latter concentrate on the experience gained over a lifetime of use. Relevant issues could include "Can the user carry out standard tasks easily?" and "What is the cognitive overhead in learning a particular function?" The cognitive aspects are more difficult to evaluate on superficial acquaintance with the product, since they pertain to issues that are more complex to evaluate.

To evaluate the cognitive aspects of a watch interface it is essential to understand the processes by which a user learns about the watch. Analysing the learning process is a central concern for educational software designers. When designing educational software, it is difficult to decide when a user has learnt something. A common approach to tackling this problem is to develop worksheets that lead the learner to perform a task aimed at eliciting structured patterns of interaction. IIn our watch case study described in section 4, we have used a worksheet in conjunction with a model to attempt to judge the level of the user's comprehension of the functionality of the watch.

There are many ways that these models can be used in an education or training environment. We can help children to learn to tell the time through using the digital and analogue clock faces. A child could match one of a number of possible clock faces to a particular digital time, and thereby learn to convert between the two. Our model could also be used to train users to make effective use of a watch. Through interaction with the model, the user could gain better understanding of the functionality and become a more competent watch user. The model could also be the basis for a tutorial on watch design. Representing different watches, and discussing their users' conceptual views of how the watch should operate, can give insight into some of the interface issues both in the particular case study and in a more general context of product design. Finally, we can use our distributed modelling tool, DtkEden, to run many instances of the watch model on many clients and monitor all users' interactions on the server. This configuration would allow a teacher to estimate the level of a child's understanding in an unobtrusive manner, without needing to formally examine their knowledge.

4. Case study

We focus on a case study in digital watch design, describing the application of EM to the construction of a model and illustrating the potential benefits that this approach offers in respect of usability.

The computer-based model of the watch we exploit has been derived from a pre-existing model that combines a simulation of the watch with a formal model of aspects of its current state supplied by a statechart (cf. [6,7]). Our new model differs in two respects: its functionality is based upon an actual watch, and the statechart is replaced by an alternative visualisation of state. These modifications incidentally indicate the potential for reuse in EM.

The model features two clock faces, one digital and one analogue, both displaying the current time. The digital watch comprises a display and a set of buttons that can be used to access its extensive functionality. Alongside the watch displays, there is a visualisation of a mental model that a user may construct when learning to use the watch artefact. This visualisation shows the state transitions that can be deemed to be familiar to the user. Arrows between states are coloured to correspond to the button that needs to be pressed for the watch to move between those states. The current state is highlighted with a bold border. As the user gains experience of interacting with the watch, so the number of familiar states increases, and these are added to the visualisation during the interaction process. For a more detailed description of the watch model, see [5]. A screenshot of the digital watch model is shown on the next page. The watch is in the 'display time' mode. From the visualisation we can deduce that the user has explored changing the time of the main clock and altering the alarm, but has not yet encountered the other features.

The digital watch statechart (Fig. 1) shows all the possible states of the watch from the outset. A statechart is a good mental model of a watch that a user already understands since it represents a complete view of a situation. It is not a good mental model of a watch that a user understands only partially, since all the states are visible from the outset. The visualisation adopted in Figure 2 is disclosed only as we explore the functionality of the watch. This is an attempt to represent the knowledge we have of the watch at the present time. As we explore further and gain insight, more states become visible, in much the same way that our mind may create a mental map.

It may be that a user will understand a state and its function the first time that they visit it. What is more realistic however is that the understanding of a state will depend on the user and their environment. The user's current familiarity with a state will be governed by how often the user visits that state and how much time elapses between visits to that state.

One way to reflect this in the model is to make each state's colour become progressively darker each time it is visited and become progressively lighter during the interval between visits. This arrangement gives a useful impression of how the user navigates the visualisation space. It is less useful as a guide to a user's understanding, not least because every user will differ in the time they take to understand the functionality of the watch. There is a danger that the user is 'doing without understanding'. They may be idly circulating through states with no knowledge of how the watch works. Evaluation of the level of comprehension of a user is difficult unless we have some control over the way that they are interacting with the model.

We make use of worksheets in association with the model to make plausible inferences about the user's level of comprehension. If the user's interactions with the model are consistent with action that is purposeful and meets their expectations we may conclude that the user has a good level of understanding of the watch. If their interactions are more erratic and follow a random pattern that eventually meets the required objective then we may conclude that they have a lower level of comprehension. The concept of a worksheet coupled with the model gives the navigation of states a purpose. By setting appropriate tasks for the user to complete or investigate the worksheet designer will know the 'space' that the user should be navigating and the sequence(s) of presses that they should take to complete the task. Depending upon how closely these match, a judgment on the user's level of understanding can be made. The worksheet is a device that attempts to program the interaction of the user in a specific manner in order to achieve some goal conceived by the designer. The complexity here arises from how we determine whether a user has 'understood' a particular state – they may simply have visited it accidentally. This is a major difficulty in any attempt by one person to construct a mental model for another.

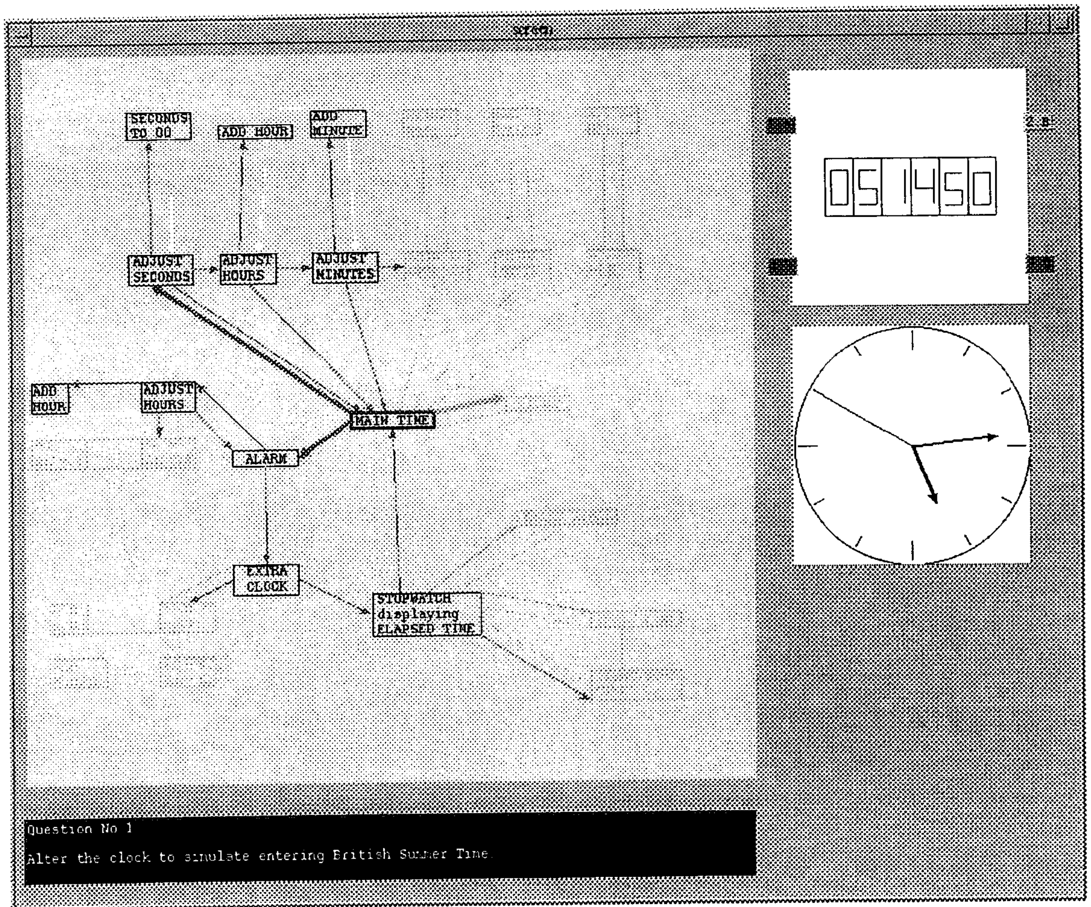

Fig 2: The digital watch model with visualisation and an example worksheet question

When using a physical watch, we learn about it through experimentation, only referring to the brief instruction guide provided if we feel we are making no progress. The accompanying 'documentation' usually comprises a textual description of the buttons that need to be pressed and a diagram to illustrate the correct sequence of button presses to perform a particular function. On the watch face itself, each button typically has a small textual annotation that serves two purposes. Firstly, for the user who has temporarily forgotten a function, the words can be a memory aid in recalling the sequence of button presses. Secondly, for the user who is exploring the functionality of the watch without having consulted the documentation, the words will be used as the clues in the construction of their mental model of the watch's functionality.

The physicality of the watch is important in learning about its functionality. Researchers studying educational software [10] have found that there are two types of real-world familiarity that an object in a simulation must possess for children to engage with it. For *cultural* familiarity an object must be drawn from the culture in which the children live. If a child cannot relate to an object in a simulation they will remain distant from it. For *surface* familiarity an object must look like and behave like its real world counterpart. Our software watch must resemble a real world watch, and its behaviour must be similar to that of a physical watch, or it will be difficult for a child to comprehend it. Since we are considering users learning about a new watch, a comparison with children using software is useful.

With a physical device there are often situational factors that aid the learning process that would be impossible to replicate in a computer model. For example, when using a telephone to dial a number, I can recognise incorrect dialling of a familiar number by the unfamiliar sound that the wrong digit combination produces rather than by observing the position of my fingers. In the case of a watch, it could be the feedback of sounds from button presses or the pattern that the fingers follow when operating the stopwatch that triggers memory of the correct functionality. Such experiential elements contrast sharply with the abstract sequence of button presses that lead to the same effect.

Such aids to learning are associated with *situated cognition*. Lave claims that the way a problem is presented is a factor in determining the probability of solving it [8]. We cannot incorporate these situational contexts into our computer model. Recalling the correct sequence of presses will rely on remembering which buttons to click through interrogating our mental model of the functionality, and not upon motor memory in our fingers.

The investigation of a number of users exploring the model could highlight common preconceptions people have about the behaviour of the watch. People starting to use a new watch might expect the behaviour to be similar to that of the old watch. There is a learning threshold that a new watch user must achieve before being able to use the watch effectively. This will be when their mental model of its functionality corresponds closely with the operation of the watch.

The data that is gathered from an experimental approach such as we have described has many potential applications to design. It could allow a designer to identify better ways of organising the availability of buttons so to allow efficient use by as wide a range of potential users as possible. Analysis of patterns of interaction that users adopt can be the basis for revision of the methods used for interaction with the artefact. Common patterns of misuse and inconvenient aspects of effective use can be highlighted during the construction of the model, helping to ensure that the resultant design is better suited to its user community

5. Conclusions

Good product design requires a good match between the state of the product as apprehended by the user and the status of current tasks within the user's conceptual model. Our case study shows that the EM approach has promise as a way of prototyping engineering products. It also illustrates the benefits that stem in general from much greater interaction throughout the design process. Our work also provides a novel approach to achieving and evaluating good design for usability.

6. Acknowledgments

We are grateful to members of the Empirical Modelling research group for comments received during the preparation of this paper.

7. References

[1] W.M.Beynon. 1994. Agent-oriented modelling and the explanation of behaviour. In *Proc. International Workshop Shape Modelling Parallelism, Interactivity and Applications*, University of Aizu, Japan, September 1994.
[2] W.M.Beynon. 1997. Empirical Modelling for Educational Technology. *Proc 2nd International Conference on Cognitive Technology, CT'97*, IEEE Computer Society 1997, pp54-68.
[3] W.M.Beynon, R.Cartwright, P-H.Sun, A.Ward. 1999. Interactive situation models for information systems development. *Proceedings of SCI'99 and ISAS'99*, Volume 2, pp9-16, Orlando, USA, July 1999.
[4] W.M.Beynon, Y.C.Chen, S.Maad, S.B.Russ, P.H.Sun. 1998. Investigating a historic railway accident. Department of Computer Science, University of Warwick, UK.
[5] C.N.Fischer, W.M.Beynon. Empirical Modelling of products. *In these proceedings.*.
[6] D.Harel. 1987. "Statecharts: A visual formalism for complex systems". Science of Computer Programming, 8: pp323-364.
[7] I.Horrocks. 1999. *Constructing the user interface with statecharts*. Addison-Wesley.
[8] J.Lave. 1998. *Cognition in practice*. Cambridge University Press, UK.
[9] D.Norman. 1998. *The design of everyday things*. MIT Press.
[10] D.Pratt. (under review). Phenomenalising the stochastic: The design of a virtual domain of abstraction. *Educational Studies in Mathematics*.
[11] P.H.Sun. 1999. "Distributed Empirical Modelling and its application to software system development". Ph.D dissertation, Department of Computer Science, University of Warwick, UK.