Stochastic Modelling of Interactive Systems

G. Doherty[1], M. Massink[2] and G. Faconti[2]

[1]Rutherford Appleton Laboratory, Oxfordshire, UK,
[2]Istituto CNUCE, Pisa, Italy

Abstract. Several techniques for specification exist to capture certain aspects of user behaviour, with the goal of reasoning about the usability of the system and other human-factors related issues. One such approach is to encode a set of assumptions about user behaviour in a user model. A difficulty with this approach is that human behaviour is inherently non-deterministic; humans make errors, perform unexpected actions, and, taken individually, both the occurrence of errors and response times can be unpredictable. Such factors, however can be expected to follow probability distributions, and so an interesting possibility is to apply stochastic or probabilistic techniques that allow the modelling of uncertainty in user models. Recently, a number of process algebra based approaches to specifying stochastic systems have been proposed and in this paper we examine the possibility of applying these stochastic modelling techniques to reasoning about performance aspects of interactive systems.

1 Introduction

Interactive systems in the modern world are becoming both increasingly pervasive, and increasingly rich in the variety of tasks supported, the amount of information potentially available, and the different ways in which the user can interact with them. Interacting with such systems can involve multiple media and interaction devices, supporting a continuous flow of information rather than discrete interactions. By discrete interactions we mean those interactions where the user communicates his intentions in the form of separate events such as commands sent when hitting the “return” button and mouse clicks. They are events of relatively short duration and the user usually has control over the time needed to observe the effect of the event and decide the next event.

Continuous interaction differs from discrete interaction in the sense that the interaction takes place over a relatively large period of time in which there is an ongoing exchange of information between the user and the system at a relatively high rate, such as in vision based gestural interaction or haptic interfaces, that cannot be modelled appropriately as a series of discrete events. This shift towards more continuous interaction between user and system means that important properties of such systems are better expressed in terms of some quality of service parameter rather than static “yes or no” properties. Time can play a particularly important role - properties such as latency and jitter (variance of latency) are often critical to the usability of a system. With this motivation, a possible approach to modelling is to apply timed specification and analysis techniques [5].

Time-based properties are certainly relevant to the usability of the system from the human perspective. However, these properties and associated approaches to modelling consider only the system, and the properties are system properties. We would like to investigate timing issues while considering explicitly the role of the user, and the performance of the system as a whole with respect to the user's performance and capabilities. We can enhance the usefulness of such techniques by including partial user models, which encode assumptions about the user's behaviour with respect to time. Such an approach is described in [10], which applies a specification language for hybrid systems to the analysis of a human-operated critical system. It describes the construction of models to represent both system and user. Properties of the combined models are explored with a model checker, for example supporting analysis of the inferences involved in user diagnosis of system failure. However, in the hybrid systems approach, constraints on variables (including timing variables) are given as upper and lower bounds, with no way of determining the likelihood of a given value occurring.

1.1 Behaviour as a Stochastic Process

An important observation with regard to timed modelling is that user behaviour is typically stochastic. While it is certainly not deterministic, it is not completely random, and hence we would expect a variable based on user performance, such as reaction time, to follow a certain probability distribution. Note that such a distribution could be for the general population, or for more restricted groups of users. Although the user behaviour is stochastic rather than deterministic, we do not preclude deterministic time to model relevant parts of system behaviour or of the environment.

We propose that given a set of statistical assumptions about both system and user performance, we can use stochastic modelling techniques to understand the character of the interaction between user and system, in a richer manner than timed modelling techniques can support. In this way, answers to design questions can be both easier to relate to empirical performance data from human factors and usability studies, and also the results of the analysis can be more meaningful for interpretation by human factors experts. Additionally, much modern and emerging user interface technology is stochastic in nature, which provides further motivation for the application of stochastic techniques to modelling interaction.

For the usability analysis of *discrete* interactive devices, probabilistic approaches have been used for analyses based on the possible choices the user may make. For example, one can compare two interface designs with a measure of the average number of state transitions (i.e. button presses and the like) required of a user to achieve some goal [27]. It has been shown that formal approaches of this kind provide a useful complementary assessment of real interfaces that is scalable and applicable throughout the software design cycle.

In this paper we apply the stochastic time approach to address continuous rather than discrete interfaces. For those interfaces the performance in time is critical to their usability.

1.2 Overview

In section 2, we look at different sources of stochastic uncertainty in interactive systems and discuss how stochastic modelling approaches can be applied to reasoning about interactive systems. In section 3 we examine how we might apply stochastic modelling techniques based on stochastic process algebra. In section 4 we look at an example application. We conclude with a discussion of the strengths and limitations of the approach and identify areas for future work.

2 Reasoning about interactive systems

The approach we take is to model both the system and aspects of user behaviour using stochastic modelling techniques. We can see these specifications as a means of making explicit the *assumptions* we make about the capabilities of both user and system, and exploring the behaviour of the combination of system and user on the basis of these assumptions. Modelling the system, and quality of service parameters within the system, is an issue which has already been explored in some preliminary case studies on the application of stochastic modelling such as [4]. A quality of service parameter may be directly related to the usability of the system (for example system response time). We may also have certain quantitative requirements (stochastic or otherwise) which the system must meet. Particular to this context is how we might model aspects of the user's behaviour, and the forms of property of the combined model of user and system which we can investigate.

2.1 Stochastic variables in interaction

Let us consider the kind of stochastic variables which are involved in human performance. Two common measures used in human performance are *performance with respect to time* and *performance with respect to errors*. In practically all cases there is some form of tradeoff between the two. It is important to remember the distinction between the overall measure of performance applied to the users operation of the system and stochastic variables identified in modelling the behaviour of both user and system.

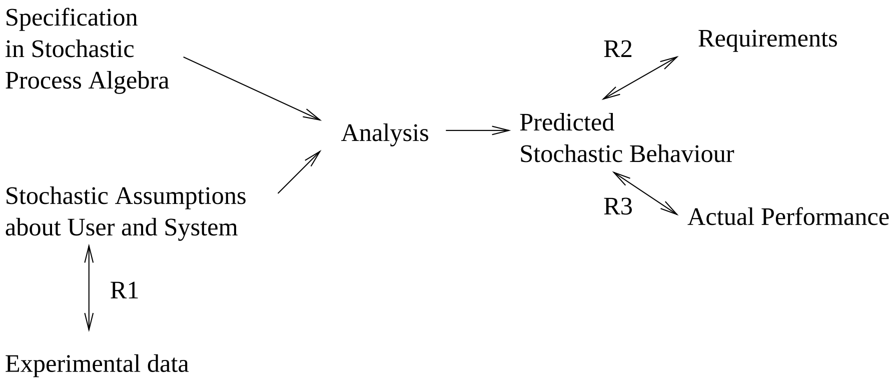
Examples of stochastic variables at this micro- or behavioural level include response times to some stimulus or request from the system, frequencies with which certain operations are invoked and so on. Certain actions are more likely to be invoked than others, and hence selection from some set can also be expected to follow a distribution. Likewise, how often "correct" or "incorrect" operations are chosen can also be expected to follow a distribution. Another possibility concerns "efficiency"; how is the length of an interaction trajectory distributed beyond the canonical or optimal trajectory [14].

2.2 Use of performance data

A necessary facet of stochastic modelling is the use of performance data, and particularly human performance data. The distributions of variables modelling user performance can be based on experimentally derived data. While a large body of data exists, which it should be possible to use immediately, another possibility is to use the process to identify and motivate appropriate experiments to carry out in the context of the development of a certain type of application - for example where novel interaction devices are in use, for which precise performance data from users is not yet available.

Requirements on the performance of a system may also be derived from human factors knowledge. In the case of media streams these might include the performance necessary to preserve a sense of continuity. The performance could be with respect to the noise present in an audio stream, or the synchronisation of video and audio output. We can then relate the predicted results from our stochastic model to the human factors generated performance requirements. Finally we have the opportunity to test the validity of a model by experiment with a system once constructed.

Thus we have a number of roles that performance data (whether it concerns user or system) can play in the analysis of a system (illustrated in figure 1).



Specification in Stochastic Process Algebra
Analysis
Predicted Stochastic Behaviour
R2
Requirements
R3
Actual Performance
Stochastic Assumptions about User and System
R1
Experimental data

Fig. 1. Analysis process

- R1: Incorporating known data into specification, and also motivating experimentation where appropriate data is not available.
- R2: From assumptions about distribution of stochastic variables, ascertain whether system would meet performance requirements, which might for example be expressed as goodness of fit to a distribution.
- R3: Validation of a model with respect to actual performance of user and system.

From the point of view of the overall development process, introducing performance data at an early stage is an attractive proposition since it encourages consideration of problems which might otherwise only emerge during testing, since neither prototypes nor high-specification development platforms are constructed with such issues in mind. Hence, we see such analysis as allowing an iteractive development loop to occur much earlier in the process.

2.3 Distributions of human performance variables

Assigning distributions to aspects of human performance is no simple task. While there is a wealth of experimental data concerning motor skills, response times, and so on, experimental conditions rarely map directly on to real-world conditions.

Swain and Guttmann [32] recommend that the lognormal distribution be used for probabilistic risk assessment concerning human performance. This applies both for performance variables based on time (particularly response times) or quality (such as error rate). Although some performance measures conform to other distributions such as the normal, Weibull, Gamma and exponential distributions [23], in general the lognormal tends to provide a good fit for human performance since competent or skilled workers will tend to have performance around the low (better) end of the performance distribution. An example of a (not so smooth) lognormal distribution is presented in Fig. 3. A lognormal distribution is essentially a normal distribution skewed to the left, i.e. to the ‘faster’ end of the distribution.

It is useful to consider some of the factors which may alter the shape of the distribution. When tasks become more complicated, the lognormal distribution tends to become skewed and approaches the normal for very complex tasks. High stress situations will shift the entire performance distribution to the right, with a skew towards the left. Another possibility is that where there are two or more possible behaviours (eg. two possible diagnoses of a fault) then we may get two clusters of values for a measure such as performance time, and hence we may obtain a bimodal distribution.

2.4 Motor skills

With a wide variety of interaction technologies likely to be based around physical (real-world) artefacts, we expect future computing systems to make greater and more varied use of the users’ perceptual-motor skills.

There is a substantial literature dealing with these skills; perhaps the best known result in HCI is Fitts’ law, which relates mean movement time to movement amplitude and target size. The relationship has been validated for a wide range of tasks and contexts. Within the Fitts’ law literature, of particular interest are studies on variations due to the scale of a movement [24]. Where output is mediated by a computer, lag makes final corrective submovements more difficult and can dramatically increase movement times. Ware and Balakrishnan [33] give a formula for movement time including lag as a parameter. Other variations in sensory-motor performance concern manual control situations; for example the index of performance is lower for a velocity control than it is with a position control [22].

While Fitts’ Law provides us with mean movement times, a problem for the stochastic modelling of motor performance is that Fitts’ law studies typically do not talk about the distribution or variability of movement times. While the distribution in space of “hits” around a target will be normal, the data we have available suggests the lognormal distribution gives a better fit for movement times.

2.5 Human error

One area in which probabilistic issues have received a lot of attention is that of human error, especially for critical tasks, and hence we review here some basic concepts and measurement figures from this area. We would stress that the use of stochastic modeling techniques is not restricted to this domain, but a significant body of research and empirical data is available, which can be incorporated into approaches using eg. stochastic model checking and discrete event simulation.

Human error is defined [30] as: *“A failure on the part of the human to perform a prescribed act (or performance of a prohibited act) within specified limits of accuracy, sequence or time, which could result in damage to equipment and property, or disruption of scheduled operations”*. This applies to both continuous and discrete tasks. In terms of sequencing problems traditional approaches to HCI have considered the behaviour of the system under sequencing errors, eg. omission, commission and transposition [20] and the impact of such errors in some depth [8]. Probabilities associated with such errors have also received attention, but dealing with complex user-system interactions involving different sources and types of error is extremely difficult.

There are a number of different quantities which can be measured when analysing human error, each of which may have a given distribution. Two fundamental concepts - human error probability and reliability can be defined for both discrete and continuous-time tasks. A task is discrete if its content is well predefined with a definite beginning and ending. Human reliability can be expressed in terms of demand reliability for the first, and time reliability for the second.

2.6 Analysis

In the above we have considered briefly some different aspects of human computer interaction which are suitable candidates for representation as stochastic processes. Since the models are constructed in order to facilitate analysis, the level of abstraction and appropriate detail depends on the performance issues of relevance to the analyst. Because of the difficulty in obtaining performance data which precisely matches a given application and context, we would recommend a focus on *qualitative* differences and features in both modelling and simulation. In modelling these would include different modes of use, different modalities and so on. In simulation these would include brittleness and sensitivity with respect to model parameters, the appearance of bimodal distributions and so on.

3 Stochastic modelling

Our discussion of stochastic modelling up to this point has not been based on a particular modelling language. Now that we come to consider the mechanics of construction and analysis of stochastic models, it is appropriate to look at the available technology. Recently, there has been much progress in the verification and analysis of system models that reflect qualitative and performance aspects in the same behavioural model. Much of this work has been developed in the context of process algebras, automata theory and Petri-Nets [28, 18]. The aim is to have a single system model on which different kinds of analysis can be performed by simply adding further details about system behaviour such as real-time aspects and stochastic time features. In traditional approaches qualitative models and performance models have been developed separately giving results that are often difficult to relate to each other. For example, correctness results obtained for the qualitative models could not be assumed to hold for the performance models because they were too different in nature.

Most of the work we have examined on the specification of probabilistic systems has centred on stochastic time process algebras and stochastic automata. The models that are dealt with are essentially of two kinds: Markovian models, i.e. models where the next system state depends only on the current system state, and non-Markovian models. Markovian models have a sound and well-understood mathematical theory, but are restricted in the sense that they must satisfy the memoryless property. In the stochastic time extension of Markov models this leads to the restriction that only (negative) exponential distributions can be used to model stochastic time variables. Tools based on this approach are for example TIPP [16],ETMCC [13], PEPA [11] and PEPP [15]. The advantage of the Markovian approach is that numeric solutions can often be obtained.

In non-Markovian models the next state may depend on the history of how the state is reached, which is the case in the majority of systems. In the timed case this means that distributions other than the exponential can be used to model stochastic time. In general this also means that no numerical solutions can be obtained and that analysis relies on less precise and more laborious (discrete event) simulation techniques. In the context of user modelling Bayesian Networks (BN) [29] have been used quite extensively to model hypotheses or beliefs about human behaviour and to calculate dependencies between these. A drawback of these models is that they give a static view of a situation rather than modelling dynamic aspects of behaviour. Timed extensions of BN [31] have been proposed to deal also with dynamic aspects, but in that setting time is modelled as discrete slots and not as a stochastic variable.

Among specification oriented approaches, stochastic process algebras have a number of advantages over stochastic Petri-nets, particularly in terms of compositionality and formality [17]. In this paper we use a variant of the stochastic time process algebra SPADES (Stochastic Process Algebra for Discrete Event Simulation) [28] that allows the use of arbitrary distributions and in particular normal distributions and lognormal distributions that have shown to fit many human performance variables [32]. A prototype tool for performance analysis of SPADES specifications provides discrete event simulation. In the remainder of this section we give an overview of SPADES and an example. Much of what follows, however, is equally applicable to other stochastic time modelling techniques that allow general distributions.

3.1 Stochastic process algebra representation

The SPADES (♤) representation has many similarities to standard process algebraic specifications such as those in Communicating Sequential Processes (CSP) [19], but with the addition of clocks. Expressions may include the initialisation of clocks (according to a probability distribution associated with the clock), following which they begin to decrease uniformly towards zero, at which point they expire. Expressions may be guarded by more than one clock, in which case the expression is not enabled until all the clocks in the guard have expired. The guards may contain clocks that have been set previously in the same process or by other processes, i.e. clocks are global variables in SPADES. The language of SPADES is restricted to the minimum needed to model stochastic processes in order to study the concepts of such an algebra and its prototype simulation tool. There is ongoing work to extend the language to a richer version and a tool with a proper graphical interface[1]. While we do not intend to go into detail on the stochastic modeling techniques themselves, we give here a short overview of the ♤ process algebra language and automata language, which we use to illustrate the use of such techniques in interactive systems.

The syntax of the language is as follows, for processes p, clocks C and process variables X.

$$\mathit{process} \ ::= \ \textbf{stop} \mid a;\ p \mid C \mapsto p \mid p + p \mid \ \{C\}p \mid p \mid\mid_L p \mid p[f] \mid X$$

We can summarise the meaning of these expressions as follows:

the **stop** process performs no action.
$a;\ p$ - the process performs action a then behaves like p.
$C \mapsto p$ - after the clocks in C reach zero, the process behaves like p.
$p + q$ the first of p or q to be enabled is selected.
$\{C\}p$ - initialise clocks in C, then behaves like p
$p \mid\mid_L q$ - p and q are performed in parallel, synchronised on actions listed in L
$p[f]$ - process p with actions renamed according to renaming function f.

A formal structured operational semantics of SPADES can be found in [7]. As with early work on modelling interaction through process algebra [1], actions can correspond to user operations, system operations, or synchronised operations involving both user and system. The process algebraic representation is compositional, which facilitates the interpretation of the specifications. Process algebraic representations can also be obtained from a graphical representation of stochastic automata which are composed in parallel. Such a graphical representation can be very useful in facilitating discussions on the models in an interdisciplinary setting, in particular when the automata involved are not too complex.

To illustrate the SPADES algebra and the graphical representation, consider a small example involving polling behaviour, where the user checks periodically for an output from the system. We specify this in the process algebra as the parallel composition of two expressions, representing system and user, synchronised on the action where the user ‘notices’ that the system has produced an output.

$$\begin{aligned}&\mathit{stochastic\_sys} == \mathit{system} \mid\mid_{\mathit{notice}} \mathit{user}\\ &\mathit{system} == \{\mathit{st}\}\mathit{st} \mapsto (\mathit{notice};\ \mathit{system})\\ &\mathit{user} == \{\mathit{ut}\}\mathit{ut} \mapsto ((\mathit{check};\ \mathit{user}) + (\mathit{notice};\ \mathit{user}))\end{aligned}$$

The system is ready for the user to notice that it has finished at any time after clock *st* has expired. The user specification states that the user can only participate in the *notice* action when *ut* has expired; if the system is not also ready to participate in the action, then *ut* is reset and the user continues waiting and checking.

[1] For more information, see http://fmt.cs.utwente.nl/HaaST/

Stochastic automata representation The process algebraic specification can be represented graphically as a set of communicating automata as shown in Fig. 2. The semantics of those automata are such that on entry into a given location, the clocks listed for that location are initialised according to their distributions. Associated with outward transitions are both labels which can be synchronised with other automata, and the set of clocks which must have expired before the transition can be taken. Given a choice between different transitions, the first transition to become enabled is taken.

Where we specify a system involving more than one automaton, they are assumed to be composed in parallel, and synchronised as defined by the composition rule in the configuration box at the right in Fig. 2. Also the clocks and their distribution functions are defined in this box. Initial states of the automata are indicated by a double concentric circle. It is important to note however that stochastic process algebra allows hierarchical parallel composition, which is awkward to represent and reason about using graphical automata languages.

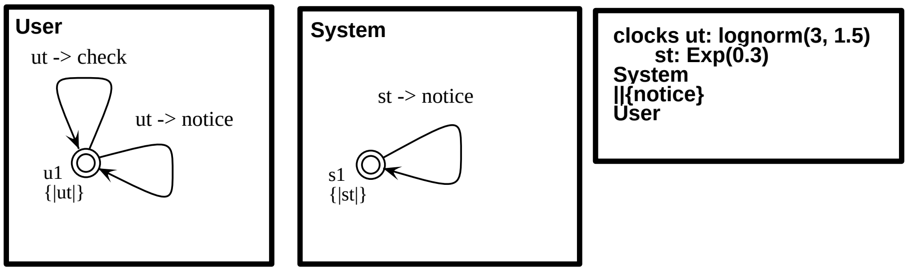


Fig. 2. Stochastic automata for polling

In this example, the left hand automaton represents the behaviour of the user, and the right hand automaton represents the behaviour of the system. As in the SPADES model above, the two automata are synchronised on the *notice* action, which is guarded by the *ut* clock in the user automaton, and the *st* clock in the system automaton. The user automaton may also execute the *check* action, in which case the *notice* action may not take place until the *ut* clock has expired once more.

3.2 Analysis

Stochastic automata and stochastic process algebra specifications are amenable to model checking techniques for reachability analysis if the memoryless condition is satisfied [2, 15, 17, 16]. When *general* distributions are used in the stochastic specification model checking becomes much harder. Preliminary work on model checking of stochastic process algebras with generalised distributions may be found in [21]. When clocks are used in a global way discrete event simulation techniques may be used to obtain the characteristics of the stochastic behaviour of the model. This is the approach we follow in this paper.

The use of stochastic modeling is concerned with random or non-deterministic variables with given distributions. All non-determinism in the specification, which is not represented by such variables, must be removed. In the SPADES approach, this is achieved by means of *adversaries*, which are external processes which decide the outcome of non-deterministic choices. In terms of system modeling the use of adversaries can be seen as representing the implementation architecture (including scheduling policy) of the system. Thus a complete SPADES specification consists of the combination of the stochastic model plus the adversary.

In the case of interactive systems, we can see a possible use of adversaries to model environmental factors, which affect the performance of the system, but which are not part of the system itself. The adversaries may be used to regulate the priority of actions with respect to other actions. For example, in the polling specification in Fig. 2 it is possible that both the action *notice* and the action *check* are enabled. In that case priority could be given to the action *notice*, modelling that if the system is ready and the user is consciously looking at the system to see whether it is ready, then the user notices that it is ready.

Fig. 3 shows the results of simulation for the polling example where non-determinism is solved by an adversary that chooses between enabled actions with equal probability. The chart shows the distribution of the time it takes the user to notice that the system is ready. The results are based on 10,000 simulation runs.

We can observe that in most cases the user notices that the system is ready within 8 time units (seconds). But we also observe that this is not a hard upperbound. It may happen quite often that more time is needed. The chart shows a clear lognormal distribution.

4 Application - Finger Tracking

The aim of the following example is to illustrate how stochastic models may be used to represent both user and system behaviour. The example we consider is based around MagicBoard, an augmented reality application of an interactive whiteboard with a gestural interface based on a computer vision finger tracker [3] and image projection.

What distinguishes this application from an electronic whiteboard is that the user draws directly on a board with real pens; what is drawn on the board may be scanned by the system by means of a camera. The user can use gestures to perform operations on what is drawn on the board, such as selection, copy, paste etc. The results of the operations are projected back on the board. Thus, from the interaction point of view, there is gestural input from user to system, and feedback both of the user's input (real ink on the board) and from the system ("digital ink" projected on the board) as illustrated in the data flow diagram in Fig. 4. The user's gestures are captured by the system using vision techniques that recognise and track the finger of the user after a calibration phase at the beginning of the session. A cursor is displayed (by projection) on the board when the system is tracking and gives the user feedback about whether it is able to accurately follow the position of the finger.

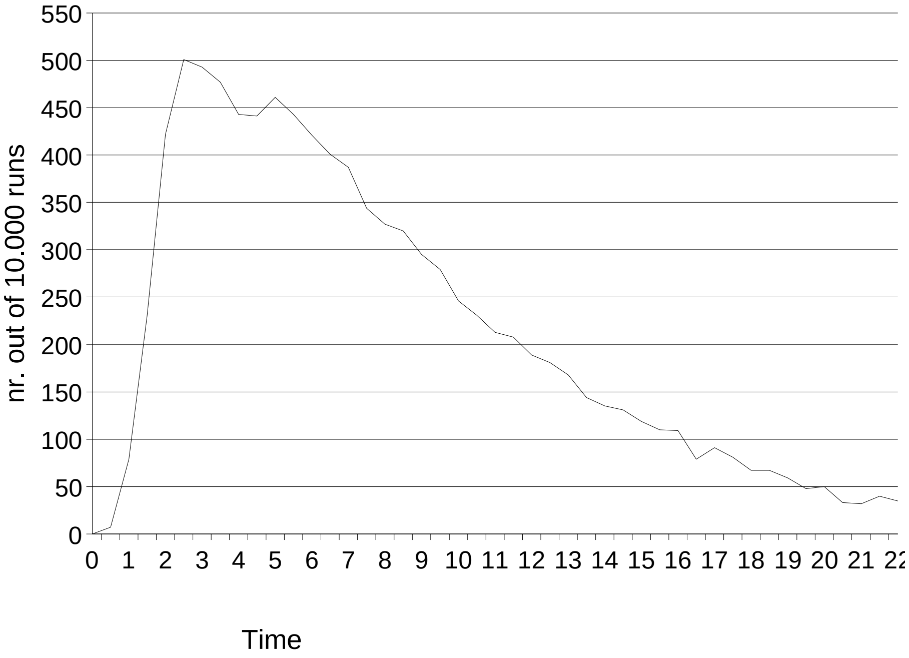


Fig. 3. Results of polling simulation

Jitter A system which does not always keep up with the user (displaying “jitter”) constrains the behaviour of the user. In the following we use the stochastic process algebra approach to model this aspect of the Magic Board interface and examine the relation between system performance and the constraints placed on the user’s behaviour.

The situation that we will examine is one in which the system never completely loses track of the finger, and does not fall behind so much that the user needs to stop moving their finger. So we consider the situation in which the user sometimes needs to slow down their movement to allow the system to keep up.

Modelling timing issues In order to develop an abstract but realistic model we need to make a number of assumptions about the real-time behaviour and its variability of both system and user behaviour. Part of the necessary information can be obtained from the literature. In particular the delay generated by the tracker can most likely be assumed to be accurately modelled by an exponential distribution because each lag occurs independently from any previous lag (memoryless property).

The timing aspects related to the behaviour of the user are of a different nature. First of all we need to know how much time it takes for a user to move their finger from one point at the whiteboard to another point in an unconstrained way. Further we need to know how much the movement slows down when the user observes a lag in feedback. For small distances the first aspect has been addressed in many articles con-

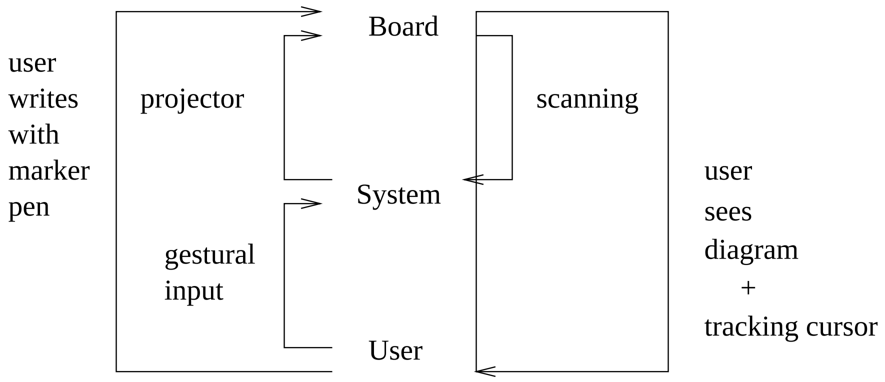


Fig. 4. Input/output flow in MagicBoard

cerning Fitts' Law [26], however, to our best knowledge the variability has never been treated explicitly. This is somewhat surprising because human behaviour is intrinsically variable but not totally unpredictable, i.e. the variability often follows particular probability distributions. Taking variability explicitly into consideration may help us obtain a more informative assessment of the performance of human and system behaviour.

While it is not the focus of this work, in order to have reasonable data for our illustration, we require data on the human performance for moving a finger over large distances and in different directions. We have available experimental data collected at a real whiteboard at the University Joseph Fourier in Grenoble. To keep the amount of work needed to obtain the data within reasonable bounds, each trajectory was divided into three parts of equal length and the time to traverse each part during a single movement has been measured using digital video recordings of each session.

The results showed that the variability in time to move a finger from one place to another on the whiteboard follows an approximately lognormal distribution. Since the distances that are covered on the whiteboard are relatively long (33, 60 and 120 cm resp.) the initial part of the movement, covering the first two thirds of the distance, is performed very quickly; from the motor skills literature (see for example [30]) we take this as the initial, "open loop" or ballistic part of the movement. The last third of the movement is performed more slowly and we take this as corresponding to the visually guided part of the movement to precisely at the target on the whiteboard. In our model, following the (uninterruptible) "open loop" part of the movement, the user checks whether the cursor is managing to follow their finger. A delay may be introduced at this point before the final part of the movement.

Finally we must formulate our assumptions about the threshold of time for the user to take account of the lag (taken as cognitive delay) and the delay introduced by the user taken as a combination of cognitive and motor delay. For these delays we use the bounds data from the model human information processor [6]. Again we are not given the distribution, so we make the mimimal assumption that it is somewhere between 25 and 170 ms, although we would expect it to follow a distribution with a more central tendency. A similar argument holds for the delay introduced by the user which we estimate to be between 55 ms and 270 ms, uniformly distributed.

Stochastic Model We construct a stochastic process algebra model, presented as stochastic automata in Fig. 5, that describes the relevant parts of system and user behaviour. The starting point is when the system is ready to track the finger and the user is about to start moving. This is indicated by label *Start1* at the initial transitions of both processes modelling user and system. After the start the system is in tracking mode modelled by location *Track1* and tries to follow the user movement at least for the time it takes the user to reach the first two thirds of the movement. This time is modelled by the stochastic variable *H1* and has a lognormal distribution that has been derived from the experimental data. Given the values in the data, a good approximation of the distribution is a lognormal with parameters $\mu = 2.3$ and $\sigma = 1.4$, where μ is the mean of the distribution and σ the standard deviation of the distribution.

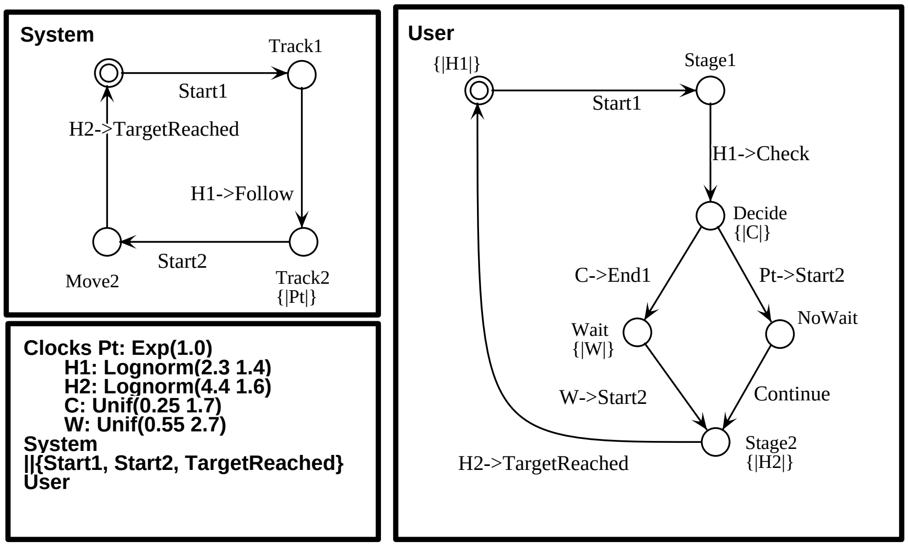
After the tracking phase, the system may show some delay updating the cursor position. This is modelled by stochastic time variable *Pt* that is set in location *Track2*. The second part of the tracking starts when the user starts the last part of the movement with or without delay. The time needed to finish this last part is modelled by the timed stochast *H2*. As soon as *H2* expires, the target is reached. This event is modelled by the synchronisation label *TargetReached*. Given the values in the exprimental data, a good approximation for this part of the movement is a lognormal distribution with parameters $\mu = 4.4$ and $\sigma = 1.6$.

At the user side, after the first part of the movement has been performed we assume that the user observes whether the cursor is sufficiently close to the finger. If this is the case, the movement continues without observable delay due to inertia of the movement. If the cursor is behind, then the user slows down their movement. This is modelled by the stochastic variable *W*. Finally the user performs the last part of the movement to reach the target. Notice how in the model both the system and the user share the variables *H1* and *H2*. The user is the only process to set these variables as they reflect the user's movements. The system uses the variable to obtain the minimum time it would need to follow the user's finger on the whiteboard. This way the natural dependency of the system on the user's behaviour is modelled. The variable *Pt* models the dependency of the user on the system behaviour in a similar way.

The SPADES specification in Fig. 6 describes both the model and the kind of analysis that is performed. In this case we used the model to obtain a histogram of the distribution of the time it would take to reach the target for 10,000 different runs of the simulation of the model. We further specified that when more than one action is enabled, one is selected with equal probability by the adversary.

4.1 Simulation

Presented in Fig. 7 are the simulation results for this model showing the distribution of time until the TargetReached action for a variety of values for system performance. As before, the horizontal axes represent the time and along the vertical axes the number of times that the target was reached is given, out of 10,000 siumulation runs.



```
Clocks Pt: Exp(1.0)
      H1: Lognorm(2.3 1.4)
      H2: Lognorm(4.4 1.6)
      C: Unif(0.25 1.7)
      W: Unif(0.55 2.7)
System
||{Start1, Start2, TargetReached}
User
```

Fig. 5. Finger tracking model - jitter

As we can see, there are two modes, corresponding to the waiting and non-waiting conditions. When system performance is good, the non-waiting mode dominates (curve on the left); as performance degrades it shifts to a bimodal distribution (curves in the middle), and as it degrades further the waiting mode dominates (curve on the right). In this case, the shift to a bimodal distribution occurs between $\lambda = 1.5$ and $\lambda = 0.5$ corresponding to a system that has an average delay of between ca. 60 and 200 ms.

This example illustrates the basic technique of identifying stochastic variables and available performance data, encoding these in a model, which can then be simulated, allowing us to investigate the effect of varying the model parameters. In the example above, we had available experimental data for some aspects of the model, but not all. Data on distributions is particularly difficult to find. We plan to collect more motor skills data, both using whiteboards, and also with a haptic input device which allows us to collect data at such a small granularity that we may plot effectively continuous movement trajectories.

5 Discussion

The tracking example above shows how stochastic models could help to visualise the possible impact of assumptions made about user and system behaviour on the overall interaction. The models allow us to study this impact using powerful software tools such as model checkers and simulators, helping us to investigate the consequences of design decisions in the early stages of interface development.

However, the modelling of assumptions about user behaviour requires a very careful approach. There are a number of problems and limitations that have to be taken into

```
actions Start1 Start2 Follow Check End1 TargetReached
        Continue
clocks  Pt:exp(0.1) H1:lognorm(2.3 1.4) H2:lognorm(4.4 1.6)
        C:unif(0.25 1.7) W: unif(0.55 2.7)
terms
system  = SSys ||Start1, Start2, TargetReached User
SSys    = Start1; Track1
Track1  = H1 -> Follow; Track2
Track2  = {|Pt|} Start2; Move2
Move2   = TargetReached; SSys
User    = {|H1|} Start1; Stage1
Stage1  = H1 -> Check; Decide
Decide  = {|C|}((C -> End1; Wait) + (Pt -> Start2; NoWait))
Wait    = {|W|} W -> Start2; Stage2
NoWait  = Continue; Stage2
Stage2  = {|H2|} H2 -> TargetReached; User
analysis
print action TargetReached as histogram(0.5, 45)
adversary EqProb
runs 10000
```

Fig. 6. SPADES model of finger tracking corresponding to that in Fig. 5

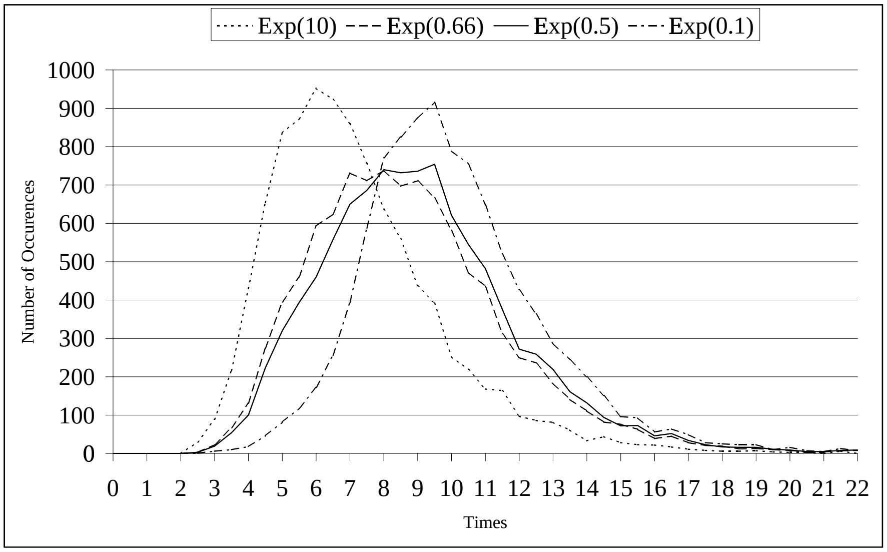


Fig. 7. Results for different tracking performance

account. The best choice of modelling approach may not always be evident, and tool support is still an active area of research. Currently, most available tools are prototypes rather than ready-to-use software packages. We briefly review some of these issues here.

Problems and limitations Although the analysis of human reliability according to procedures like those in Swain & Guttmann [32] is well established in areas such as process control, it is worth sounding a note of caution, particularly where critical applications are concerned. Wickens [34] cites a number of sources of difficulty in conducting this kind of analysis, including the following points:

1. *nonindependence of human errors.* An error which has been made at one point may either increase or decrease our likelihood of making an error in the period following. For example the increased stress may make us more likely to make another mistake, or conversely we might act with increased care and caution, making an error less likely.
2. *error monitoring.* People very often quickly correct capture errors or slips of action before they affect system performance.
3. *parallel components.* With multiple operators interacting at a social level (eg. in a cockpit or control room) operators do not work independently.
4. *integrating human and machine reliability.* Humans tend to adjust their behaviour according to the reliability of the system.

Most of these problems apply at a higher level of analysis than addressed in the current paper - where complex reasoning and decision making affect the overall level of performance. Our focus is on user behaviour and interactions with the system at a much lower level, particularly where there are soft real-time constraints on performance. Additionally, the first point requires both that we be interested in multiple errors, and that the user is immediately aware that an error has been made. The second point assumes that the errors are recoverable, and is again at a level of analysis which is not of immediate relevance for the real-time behaviours we are discussing in this paper. Also it is not just “mistakes” or slips by the user which may lead to poor performance or undesirable behaviour. Regarding the final point, it is certainly true that over time the user would adjust to the reliability of system. However, models for unadapted behaviour could provide a basis for a later comparison between this behaviour and the amount of adaptation that was required from the user to work with the system. Further, it would be interesting to see whether more complex models could be developed that reflect the adaptation process to a certain extend. In particular when questions can be answered under which conditions adaptation may be developed.

Analytical vs. behavioural modelling Where simple error probabilities can be calculated for each source of error, their product can be taken to combine the probabilities. However, when the task to be performed is complex, involving different sources of error (with associated distributions), analysis by hand using statistical rules quickly becomes impractical.

Conversely, processes described by distributions may be modelled directly, but by reducing a complex but well understood behaviour to a known distribution with given parameters, effort and complexity can be saved in constructing the model, and also the work facing the tools in automated analysis.

Tool support Approaches which overcome the compositionality problem by limiting consideration to exponential distributions have better tool support at the moment, but seem limited in the longer term. Also, the knowledge of statistical theory required in approximating distributions seems to be a serious barrier to widespread use. The SPADES process algebra also needs some minimal language features added to allow realistic examples to be specified more easily. In particular, inclusion of a probabilistic choice operator would enhance the expressiveness of the language and allow the modelling of both stochastic time and probabilities related to human error.

6 Conclusions

Stochastic modelling, simulation and analysis using stochastic automata is still a relatively new field. As such, the expressiveness of the specification languages based on the technology, the theories concerning analysis of such specifications, and the incorporation of these into automated support are still at an early stage of development. Due to the limitations of the language constructs currently available in SPADES we have been considering only stochastic time related aspects of interaction. An extension of the language with a probabilistic choice operator would allow the specification of probabilities related to human and system error.

We believe such techniques have an exciting potential for modelling performance in interactive systems, taking into account the abilities and limitations of both user and system. Such models allow us to generate a richer set of answers to design questions, which enables more meaningful comparison of the results of an analysis to human factors data and other empirical evidence. Two specific areas for further investigation are to look at the different ways in which the approach can be used, and also to look at sources of performance data and how they can be integrated.

A final area that needs careful consideration is the form of the specification language that can be used to describe the models. The design of interfaces is a multidisciplinary activity, where models need to be understood and discussed by designers with different expertise and backgrounds. A well-defined graphical modelling language could be a valuable tool for this purpose, especially when such a language is close to other languages used for the specification of software. In this context we have investigated a stochastic extension of UML statecharts [12] based on the formal semantics of [25], which can be mapped to a stochastic process algebra representation [9].

7 Acknowledgements

Many thanks are due to Leon Watts of UMIST for discussions concerning this work and access to his experimental data. This work was supported by the TACIT network under the European Union TMR programme, contract ERB FMRX CT97 0133.

References

1. H. Alexander. Structuring dialogues using CSP. In M. D. Harrison and H. W. Thimbleby, editors, *Formal methods in Human Computer Interaction*, pages 273–295. Cambridge University Press, 1990.
2. R. Alur, C. Courcoubetis, and D.L. Dill. Model-checking for probabilistic real-time systems. In *Automata, Languages and Programming: Proceedings of the 18th ICALP*, volume 510 of *Lecture Notes in Computer Science*, pages 115–136. Springer-Verlag, 1991.
3. François Bérard. *Vision par ordinateur pour l'interaction fortement couplée*. PhD thesis, L'Université Joseph Fourier Grenoble I, 1999. (submitted).
4. H. Bowman, J. W. Bryans, and J. Derrick. Analysis of a multimedia stream using stochastic process algebra. In C. Priami, editor, *Proceedings of 6th International Workshop on Process Algebras and Performance Modelling,* Nice, France, pages 51–69, September 1998.
5. H. Bowman, G. P. Faconti, and M. Massink. Specification and verification of media constraints using UPPAAL. In P. Markopoulos and P. Johnson, editors, *Proceedings of the 5th Eurographics Workshop on Design, Specification, and Verification of Interactive Systems*, Springer Computer Science, pages 261–277. Springer Wien, 1998.
6. S. Card, T. Moran, and A. Newell. *The psychology of human computer interaction*. Lawrence Erlbaum Associates, 1983.
7. P. R. D'Argenio. A compositional translation of stochastic automata into timed automata. (unpublished draft), 2000.
8. A.M. Dearden and M.D. Harrison. Using executable interactor specifications to explore the impact of operator interaction error. In P. Daniel, editor, *SAFECOMP 97: Proceedings of the 16th International Conference Computer Safety, Reliability and Security*, pages 138–147. Springer, 1997.
9. G. Doherty and M. Massink. Stochastic modelling of interactive systems with UML. In *TUPIS Workshop, UML 2000*, 2000.
10. G. Doherty, M. Massink, and G. Faconti. Using hybrid automata to support human factors analysis in a critical system. *Formal Methods in System Design*, (to appear).
11. S. Gilmore and J. Hillston. The pepa workbench: A tool to support a process algebra-based approach to performance modelling. In *Proceedings of the Seventh International Conference on Modelling Techniques and Tools for Computer Performance Evaluation*, volume 794 of *Lecture Notes in Computer Science*, pages 353–368. Springer-Verlag, 1994.
12. S. Gnesi, D. Latella, and M. Massink. A stochastic extension of a behavioural subset of UML statechart diagrams. *FAC?*, 2000.
13. J. Meyer-Kayser H. Hermanns, J.P. Katoen and M. Sigle. Tools and algorithms for the construction and analysis of systems. 1785:347–362, 2000.
14. M.D. Harrison, A.E. Blandford, and P.J. Barnard. The requirements engineering of user freedom. In F. Paternó, editor, *Proceedings of Eurographics Workshop on Design Specification and Verification of Interactive Systems, Italy*, pages 181–194. Springer-Verlag, 1995.
15. F. Hartleb. Stochastic graph models for performance evaluation of parallel programs and the evaluation tool pepp. In *Proceedings of the QMIPS Workshop on Formalisms, Principles and State-of-the-art, Erlangen/Pommersfelden, Germany*, number 14 in Arbeitsbericht Band 26, pages 207–224, March 1993.
16. H. Hermanns, V. Mertsiotakis, and M. Siegle. Tipptool: Compositional specification and analysis of markovian performance models. In *Proceedings of Compter Aided Verification (CAV) 99*, volume 1633 of *Lecture Notes in Computer Science*, pages 487–490. Springer-Verlag, 1999.
17. J. Hillston. *A Compositional Approach to Performance Modelling*. Distinguished Dissertations in Computer Science. Cambridge University Press, 1996.

18. J. Hillston and M. Ribaudo. Stochastic process algebras: a new approach to performance modeling. In K. Bagchi and G. Zobrist, editors, *Modeling and Simulation of Advanced Computer Systems*. Gordon Breach, 1998.
19. C.A.R. Hoare. *Communicating Sequential Processes.* Prentice-Hall International, 1985.
20. E. Hollnagel. The phenotype of erroneous actions. *International Journal of Man-Machine Studies*, 39(1):1–32, July 1993.
21. H. Bowman J. Bryans and J. Derrick. A model checking algorithm for stochastic systems. Technical Report Technical Report 4-00, January 2000.
22. R. Jagacinski, R.D. Repperger, M. Moran, S. Ward, and B. Glass. Fitts' law and the microstructure of rapid discrete movements. *Journal of Experimental Psychology: Human Perception and Performance*, 6(2):309–320, 1980.
23. R. Jain. *The art of computer systems performance analysis : techniques for experimental design, measurement, simulation, and modeling.* Wiley, New York, 1991.
24. G. Langolf, D. Chaffin, and J. Foulke. An investigation of Fitts' law. *Journal of Motor Behaviour*, 8:113–128, 1976.
25. D. Latella, I. Majzik, and M. Massink. Towards a formal operational semantics of UML statechart diagrams. In *IFIP TC6/WG6.1 Third International Conference on Formal Methods for Open Object-Oriented Distributed Systems*, pages 331–347. Kluwer Academic Publishers, 1999. ISBN 0-7923-8429-6.
26. I. Scott MacKenzie. Fitts' law as a research and design tool in human-computer interaction. *Human Computer Interaction*, 7:91–139, 1992.
27. M. Jones P. Cairns and H. Thimbleby. Reusable usability analysis with markov models. *ACM Transactions on Human-Computer Interaction*, (in press), 2001.
28. E. Brinksma P. R. D'Argenio, J-P. Katoen. A stochastic automaton model and its algebraic approach. In *Proceedings of 5th International Workshop on Process Algebra and Performance Modelling (PAPM'97)*, pages 1–17, 1997. CTIT Technical Report 97-09.
29. J. Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. San Mateo, CA: Morgan Kaufmann, 1988.
30. G. Salvendy, editor. *Handbook of Human Factors and Ergonomics.* Wiley-Interscience, 2nd edition, 1997.
31. R. Schäfer and Thomas Weyrath. Assessing temporally variable user properties with dynamic bayesian networks. In A. Jameson, C. Paris, and C. Tasso, editors, *User Modelling: Proceedings of the Sixth Internation Conference*. Springer Wien New York, 1997.
32. A.D. Swain and H.E. Guttmann. Handbook of human reliability analysis with emphasis on nuclear power plant applications - final report. Technical Report NRC FIN A 1188 NUREG/CR-1278 SAND80-0200, Prepared for Division of Facility Operations; Office of Nuclear Regulatory Research; US Nuclear Regulatory Commission; Washington, D.C. 20555, August 1983.
33. C. Ware and R. Balakrishnan. Reaching for objects in VR displays: Lag and frame rate. *ACM Transactions on Human Computer Interaction*, 1(4):331–356, December 1994.
34. C.D. Wickens. *Engineering Psychology and Human Performance.* Charles E. Merrill Publishing Company, 1984.