Resilience of Interaction Techniques to Interrupts: A Formal Model-Based Approach⋆

Maurice H. ter Beek[1], Giorgio P. Faconti[1], Mieke Massink[1],
Philippe A. Palanque[2], and Marco Winckler[2]

[1] Istituto di Scienza e Tecnologie dell'Informazione 'A. Faedo' (ISTI),
CNR via G. Moruzzi 1, 56124 Pisa, Italy
`{terbeek,faconti,massink}@isti.cnr.it`
[2] Institute of Research in Informatics of Toulouse (IRIT), University Paul Sabatier
118, route de Narbonne, 31062 Toulouse cedex 9, France
`{palanque,winckler}@irit.fr`

Abstract. In many modern working environments interruptions are commonplace as users must temporarily suspend a task to complete an unexpected intervening activity. As users are faced with more and more sources of information competing for their attention, it is becoming increasingly important to understand how interruptions affect their abilities to complete tasks. This paper introduces a new perspective for research in this field by employing analytical, model-based techniques that are informed by well-established cognitive theories and empirical data available in the literature. We propose stochastic modelling and model checking to predict measures of the disruptive effects of interruptions to two well-known interaction techniques: Drag 'n Drop and Speak 'n Drop. The approach also provides a way to compare the resilience of different interaction techniques to the presence of external interruptions that users need to handle. The obtained results are in a form that allows validation with results obtained by empirical studies involving real users.

1 Introduction

In many modern working environments interruptions are commonplace as users must temporarily suspend a task to complete an unexpected intervening activity. Interruptions are unpredictable and quite often cannot be disregarded by users in working environments. Web page pop-ups, phone calls, emails, instant messaging and social events can also be disruptive when people need to concentrate on certain tasks. One of the interesting aspects of interruptions, according to O'Connaill and Frohlich [17], is that they reveal that the timespace of any individual is not owned and controlled in the same way as their workspace, but can collide and merge with that of another individual unexpectedly.

Research has shown that different types of interruptions can affect their disruptiveness. Quite often, interruptions are associated with negative effects:

---

resuming a task after an interruption is difficult and may take a long time, interrupted tasks are perceived as harder than uninterrupted ones, interruptions cause more cognitive workload and they are quite often annoying and frustrating because they disrupt people from completing their work.

Interruptions can also lead to incidents due to human error. According to Trafton & Monk [22], pilots experiencing interruptions during preflight check-lists have been blamed for multiple aviation crashes. In addition, recent studies have shown that interruptions may be an important factor in driving, emergency room care and nursing errors. Indeed, frequent interruptions can reduce user performance. However, not all interruptions have negative impact: awareness systems such as alarms and alert systems effectively shift our attention to matters that need immediate care and, at least for simple tasks, interruptions may actually increase performance.

As users are faced with more and more sources of information competing for their attention at any time, it is becoming increasingly important to understand how interruptions affect one's abilities to complete tasks [5]. Interruptions raise questions of non-exclusive practical and theoretical significance including: How many interruptions occur at work? How is performance affected by various interruption characteristics, like complexity, duration, timing and frequency? How many interruptions are disregarded rather than handled? Who benefits from the ensuing interactions? How disruptive are interruptions to prior tasks? What can be done to mitigate negative disruptive effects? Most current research tackles these questions by conducting empirical studies with users, either on controlled conditions (i.e. usability labs) or on working environments (e.g. ethnographical studies). This paper introduces a new perspective for the research in this field by employing stochastic model-based techniques during early phases (i.e. specification phases) of the development process of interactive systems to investigate potential disruptive effects of interruptions on user performance and the *resilience* of interaction techniques to such interruptions, i.e. the ability to sustain the impact of interruptions and recover and resume its operations [11].

Outline of the paper. We start by a review of the literature on interruptions in Sect. 2 followed by a description of our proposed methodology in Sect. 3. In Sect. 4 we briefly recall the cognitive theory ICS that we use to motivate the models of the user aspects of the interaction techniques presented in Sect. 5. The stochastic model addressing the multi-modal Speak 'n Drop interaction technique is presented in Sect. 6. In Sect. 7 the results of the performance analysis of the models are presented and their differences in resilience to external interrupts are discussed for different assumptions on the average number of interrupts that occur. Finally, in Sect. 8 we conclude the paper with some considerations on current work and a discussion of future research on this topic.

2   Task Interruptions

Interruptions occur when a person is working on a primary task (usually long-lasting) and an alert for a secondary task occurs [22]. Sources of alerts could be either internal (i.e. user thoughts) or external (e.g. a person coming into the

room to ask the person a question, a fire alarm, instant messaging). An important aspect of alerts is that they create an interruption lag as the user has to turn her attention to the interruption. The person then starts the secondary task. After completing it, the person must resume the primary task. During the resumption lag, the person must figure out what she was doing during the primary task and what to do next. Finally, the person resumes the primary task. From this task analysis and real-world examples, it is clear that different aspects of the cognitive system are relevant to the study of interruptions and resumptions.

Since the seminal work of Zeigarnik [24], who was the first to publish the relations between interruption and selective memory, researchers have not ceased to document other effects of interruption in human behaviour.

There have been several attempts to formalise cognitive models describing the impact of interruptions in human behaviour. Only a few, however, address formal description techniques to describe the occurrence of interruptions in system specifications [12]. In fact, there seems to be a gap in the literature concerning predictive methods to system specifications towards hazardous effects of interruptions. A main problem is to identify suitable notation to formalise interruptions occurring in interactive systems. The unpredictability of interruptions would favour the use of declarative models to describe what should be accomplished by the user system (whatever happens) rather than describing the steps required (i.e. control flow) to accomplish it [21]. Nevertheless, there are situations where the interruption of an actual task should be considered part of the user goals, e.g. to cancel document printing. Indeed, task models like CTT [20] explicitly provide the operator suspend/resume to allow explicit modelling between tasks. Similarly, West and Nagy [23] have added theoretical structures to the notation GOMS to overcome its limitations for analysing interruptions when task switching is common. Jambon [12] has analysed the idiosyncrasies of relationships between tasks (e.g. parallelism and sequence) to derive a formal model (using automata) describing the semantics of interruptions in notations like MAD, UAN and Petri nets. However, none of these works has yet evolved to a systematic analysis approach to deal with interruptions during task execution.

3   Methodology

The methodology we propose and experiment in this paper exceeds the specific case of interruptions and can be used whenever it is necessary to make design decisions concerned with usability issues. For this purpose we address syndetic modelling [6], relevant human factor studies and formal specification with particular emphasis on stochastic techniques [4]. Syndetic modelling is a conceptual breakthrough in interactive system and man-machine interface design. It provides design and formal specification and verification techniques that take into account both human's capabilities and limitations together with robustness of interactive systems, thus enabling the study of the joint man-machine behaviour. Such joint models allow for the investigation of properties expressing requirements or expectations and provide insight in the extent to which an interactive

system meets such requirements and constraints. The point of departure from known design methods is the requirement that the system should be usable. This is not just a mechanical property of the system, but a statement that implicitly or explicitly must embody some claim or understanding about human capabilities and limitations. In other words, in addition to being a (formally) provable consequence of the specification, the property must also be psychologically valid.

With a syndetic approach, capabilities of and constraints on user behaviour are expressed explicitly by representing a cognitive model (or an approximation) as a formal theory, which can then be integrated or combined with the model of system behaviour. In this way, the user model is explicit and it contains the theoretical basis for the claims based on it; the model is as correct as the theory it encapsulates. However, properties can be expressed and verified only at the level of abstraction at which available models describe cognitive behaviour. If this is not the case, we need to refine aspects of user behaviour in the model by addressing human factors and usability studies. Given a set of statistical characteristics of both system and user performance derived from literature and available empirical data, we can use stochastic modelling techniques to understand the character of the interaction between user and system. We can see the resulting specification as a means to make explicit the assumptions about the capabilities of both user and system, and to explore the behaviour of the combination of system and user based on these assumptions. In this way, answers to design questions can be both easier to relate to empirical performance data from human factors and usability studies, and the analysis results can be more meaningful for interpretation by human factors experts. Moreover, much modern and emerging user interface technology is stochastic in nature, which provides additional motivation to apply stochastic techniques to model interaction.

Our methodology uses a range of theories and techniques, from cognition to formal languages, to model interaction. For the purpose of the current work we make use of the following theories and techniques and, consequently, apply them: ICS theory [2]; studies on pointing gestures [14,8]; the PEPA process algebra [10]; the stochastic model checker PRISM [13]. We use these to develop a case study for comparing two interaction techniques to remove objects from a display in the presence of interrupts.

4 Interacting Cognitive Subsystems

Interacting Cognitive Subsystems (ICS) [2] is a comprehensive model of human information processing that describes cognition in terms of a collection of subsystems, each of which process different mental representations. There are three sensory subsystems (*visual, acoustic, body state*), four central subsystems composed of two structural subsystems (*morphonolexical, object*) and two meaning subsystems (*propositional, implicational*), and two effectors subsystems (*articulatory, limb*). These representational subsystems are supplemented by peripheral somatic and visceral response systems (Fig. 1).

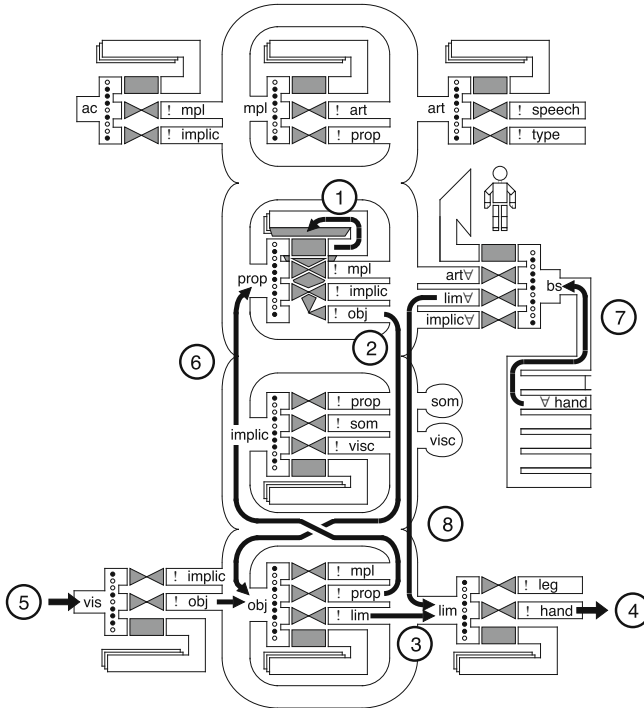**Fig. 1.** ICS architecture configured for gestural interaction

The internal decomposition of all subsystems is identical. Incoming data streams arrive at an input array, from which they are copied to an image record representing an unbounded episodic store of all data received by that subsystem. In parallel with this copy process, each subsystem also contains transformation processes to convert incoming data to certain other mental codes. This output is passed through a data network to other subsystems. If the incoming data stream is incomplete or unstable, a process can augment it by accessing or *buffering* the data stream via the image record. However, only one transformation a time can be buffered in a given processing configuration. Coherent data streams may be blended at a subsystem's input array, as a result of which a process can *engage* and transform data streams derived from multiple input sources.

Overall behaviour of the cognitive system is constrained by the possible transformations and by several principles of processing. Visual information for instance cannot be translated directly into propositional code, but must be processed via the object system that addresses spatial structure. Although in principle all processes are continuously trying to generate code, only some of the processes will generate stable output that is relevant to a given task. This collection of processes is called a *configuration*. The thick lines in Fig. 1 show the configuration of resources deployed while using a hand-controlled input device to operate on some object within a visual scene.
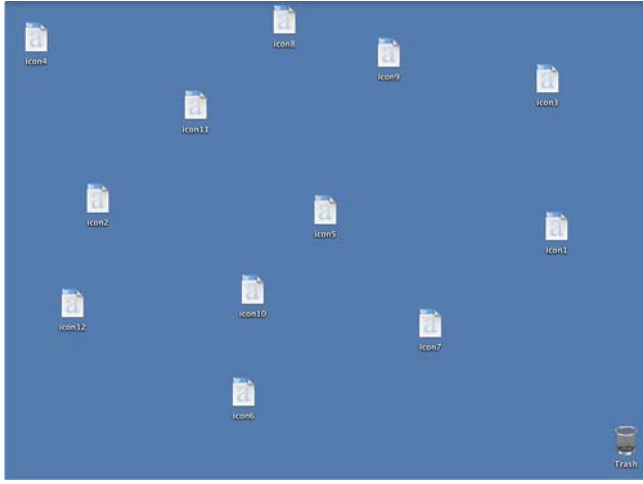
Fig. 2. Example of display

5   Case Study

We developed a case study in order to highlight the potential of the proposed methodology. The choice of the case study is motivated by the fact that a number of real-life applications require operators to remove objects from the current system presentation, such as, e.g., in multi-modal man-machine interfaces for space ground segment applications [19].

   We assume the presentation to be composed of a set of icons on a display, the standard device for output communication in man-machine interfaces. It is well known from psychological theories [9] that design and layout of objects in a visual scene, as well as other (multi-)media structures, play a fundamental role in the way people perceive, think and react to sensorial stimuli. However, this level of detail is beyond the scope of our current work. We assume that the symbolic configuration or pattern of icons is based on a set of features that directs the structuring of the visual scene into a group of icons with one distinguished icon (the trash in our case). The user is asked to remove the icons from the display (see Fig. 2) using one of the following two interaction techniques.

Drag 'n Drop. This well-known technique works as follows: (i) by means of a mouse device, the user moves the cursor on the display over one of the icons; (ii) by pressing the appropriate button of the mouse, she selects that icon which gets highlighted and linked to the cursor; (iii) while keeping the button pressed, she moves the mouse dragging the icon over the trash; (iv) she releases the button causing the system to remove the icon from the display.

Speak 'n Drop. This multimodal technique uses a combination of mouse and voice commands: (i) the user selects one of the icons by positioning the cursor over it and by pressing and releasing (click) the appropriate button of the mouse;

(ii) she pronounces the word *delete* interpreted as a command by the system. When both the icon selection and the delete command are acquired by the system, that icon is removed from the display. No constraints are imposed on the users' behaviour, i.e. the modalities can be used in any order as well as concurrently, and the system must be designed to cope with this.

The performance of a user using a particular technique can be characterised by the number of items she is able to drop during a fixed period of time. We assume an interrupt to manifest itself as a pop-up window fully covering the display. The user has to click the mouse button over a push button positioned in the centre of the window to make it disappear and to resume the previous task.

6   Model Development

To illustrate the methodology, we first describe the general assumptions w.r.t. the case study, followed by the steps necessary to develop a stochastic model of it.

We include in the model only correct behaviour of both the user and the system. We do this deliberately in order to be able to separate two concerns: reduction of performance due to the occurrence of external interrupts and reduction of performance due to user and/or system errors. Although the consideration of the combination of erroneous behaviour and interrupts would be interesting too, it is useful to be able to study the effects of these two aspects first in separation. In fact, if both aspects would be included in the model from the beginning it would be no longer clear to which extent a certain outcome should be attributed to erroneous behaviour, to the effect of interrupts or to both. In this work we are mainly interested in the effect of interrupts and a formal model based methodology that allows one to compare the resilience to external interrupts of different interaction techniques.

A further concern is the level of abstraction used for modelling. We choose to keep the models relatively abstract, modelling observable events, and to refine the models only after a clear indication that this would lead to significantly better approximations. Finally, we use exponential distributions to approximate the average duration of activities. This way we make minimal assumptions on the exact shape of the distribution of the duration of individual actions.

6.1   User Model

In the first phase of model development, we consider an abstract view of the flow of information between interaction devices, system and user with the aim of getting insights in the cognitive resources required to perform the task at hand.

Referring to the ICS theory, features, such as colours and edges, contribute to form a mental *visual representation* derived from raw data acquired by the eyes. The structure of the visual scene in terms of icons is a more abstract *object representation* obtained by combining the visual representation with the knowledge and the experience the perceiver has of the world. This knowledge comes from a level of abstraction, where objects are named and their properties identified

in terms of a *propositional representation*. Thus, the current structuring of the visual scene is used to augment the propositional knowledge about the objects being sensed. Visual and propositional representations can be combined to produce a further level, called *implicational representation*, where the general meaning of information is stated. Combining implicational and propositional knowledge, people are able to define goals and to act accordingly. Consequently, the object representation enriched by propositional data can be transformed into *limb representation* controlling physical actions performed, e.g., by hands and eyes. A continuous source of sensorial information, *body state representation*, provides feedback to the co-ordination of the physical actions. In the Drag 'n Drop case, this is thinking mentally of moving the cursor over one of the icons, push the mouse button to select it, drag it over the trash and release the button. The propositional knowledge can be transformed into a *morphonolexical representation* describing the phrases' structure. This is transformed into an *articulatory representation* controlling the physical production of speech. For Speak 'n Drop, this involves thinking mentally of moving the cursor over one of the icons on the display followed by a mouse click, while pronouncing the word delete.

The above reasoning is described by a set of subsystems, where knowledge is stored as representations, a set of processes transforming the knowledge from one representation into another, and a set of communication paths carrying the information from one subsystem to another. The set of transformations in place at a given moment in time, fully characterises the mental activity and is referred to as a *configuration*. The *reaction time* is the interval from the acquisition of sensorial data to the production of physical actions. It represents the time the user needs to put in place the appropriate mental configuration in order to react to sensorial data. From literature on cognitive psychology it is known that each transformation step in such a configuration takes approximately 40 milliseconds on average. Consequently, the time required to deploy the configuration for graphical interaction in Drag 'n Drop is at least 240 ms. The configurations for Drag 'n Drop and Speak 'n Drop, resp., are specified by

$$\text{DnDconfig} \supseteq \bigcup \left(\begin{array}{l} \{\text{:vis-implic:, :prop-implic:}\}, \\ \{\text{:obj-prop:, :implic-prop:}\}, \\ \{\text{:vis-obj:, :prop-obj:}\}, \{\text{:obj-lim:, :bs-lim:}\} \end{array} \right) \quad \text{and}$$

$$\text{SnDconfig} \supseteq \bigcup \left(\begin{array}{l} \{\text{:ac-mpl:, :prop-mpl:}\}, \{\text{:mpl-art:}\}, \\ \{\text{:vis-implic:, :prop-implic:}\}, \\ \{\text{:obj-prop:, :implic-prop:, :mpl-prop:}\}, \\ \{\text{:vis-obj:, :prop-obj:}\}, \{\text{:obj-lim:, :bs-lim:}\} \end{array} \right)$$

The ICS theory can be used also to describe which changes in mental configurations occur when unexpected events happen during a user's activity aiming at satisfying a specific goal. The mechanism put in action is the stopping of the activity of effector subsystems and the deployment of the following configuration:

$$\text{RecoveryConfig} \supseteq \bigcup \left(\begin{array}{l} \{\text{:ac-implic:, :bs-implic:, :vis-implic:, :prop-implic:}\}, \\ \{\text{:implic-prop:}\} \end{array} \right)$$

The sensorial and propositional information is blended at the *implicational subsystem* and fed back into the *propositional subsystem*. This loop with information mutually exchanged between implicational and propositional subsystems,

enables to get insights on what we know both as facts and feelings and to reason about the current context in which the unexpected event has occurred.

To get better insight in the user's performance in pointing movements, we need to refine the ICS model by addressing human factors and usability studies. It is well known that the duration of pointing movements is fairly well approximated by Fitts' law. Fitts' law based experiments show that the average time spent to point at an icon on a computer display by operating a mouse is in the order of 1000 ms [14]. Additionally, more recent findings [8] show that the movement itself can be distinguished into several phases including a *planning phase*, where the display is investigated and the propositional goal is formed; a *ballistic phase*, where the movement is based on low-level hand control (i.e. without buffering taking place at limb subsystem); an *approach phase*, performed under *visual control* requiring focus of attention (i.e. with buffering taking place at the limb subsystem [7]); and an *adjustment phase* to check that the target has been reached, similarly requiring the focus of attention. With this information, it is possible to refine the ICS analysis by splitting the pointing movement into two distinct phases: the first includes the planning phase plus the ballistic movement; the second consists of visual control. Consequently, the transfer of the buffer may occur later w.r.t the start of the movement and this will have important effects in the case of multimodal interaction.

6.2   Stochastic Automata

We move from the semi-formal and qualitative reasoning to a formal and quantitative one by specifying the identified user activity as PEPA (Performance Evaluation Process Algebra) models [10], described via a parallel composition of stochastic automata.

Configurations are modelled as states of the automata. Configuration changes usually identified by an observable action, are modelled as pairs *(action type, rate)* linking two states. *action type* denotes the type of the action and *rate* the negative exponential distribution of the activity duration; that is the average period of time during which a particular configuration (state) is in place. A special case are actions with infinite rates called *passive actions* as opposed to *active actions*. These play a special role in the parallel composition of the automata in which synchronisation on *(action type, rate)* pairs can be specified. The expected duration of a passive action cooperating with an active one is fully determined by the rate of the active one. The expected duration of a cooperation of active actions is a function of the expected durations of the corresponding activities in the components (typically corresponding to the longest one).

Due to lack of space, we only briefly describe one of the stochastic automata developed for the Speak 'n Drop interaction, shown in Fig. 3 (a); the interested reader is referred to [3] for details. To illustrate our approach to calculate the values of the rate parameters, consider transition Usr $\overset{(\text{move},im)}{\longrightarrow}$ UsrMve modelling the initial part of the pointing movement. The value of rate im is composed of a planning phase of 240 ms, which is the time to deploy the relevant mental

configuration, and the average time of the ballistic mouse movement, estimated to be 670 ms on average given the average distance that a user needs to cover to reach an icon on the screen obtained from empirical data available in the literature. Rate im then equals $1/(0.240+0.670) = 1.1$. Other values are obtained in similar ways, resulting in $ss = 1.6$, $vc = 3.4$ and $mc = 8.33$ for this case study.

Speak 'n Drop interaction is multimodal in nature, requiring both speech and gesture. In a *real setting*, where pointing gestures are performed in the same visual space where the referred objects exist, it will result in the performance of deictic references. In our case, the key question is whether users will be able to deploy the resources of the interface to achieve their tasks by performing a deixis or by using a sequential construct. Two major facts constrain user performance of Speak 'n Drop: the performance of the speech recognition system and the use of the mouse. Realistic speech recognition systems matching an acceptable performance accuracy operate at a rate in the range of $2/2.5 \times real\ time$. For the use of the mouse, it is known [7] that operating any device in a space different from the visual one, under visual control, requires the transfer of the buffer to the limb subsystem. Consequently, users will focus on operating the mouse device and they will be unable to initiate speech. However, they will be able to sustain previously initiated speech that is not in conflict with resource allocation.

In Fig. 3(a), transition UsrMve $\overset{(\text{startSpeak},ss)}{\longrightarrow}$ UsrSpeak1 defines a user that starts to speak while operating the mouse before entering visual control and that will achieve both goals in parallel. The alternate UsrMve $\overset{(\text{visualCtrl},vc)}{\longrightarrow}$ UsrOp1 refers to the case in which the visual control is entered before speech is started. Consequently, pointing occurs first and speaking is delayed.

It is interesting to note the duality of the states *UsrEndSpeakOp* and *UsrSelSpeak*. Both identify a condition in which one of the modalities has reached a stable point, in the sense that the task can be completed re-starting from that state after an interrupt has occurred. This requires special attention to be paid to the presentation of the system state in order to help the recovery of the propositional goal. For example, how do users know whether or not the system has already recognised the pronounced words? and by what means can this knowledge be made persistent over a period of time?

The system component is specified using the same formalism, allowing both user and system models to be combined to study the resulting conjoint behaviour. The system is split into two automata, similar in structure, specifying the selection and speaking tasks, as shown in Fig. 3(b–c). Both are always ready to reply to user's initiated actions as well as to interrupts. In addition, the *SysSpeak* automaton defines the rate at which speech is recognised with variable $es = 1$.

Interrupt generation rate and interrupts handling is specified by the automaton of Fig. 3(d). We assume that no nested interrupts occur. The Speak 'n Drop model is the parallel composition of the four automata described above:

$$\text{Usr} \bowtie_{\{\text{move,startSpeak,click,endSpeak,drop,interrupt,clickOK}\}}$$
$$((\text{SysMouse} \bowtie_{\{\text{drop,interrupt,clickOK}\}} \text{SysSpeak}) \bowtie_{\{\text{interrupt,clickOK}\}} \text{Interrupt}),$$
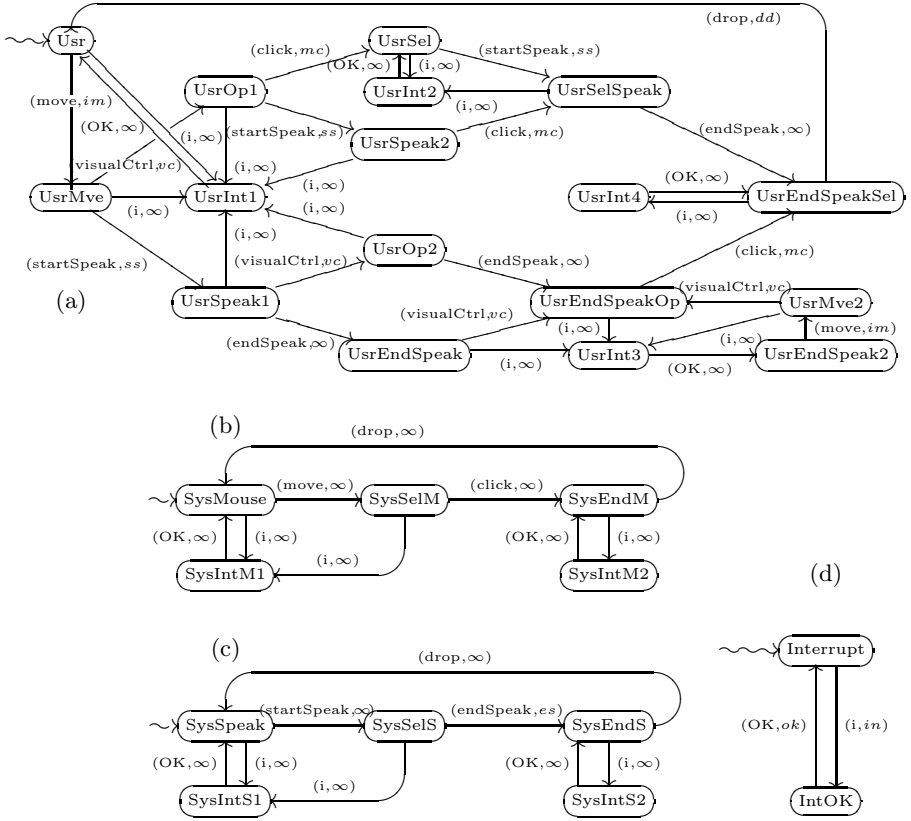
Fig. 3. Stochastic automata of (a) Usr, (b) SysMouse, (c) SysSpeak and (d) Interrupt, in which i=interrupt and OK=clickOK

where the sets of actions identify the synchronisation of automata over the named actions and ⋈ stands for the PEPA cooperation operator. Actions *interrupt* and *clickOK*, e.g., represent the occurrence of an interrupt and its handling. The above composition expresses that they may occur only with the participation (i.e. synchronisation) of *all* component automata. Action *drop* instead requires the participation of all except the Interrupt automaton, as it is not part of its synchronisation set at the cooperation operator. The resulting PEPA specification can be analysed with the stochastic model checker PRISM [13], a prototype tool that supports, e.g., the automatic verification of temporal logic properties and reward properties. One kind of reward property is particularly useful for the analysis of the effects of interrupts and will be explained in the next section. Reward formulae implicitly use reward structures that must be included in the specification to define those transitions that generate a certain amount of reward when executed. In our case, e.g., we have assigned a reward of 1 to each drop action to analyse the total number of such actions over a certain period of time.

7 Analysis Results

As an indicator of the resilience of an interaction technique to external interrupts we study the number of effective *drops* a user manages to perform during a fixed period of time under a varying number of interrupts occurring randomly during that period. The expected number of *drops* and *interrupts* occurring during an interval of 5 minutes (i.e. 300 s) are cumulative reward measures that can be formalised in reward stochastic temporal logic supported by the PRISM model checker as:

$$R\{"drops"\}_{=?}\ [C \leq 300] \qquad \text{and} \qquad R\{"interrupts"\}_{=?}\ [C \leq 300].$$

The notation $R\{\}_{=?}$ means that instead of comparing the results with a specific bound, the effective number of drops and interrupts is calculated.
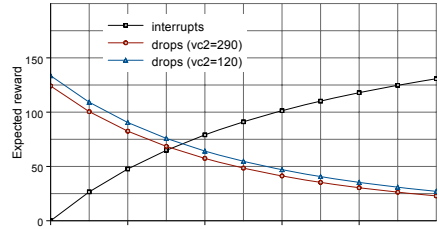
Fig. 4(f) shows the expected number of drops a user manages to perform in the presence of a number of interrupts over the time span, for the Drag 'n Drop and Speak 'n Drop interaction techniques. The effective number of interrupts at rate 1 in Fig. 4(f) is 130 and not 300 because the user needs time to handle an interrupt (i.e. moving the cursor to a button and click on it). Recall that our model assumes no new interrupts arrive while a user is still handling a previous interrupt. With an interrupt rate close to 0, the user performs on average (given the values chosen for the model's parameters) 134 drops when using Drag 'n Drop and only 109 when using Speak 'n Drop. As expected, the user's performance decreases when the interrupt rate increases. In the presence of about 130 interrupts in 300 s, the user manages to perform only 27 and 33 drops.

Although the differences are relatively small, we can nevertheless make some observations. First of all, in the absence of interrupts Drag 'n Drop leads to a higher average number of drops. This is explained by the fact that we have assumed a time of recognition of spoken words equal to $2.5 \times real\ time$ for the Speak 'n Drop model; this is a realistic assumption, but it limits the speed of interaction a user can reach. Furthermore, when the number of interrupts increases, Speak 'n Drop leads to better performance than Drag 'n Drop. The latter is more sensitive to interrupts because the total time involved in dragging is relatively long and when an interrupt occurs the user needs to start all over.

Apart from a basic comparison of the performance of the two techniques, we also investigated their sensitivity to the variation of the various action rates. In Fig. 4(a), the performance of Drag 'n Drop is shown for different user behaviour concerning the distribution of time between the various phases of a movement. As expected, when the percentage of time spent on the ballistic phase increases w.r.t. that spent on the visually-controlled phase, the performance improves slightly. However, a variation of 12% of splitting between the two phases accounts for only a 4% difference in the number of drops, showing that the model is not very sensitive to how the movement is partitioned over time in the two phases. Regarding Speak 'n Drop, the same analysis shows a negligible difference of performance: a variation of 12% in the distribution of the time accounts for only a 1% difference in the number of drops. Likewise, in Fig. 4(b) the line labelled *drops*
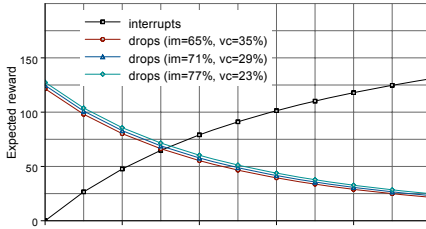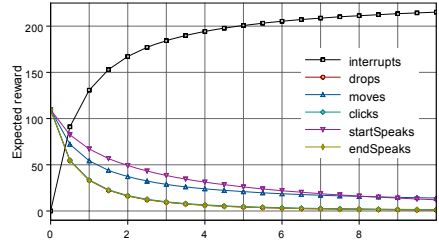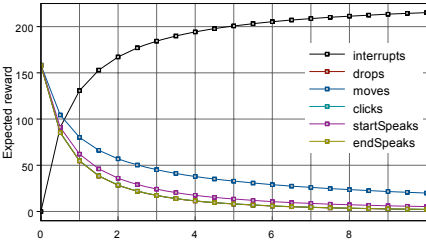
(a) Drag 'n Drop for varying behaviour.   (b) Effect of learning in Drag 'n Drop.
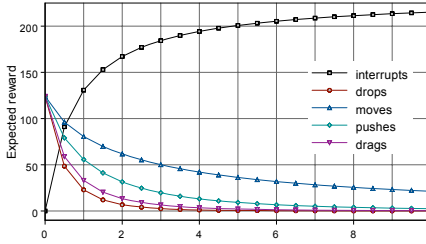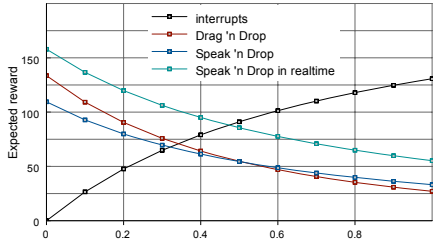
(c) Intermediate actions in Drag 'n Drop. (d) Intermediate actions in Speak 'n Drop.

(e) Intermediate actions in Speak 'n Drop (f) Drag 'n Drop vs. Speak 'n Drop (with
(with real-time speech recognition).   and without real-time speech recognition).

Fig. 4. (a) Case study; (b)–(h) Results performance analyses performed for this paper
(on the x-axes the values of the rate of interrupts used, ranging from 0 to 1 per second)

(vc2=1/0.120) shows the effect on performance of a user skilled in sustaining
the overall propositional goal, while that labelled *drops (vc2=1/0.290)* shows the
performance of a less skilled user. Clearly, in the former case the performance is
uniformly better than in the latter; this describes the effect of learning due to
frequent recurrence of the operation. According to the modelling experiments,
an already skilled user can increase her performance by 8% with such a learning
effect. This learning effect does not apply to Speak 'n Drop: there is no significant
variation in performance when the user is behaving procedurally or not.

Fig. 4(c) relates the performance of various actions to the overall performance,
i.e. it shows the total number of *move*, *push* and *drag* actions needed to obtain
the corresponding number of drops for the given number of interrupts occurring
in 300 s. About 50% of moves, 35% of pushes and 10% of drags are interrupted

not leading to a successful drop. In fact, since the Drag 'n Drop technique is sequential, clearly actions that occur towards the beginning of the sequence fail more often than those close to the end. For Speak 'n Drop, it is particularly interesting to relate the intermediate actions to the number of drops. In Fig. 4(d) two facts show up clearly. First, the number of interrupted *startSpeak* actions keeps decreasing after the other actions have reached a more or less stable number. This is explained by the relatively slow speech recognition, which forces users to adapt themselves to the performance of the system. Second, the number of failures of the *click* and *endSpeak* actions is substantially equal to the number of *drops* for any number of interrupts considered. In fact, given the scale of Fig. 4(c), those actions collapse in the bottom curve and are not distinguishable. This is explained by the parallel execution of the subtasks and the very short time that passes between their respective completion, making the occurrence of an interrupt unlikely.

From these results, one might conclude that Drag 'n Drop is better suited for low interrupt rates while Speak 'n Drop takes the lead as the number of interrupts increases. On the other hand, the Speak 'n Drop proves to be more resilient, both to the number of interrupts and to the varying performance of users. Also, taking into account the percentage of failures of intermediate actions caused by interrupts, it can be expected that Speak 'n Drop might be more appreciated by users because they experience less frustration than with Drag 'n Drop, where for many interrupts they continuously need to restart their activity.

After these observations, we repeated the analysis specifying a *real-time* speech recogniser as shown in Fig. 4(e). In this case we observe that the number of *startSpeak* actions decreases in a similar way as for the other actions, the percentage of failing *clicks* and *endSpeaks* is very low, the increase of failing actions is moderate, and speech fails less often than selection. As before, the *click*, *endSpeak* and *drop* actions cannot be distinguished in Fig. 4(e) since they collapse in the bottom curve due to the scaling factor.

Finally, under the objective assumption that simple command languages can be recognised in real time by a speech recogniser, Fig. 4(f) shows the overall performance of the interaction techniques (i.e. number of *drops*) comparing Drag 'n Drop and Speak 'n Drop, both without and with real-time speech recognition. Real-time multimodal interaction clearly gives the best results for any number of interrupts.

8   Conclusions and Future Work

In this paper we have developed syndetic stochastic models of the Drag 'n Drop and Speak 'n Drop interaction techniques and analysed their resilience to the presence of external interrupts. We have modelled the human behaviour based on the well-established cognitive theory ICS and used literature on Fitts' law to obtain timing information on task execution. The models have been specified in the process algebra PEPA and analysed with the stochastic model checker PRISM. The models produce surprisingly plausible results given their level of

abstraction and the fact that the parameter values come from published experiments each referring to different concerns of the overall user behaviour. To exactly what extent our models may serve to predict performance of human interaction in the presence of external interrupts requires further validation and is a topic for future research. This brings us to another issue: in reality human errors cannot be excluded and may clearly influence overall performance. Extending the models to cover erroneous behaviour is another interesting topic for future research. A particular challenge is to take into account the effects of cognitive load. A high cognitive load may result in more errors, but perhaps also in a slower capability of resuming the original task after an interrupt. A further topic for future research is the adaptation of the proposed methodology to the formalisms more commonly used in the HCI community, like ICO [18].

References

1. Bailey, B.P., Konstan, J.A., Carlis, J.V.: Measuring the effects of interruptions on task performance in the user interface. In: Proceedings SMC 2000, pp. 757–762. IEEE, Los Alamitos (2000)
2. Barnard, P.J.: Interacting Cognitive Subsystems: A Psycholinguistic Approach to Short-term Memory. In: Ellis, A. (ed.) Progress in the Psychology of Language, ch. 6, vol. 2, pp. 197–258. Lawrence Erlbaum, Mahwah (1985)
3. ter Beek, M.H., Faconti, G.P., Massink, M., Palanque, P.A., Winckler, M.: Resilience of Interaction Techniques to Interrupts: A Formal Model-based Approach—FULL VERSION. Technical Report, 2009-TR-001, ISTI–CNR (2009)
4. Brinksma, E., Hermanns, H., Katoen, J.-P. (eds.): EEF School 2000 and FMPA 2000. LNCS, vol. 2090. Springer, Heidelberg (2001)
5. Cades, D.M., Trafton, J.G., Boehm-Davis, D.A., Monk, C.A.: Does the difficulty of an interruption affect our ability to resume? In: Proceedings HFES 2007, Human Factors and Ergonomics Society, pp. 234–238 (2007)
6. Duke, D.J., Barnard, P.J., Duce, D.A., May, J.: Syndetic modelling. International Journal of Human Computer Interaction 13(4), 337–393 (1998)
7. Faconti, G.P., Duke, D.J.: Device models. In: Proceedings DSV-IS 1996, pp. 73–91. Springer, Heidelberg (1996)
8. Faconti, G., Massink, M.: Analysis of pointing tasks on a white board. In: Doherty, G., Blandford, A. (eds.) DSVIS 2006. LNCS, vol. 4323, pp. 185–198. Springer, Heidelberg (2007)
9. Gordon, I.E.: Theories of Visual Percept. Psychology Press (2004)
10. Hillston, J.: A Compositional Approach to Performance Modelling. Cambridge University Press, Cambridge (1996)
11. Hollnagel, E., Woods, D.D., Leveson, N. (eds.): Resilience Engineering: Concepts and Precepts. Ashgate (2006)
12. Jambon, F.: Formal modelling of task interruptions. In: Proceedings CHI 1996, pp. 45–46. ACM, New York (1996)
13. Kwiatkowska, M., Norman, G., Parker, D.: Probabilistic symbolic model checking with PRISM: A hybrid approach. In: Katoen, J.-P., Stevens, P. (eds.) TACAS 2002. LNCS, vol. 2280, pp. 52–66. Springer, Heidelberg (2002)
14. MacKenzie, I.S.: Fitts' law as a research and design tool in human-computer interaction. International Journal of Human Computer Interaction 7(1), 91–139 (1992)

15. McCrickard, D.S., Chewar, C.M.: Attuning notification design to user goals and attention costs. Communications of the ACM 46(3), 67–72 (2003)
16. McDaniel, M.A., Einstein, G.O., Graham, T., Rall, E.: Delaying execution of intentions: overcoming the costs of interruptions. Applied Cognitive Psychology 18(5), 533–547 (2004)
17. O'Conaill, B., Frohlich, D.: Timespace in the workplace: Dealing with interruptions. In: Proceedings CHI 1995, pp. 262–263. ACM, New York (1995)
18. Palanque, P.A., Bastide, R.: Petri net based Design of User-driven Interfaces Using the Interactive Cooperative Objects Formalism. In: Proceedings DSV-IS 1994, pp. 383–400. Springer, Heidelberg (1994)
19. Palanque, P.A., Bernhaupt, R., Navarre, D., Ould, M., Winckler, M.: Supporting usability evaluation of multimodal man-machine interfaces for space ground segment applications using Petri-net based formal specification. In: Proceedings SpaceOps 2006, American Institute of Aeronautics and Astronautics (2006)
20. Paternò, F., Mancini, C., Meniconi, S.: ConcurTaskTrees: A Diagrammatic Notation for Specifying Task Models. In: Proceedings INTERACT 1997, pp. 362–369. Chapman & Hall, Boca Raton (1997)
21. Pinheiro da Silva, P.: User interface declarative models and development environments: A survey. In: Palanque, P., Paternó, F. (eds.) DSV-IS 2000. LNCS, vol. 1946, pp. 207–226. Springer, Heidelberg (2001)
22. Trafton, J.G., Monk, C.A.: Task Interruptions. Reviews of Human Factors and Ergonomics 3, 111–126 (2007)
23. West, R.L., Nagy, G.: Using GOMS for Modeling Routine Tasks Within Complex Sociotechnical Systems: Connecting Macrocognitive Models to Microcognition. Journal of Cognitive Engineering and Decision Making 1(2), 186–211 (2007)
24. Zeigarnik, B.: Das behalten erledigter und unerledigter handlungen. Psychologische Forschung 9, 1–85 (1927)