THE PUBLISHING HOUSE
OF THE ROMANIAN ACADEMY

PROCEEDINGS OF THE ROMANIAN ACADEMY, Series A,
Volume 16, Number 2/2015, pp. 243–250

STOCHASTIC PROCESS CALCULI FOR HUMAN-COMPUTER INTERACTION

Gabriel CIOBANU, Armand ROTARU

Romanian Academy, Institute of Computer Science,
Blvd. Carol I no. 8, 700505, Iasi, Romania
E-mail: gabriel@info.uaic.ro

Stochastic process calculi are powerful formalisms which can be used in the specification and analysis of human-computer interaction. We present the context in which the application of these calculi is especially rewarding, and then focus on the advantages of employing them in modelling interactive systems. Advantages include being able to handle complex and very large systems, to describe human and machine behaviours within a single formalism (aided by cognitive architectures), and to select the level of detail that is most appropriate for expressing a particular interaction.

Key words: human-computer interaction, stochastic process calculus, performance measure, cognitive architecture.

1. INTRODUCTION

The application of formal methods in the analysis of human-computer interaction [27] represents a relatively recent advance, even though there are obvious advantages associated with model-based approaches, and a few of the mathematical tools used by these approaches have been around for more than 50 years. Among the major benefits of constructing formal models of interaction are being able to prove, in a rigorous manner, that an interaction has certain properties, and doing so without having to build the actual interactive system. In the field of human-computer interaction, a modelling approach has been particularly successful in studying the usability of interactive systems and the circumstances under which an interaction can break down and fail (*e.g.*, due to a previously unknown flaw in the behaviour of the interactive system), especially in the case of safety-critical systems. The advantages brought about by modelling are perhaps most visible when investigating complex systems, for which user testing can become prohibitively expensive in terms of financial cost/duration, and where the existence of subtle errors in the design of the system and the users' understanding of the interaction can go unnoticed, if the conditions that bring about those errors are improbable. Some of the formal methods that have been employed in modelling include formal grammars [29], interactors [28], abstractions [11], Petri nets [24], and non-stochastic process calculi, such as LOTOS [3] and CSP [8].

Stochastic process calculi, such as PEPA [17], Bio-PEPA [6] and PRISM [20], are another class of formalisms that have been applied in modelling. Some of the most representative and comprehensive studies in this area focus on analysing the impact of lags on workplace productivity [9], attentional limitations in human task performance [30], the effects of interruptions on workplace productivity [32], the dynamics of emergency egress [23], and performance aspects of collaborative software [33].

Up to now it appears that stochastic process calculi have been employed extensively in describing and analysing computational systems, but not human-computer interaction. We think that the lack of studies on this topic has three likely causes. Firstly, the field of human-computer interaction is interdisciplinary by its very nature, which means that novel approaches proposed by researchers in one area (*e.g.*, theoretical computer science, in the case of process calculi) may require additional time and effort before they are assimilated by researchers in other areas (*e.g.*, cognitive psychology, industrial design). Secondly, given that process calculi were created specifically for addressing issues within the context of computer science, modellers cannot immediately apply these formalisms to human-computer interaction, but instead have to

derive the process calculus model for human behaviour from a psychological model. Finally, almost all the models of human behaviour developed using stochastic process calculi have not been subjected to systematic theoretical and empirical validation.

In this context, the main purpose of our paper is to emphasize the advantages of stochastic process calculi, when it comes to modelling interactive systems. We begin with a brief, but accessible presentation of these formalisms and of their associated performance measures, oriented towards both psychologists and computer scientists. Next, we examine the situations in which the use of stochastic process calculi is justified and can make a valuable contribution to the analysis of human-computer interaction. We then tackle the main topic of our study and discuss the top advantages of a formal stochastic approach, namely the ability to:

- deal with complex and very large interactive systems,
- represent human and machine behaviours in a single formalism,
- model interaction dynamics at various levels of detail.

2. STOCHASTIC PROCESS CALCULI: A BRIEF DESCRIPTION

In many respects, stochastic process calculi are similar to non-stochastic process calculi, but with the fundamental difference that the former concentrate mainly on *quantitative* properties, such as the duration and/or the probability of a certain behaviour, whereas the latter are concerned mostly with *qualitative* properties, such as whether an undesirable form of behaviour can occur or not. The basic elements of stochastic process calculi are the possible *states* through which the system can pass, and the *transitions* that take the system from one state to another. In order to define states and transitions, stochastic process calculi usually rely on three types of operators, namely the *sequential operator*, the *choice operator*, and the *parallel operator*. The sequential operator $tr.B$ specifies that a process P, which is in state A, behaves in a deterministic manner and can only perform a single transition tr, between the current state A and the next state B. In contrast, the choice operator $tr_1.B_1 + \ldots + tr_n.B_n$ indicates that a process P can evolve in a nondeterministic manner, by performing one of a set of transitions $\{tr_i \mid 1 \leq i \leq n\}$ between the current state A and one of the states B_i, where $1 \leq i \leq n$; the resulting transition is the one with the shortest duration, according to a *race condition* (*i.e.*, the transitions "race" each other, and the "winner" is the transition with the best time). Finally, the parallel operator $P \|_L Q$ describes the manner in which two processes P and Q can interact through a set L of shared transitions. More precisely, if a transition tr_L is included in L and both P and Q can perform that transition, then the two processes synchronize on tr_L: if one process performs the transition tr_L, than the other process must do so as well. The duration of this shared transition is that of the longest of the two individual transitions, according to a *waiting condition* (*i.e.*, the first process to finish the transition has to "wait" for the other process to "catch up"). Thus, processes consist of a set of states and a set of transitions, generated by applying operators. Moreover, processes have a temporal dimension, given by the fact that each transitions has a duration, as well as a probabilistic dimension, arising from the fact that the choice and the parallel operators assign probabilities to transitions, based on the stochastic variables for transition durations, the race condition, and the waiting condition.

The states and the transitions of such a stochastic model can be represented as a *labelled continuous-time Markov chain* (CTMC), which provides the basis for virtually any kind of efficient quantitative analysis for the model. Traditionally, performance analysis deals with three types of measures, namely *steady-state*, *transient*, and *passage-time*. From a temporal point of view, the steady-state measure is applied to the entire behaviour of the model (*i.e.*, from $t_I = 0$ to $t_F = \infty$), while the remaining two measures are concerned with only a fragment of behaviour (*i.e.*, from t_I to t_F, where $0 \leq t_I < t_F < \infty$). Steady-state measures describe the long-term behaviour of the system in terms of the percentage of time that the system spends in a particular state s (or, equivalently, the probability that the system is in state s at time t, when $t \to \infty$). Transient measures evaluate the probability that the system is in state s at time t, given that it starts in state s_0. Steady-state and transient measures are complementary to each other, as both offer a description of system behaviour at time t, where t is either finite (in the transient case) or infinite (in the steady-state case). The only difference is that transient measures are defined relative to a specific initial state s_0, while steady-state measures are independent of the initial state. Finally, passage-time measures estimate the

probability that the system reaches state s before time t, again assuming that the system starts in state s_0 (which is akin to a continuous analogue of transient measures).

Given the labelled CTMC associated with a stochastic model, one can also verify *reward* (or *cost*) properties, which build upon traditional transient, steady-state, and passage-time measures. Instead of reasoning about the probabilities of certain behaviours, as is usually done when employing stochastic process calculi, it is possible to assign numerical values (*i.e.*, rewards) to the states and the transitions which form a particular behaviour, in order to compute the expected values of the rewards associated with that behaviour. For instance, if a model contains "desirable" states and "undesirable" states, a one can find the long-term probability that the system is in a desirable state by setting a reward of 1 for desirable states (and a reward of 0 for undesirable states), and then using a steady-state measure. Another example would be one in which one is interested in determining the average number of transitions that are performed in moving from a state A to a state B, which can be achieved by placing a reward of 1 on all transitions, and then using a passage-time measure. The most frequently verified reward properties can be classified into four types: *instantaneous* reward properties, *cumulative* reward properties, *reachability* reward properties, and *steady-state* reward properties. Instantaneous properties assign rewards only to states, and calculate the expected value of those rewards at time t. While instantaneous properties are based on transient measures, cumulative and reachability properties rely on passage-time measures. Moreover, the rewards involved are of a greater complexity: firstly, they refer to both states and transitions; secondly, rewards for states are now dependent on the amount of time spent in a given state (*i.e.*, if the reward for state C is equal to r, the actual reward accumulated by spending h time units in state C is equal to $h \cdot r$, meaning that r represents the rate at which the reward is obtained); thirdly, properties operate with rewards for complete behaviours, which are derived by adding together the rewards for all the states and the transitions that make up those behaviours. The difference between cumulative and reachability properties is that the former calculate the reward gained up until time t, whereas the latter calculate the reward received until reaching a certain state D. Finally, steady-state properties, like instantaneous properties, associate rewards only with states, and compute the expected value of those rewards in the long run (*i.e.*, when $t \rightarrow \infty$).

3. WHEN IS A STOCHASTIC PROCESS CALCULUS USEFUL?

Before choosing a stochastic modelling approach to describing and analysing human-computer interaction, the modeller should consider three aspects: (a) whether the interaction can be modelled by current stochastic process calculi with sufficient fidelity, (b) whether the quantitative properties that are of interest to the modeller are also compatible with the existing performance measures for stochastic models, and (c) whether the use of stochastic process calculi is strongly justified, when compared to alternative approaches, such as those employing Petri nets. Consequently, an approach based on stochastic process calculi is most effective when the interactive system has the following features:

1. a prominent stochastic character, of particular relevance to the modeller;
2. a well understood user behaviour, depending on some underlying stochastic variables;
3. several subcomponents, operating in parallel and interacting with each other.

As the first item indicates, stochastic process calculi are useful mainly in representing and investigating stochastic systems from a quantitative point of view. This means that if the system is not stochastic, or if the system is stochastic, but the quantitative properties of the system (*e.g.*, the average duration and/or probability of an interaction) are not of interest to the modeller, then a stochastic approach is not well motivated. In essence, the distinction being made here is that between *quantitative* and *qualitative* properties. Stochastic process calculi deal with the former, which include, among others, steady-state measures (*i.e.*, the probability that the system is in state A in the long run), transient measures (*i.e.*, the probability that the system is in state A at time t), and passage-time measures (*i.e.*, the probability that the system reaches state A before time t). However, if the modeller is concerned with qualitative properties, such as the absence of *deadlocks* (*i.e.*, states or groups of states from which the system cannot escape, causing erroneous behaviour), *liveness* (*i.e.*, whether a particular transition tr is enabled in state A), *reachability* (*i.e.*, whether the system eventually reaches a desirable state D, or an undesirable state U), and *causality* (*i.e.*, whether

being in state *A* is a necessary/sufficient condition for later entering state *B*), then stochastic process calculi are not the most appropriate modelling choice, and the modeller should use non-stochastic formalisms.

The second item refers to the fact that human behaviour is typically more complex and more difficult to formalize than the operation of software and hardware components. While a computer scientist can simply look up the technical manual for a computational system (or the performance logs recorded by the system) and create an accurate stochastic model, the cognitive psychologist seeking to represent human behaviour is constrained by the existing body of psychological theory and experimental results. Consequently, it is crucial that the mental processes and physical movements that form the task behaviour can be expressed rigorously in terms of either a comprehensive cognitive architecture, or a more specific psychological theory that applies to the task at hand. Furthermore, as the focus of the modelling approach falls on quantitative elements of behaviour, it is necessary to describe the stochastic variables that are associated with task behaviour, such as the duration of a particular hand movement, or the probability of attending to a certain visual stimulus.

The last item requires that the interaction between the components of an interactive system should be a key element in defining the behaviour of the system (*e.g.*, as is the case with most scenarios involving multi-agent systems), given that stochastic process calculi are designed specifically for modelling concurrent systems. Ideally, interactions should occur at all levels within the system, from the lowest, which is that of psychological processes and electronic/mechanical parts, to the highest, which consists of (groups of) users and computational devices. If meaningful interactions are mostly absent, then representing the system in the framework of stochastic process calculi can become a complex and time-consuming operation.

4. ADVANTAGES OF STOCHASTIC PROCESS CALCULI

4.1. MODELLING COMPLEX AND LARGE INTERACTIVE SYSTEMS

Stochastic process calculi have several features that facilitate the modelling of complex, large-scale interactive systems. The key strength of these formalisms, when it comes to handling complexity, is that elaborate systems can be easily represented through the composition of simple syntactic constructs, by using the parallel operator and sets of shared transitions. *Compositionality* is a defining element, as it has far reaching consequences for model design and analysis. To begin with, the dynamics of stochastic process calculus models is built on the notion of interaction: a model is not a single level, monolithic structure, but a multi-level, hierarchical interplay of components, subcomponents, and so on. Thus, cooperation between components allows for intricate models to be built almost effortlessly, either bottom-up (*i.e.*, starting from basic states and transitions between states, in an incremental fashion, and eventually generating the entire model) or top-down (*i.e.*, breaking down high-level components into low-level subcomponents, until no further decomposition is possible or necessary). In addition, the hierarchical architecture of most models facilitates their analysis, and it allows one to move freely from one level of detail to another, for each individual component.

Another advantage of stochastic process calculi is that they are *intuitive* formalisms: the syntax is made up of states and transitions, defined in terms of only a very limited number of operators, and the semantics follows immediately from the syntax. Furthermore, models can be represented graphically as stochastic automata, which means that the semantics of a model can be illustrated *visually* without making any direct reference to the syntax of that model. Taken together, compositionality and elegance (in terms of syntax and semantics) guarantee that stochastic models can be created and studied with relatively little effort, by researchers from the various subfields of human-computer interaction, including those not traditionally associated with theoretical computer science, such as cognitive psychology.

With respect to model size, stochastic process calculi afford several solutions for dealing with very large models (*i.e.*, models having over 10^{10} states and transitions). One solution is to use *behavioural equivalences* between models. More specifically, given an initial model *A*, these formal instruments allow one to build the smallest model *B* that is equivalent to *A*, according to an appropriate notion of behavioural equivalence. The main concept underlying this transformation is that of state aggregation, which involves grouping together states that have matching behaviours and focusing on transitions between groups of

equivalent states, instead of transitions between individual states. Through state aggregation, a clear correspondence can be made between the labelled CTMCs for A and B, meaning that any performance measure defined over A can be derived effortlessly from a corresponding performance measure computed over B. Quite frequently, the minimised model B is much smaller than the initial model A, in terms of the total number of states and transitions. For instance, in [16] it is shown how a model of a telephone system, consisting of more than 10 million states, can be reduced to an equivalent system of about 700 states, which is considerably easier to analyse. A review of equivalence-based minimisation methods can be found in [15].

Another solution is to employ a *fluid flow* approach to performance evaluation [18]. Such an approach is applicable in situations where the considerable size of the model is produced not by the interaction of a relatively small number of very complex processes, but instead by the interaction of a great number of repeated, structurally simple processes. In the latter case, one might be interested in determining the average number of processes that are in a given state, at time t, where t is either finite (like in transient measures) or infinite (like in steady-state measures). When this happens, the behaviour of the model can be expressed through a system of *ordinary differential equations* (ODEs). Importantly, the size of the ODE system does not depend on the number of sub-processes that form the model, which makes it possible to investigate the properties of arbitrarily large models.

Yet another course of action is to rely on *truncation* techniques for quantitative model checking [14]. If one is concerned with transient or passage-time measures in the short run (*i.e.*, properties regarding the behaviour of the model before time t, where t must be finite), then there is the option of examining only a limited subset of the model's entire state space. More precisely, given a property P and a model M, a truncation algorithm constructs and examines the labelled CTMC for M in an on-the-fly, breadth-first manner, until a highly accurate evaluation of the validity of P has been reached. An outstanding advantage of truncation-based analysis is that it can be applied to models having infinite state spaces.

4.2. INTEGRATING THE HUMAN USER AND THE COMPUTATIONAL DEVICE

Stochastic process calculi are expressive enough to allow the modeller to capture both the human and the computational aspects of interactive systems. As such, it is no longer necessary to have one formalism each for describing human behaviour, representing the operation of the computational system, and constructing the interface between the user model and the system model. In order to produce a formal model of human task behaviour, a convenient solution is to start from an appropriate cognitive architecture. Also, it is preferable to have a cognitive architecture which is distributed and in which the interactions between its components have a prominent role, to facilitate the derivation of stochastic models that implement the behaviour being analysed. One such architecture is ICS [1, 2], which models cognition in terms of a modular, hierarchical system of psychological processes, whose primary purposes are the generation, transformation, storage and retrieval of mental representations. Unfortunately, the focus of ICS is on qualitative aspects of behaviour, and not on their quantitative characterization (*i.e.*, duration and probability). However, this lack of quantitative data can be remedied by importing the necessary performance variables from another cognitive architecture, namely the MHP [4, 5].

The MHP was created specifically for approximating the temporal (and, to a lesser degree, probabilistic) properties of human behaviour in tasks which involve human-computer interaction. It can be described as a parallel information-processing system, consisting of "memories" and "processors", as well as a set of psychological laws, referred to as the "principles of operation". In the case of covert, mental processes, their duration is derived from parameters associated with the memories and the processors. For any memory, these parameters are μ, the size or storage capacity (*i.e.*, the maximum number of items that can be stored), δ, the rate of decay (*i.e.*, the half-life of a memory item, which is the time after which the probability of retrieving that item is less than 50%), and κ, the type of information that it stores. For any processor, the only parameter that is crucial is τ, the cycle time (*i.e.*, the amount of time necessary for analysing a unit of information). On the other hand, the duration of overt, physical actions is computed based on Fitts's law [22]. The model rests on a solid empirical base (*i.e.*, the parameters of the model are psychologically plausible, and are derived from a vast set of experimental studies), and has been validated repeatedly in practice.

Instead of using two cognitive architectures, which increases the complexity of the modelling enterprise, it is also possible to employ a single architecture, such as the QN-MHP [21, 35], which is inspired by the MHP and has all the desirable features of ICS. The QN-MHP builds upon the theory behind the MHP by adding another level of structure to the description of the perceptual, cognitive, and motor subsystems, as well as by incorporating a number of findings from the field of neuroscience. In addition, the focus of modelling shifts from describing single tasks, to representing sets of concurrent tasks. Therefore, the QN-MHP also addresses the problem of how the limited set of available cognitive resources is divided among competing tasks.

4.3. CHOOSING THE LEVEL OF DETAIL FOR TRANSITION DURATIONS

An advantage of stochastic process calculi is the flexibility they allow in selecting the level of detail at which the duration of transitions is represented. Typically, one is interested mainly in the average duration. In such cases, it is possible to use a *Markovian* process calculus [15], by assuming that the duration of each transition is drawn from an exponential distribution, and then choosing a rate that produces the desired average duration. The resulting system is a labelled CTMC, for which a solid mathematical theory exists [26], thus greatly facilitating performance analysis [19]. Nevertheless, it might sometimes happen that one wishes to go beyond average durations, being concerned with the nature of the distributions for the durations. For instance, such a situation might occur for interactive systems that require the accurate modelling of human behaviour, for which performance variables usually follow a normal or log-normal distribution [31, 9]. This does not pose a serious problem for stochastic process calculi, as there are at least two readily available solutions, namely phase-type distributions and non-Markovian formalisms.

The first solution involves employing Markovian process calculi in approximating the generalised distributions for durations. For instance, suppose that we have a transition whose duration is drawn from an arbitrary probability distribution. The aforementioned transition (and its associated probability distribution) can be approximated by a more complex Markovian process, consisting mainly of intermediate states and transitions. In this approach, (non-exponential) generalised distributions are replaced by phase-type distributions, which can provide arbitrarily accurate approximations for any positive-valued distribution [25]. Moreover, all the benefits related to the analysis of labelled CTMCs are retained. The derivation of phase-type approximations, in the context of PEPA [17], is supported by the software application EMPEPA [12], which implements several expectation-maximization algorithms for fitting a phase-type distribution to any (user-supplied) discrete probability distribution.

The second solution is based on the use of *generalised* (*i.e*, *non-Markovian*) stochastic process calculi and *discrete event simulation* techniques. Unlike Markovian calculi, generalised calculi can represent generalised distributions directly, yielding models that are expressed as *generalised semi-Markov processes* (GSMPs). The theory for GSMPs [13] is far less developed than that for CTMCs, which means that performance measures typically cannot be derived analytically. Instead, performance analysis is conducted by simulating and examining the behaviour of the resulting model: for any property of interest, multiple executions of the model are generated and tested, until the validity of the property can be determined with a high degree of confidence. Software support for statistical model checking is provided by the tool YMER [36], which operates over a stochastic language similar to that of PRISM [20].

5. CONCLUSION

The use of stochastic process calculi for modelling human-computer interaction is a relatively recent approach, which is beginning to prove its utility. In order to highlight its mostly unexplored potential, we provide a gentle non-technical introduction to stochastic modelling with process calculi, aimed at researchers from the various disciplines subsumed by the field of human-computer interaction, and especially at those outside the area of theoretical computer science. Our preference for these formalisms is motivated by the fact that they have a simple syntax, an intuitive semantics, a wide set of tractable performance measures that can be computed, and several user-friendly, well-documented software tools [34, 10, 7, 20] that support model building and evaluation. These tools include features such as syntax highlighting, static analyses (*i.e*.,

verifying whether the model is both syntactically and semantically correct), visual state space exploration, interactive simulation (*i.e*., a user-guided, step-by-step simulation of model behaviour), saving the results of the performance analyses as images and/or data files, and exporting the labelled CTMC underlying the model into various formats.

Based on the characteristics of the available formalisms, interactive systems which can be easily represented and analysed by employing stochastic process calculi usually involve a well-documented and relevant stochastic behaviour, a clear understanding of human task behaviour, and an important role for the interaction between subcomponents. There are at least three major advantages that recommend a formal stochastic approach. Firstly, multiple options are available for modelling complex and very large interactive systems. Some of these options (*i.e*., compositionality and behavioural equivalences) are inherited from non-stochastic calculi, while others (*i.e.*, fluid flow and truncation analyses) come from translating stochastic models into labelled CTMCs. Secondly, human and machine behaviour can be captured in an uniform manner, within a single formalism. Thirdly, one has the freedom of choosing how faithful the representation of an interaction must be, in terms of the durations for the transitions that make up the interaction (*i.e*., Markovian versus non-Markovian approaches). In other words, one can specify only mean durations, or instead decide to accurately describe the temporal distributions from which the durations are drawn.

REFERENCES

1. P. BARNARD, *Interacting cognitive subsystems: A psycholinguistic approach to short term memory*, in A. Ellis (Editor), *Progress in the Psychology of Language*, Lawrence Erlbaum Associates, Hillsdale, NJ, 1985, pp. 197–258.
2. P. BARNARD, *Interacting Cognitive Subsystems: Modeling working memory phenomena within a multi-processor architecture*, in A. Miyake and P. Shah (Editors), *Models of Working Memory*, Cambridge University Press, Cambridge, United Kingdom, 1999, pp. 298–339.
3. H. BOWMAN, G. FACONTI, *Analysing cognitive behaviour using LOTOS and Mexitl*, Formal Aspects of Computing, **11**, *2*, pp. 132–159, 1999,
4. S. CARD, T. MORAN, A. NEWELL, *The Psychology of Human-Computer Interaction*, Lawrence Erlbaum Associates, Hillsdale, NJ, 1983.
5. S. CARD, T. MORAN, A. NEWELL, *The Model Human Processor*, in K. Boff, L. Kaufman, and J. Thomas (Editors), *Handbook of Perception and Human Performance*, Vol. 2, Wiley-Interscience, New York, NY, 1986, pp. 45:1–45:35..
6. F. CIOCCHETTA, J. HILLSTON, *Bio-PEPA: A framework for the modelling and analysis of biological systems*, Theoretical Computer Science, **410**, *33–34*, pp. 3065–3084, 2009.
7. A. CLARK, S. GILMORE, M. GUERRIERO, P. KEMPER, *On verifying Bio-PEPA models*, in P. Quaglia (Editor), Proceedings of the 8th International Conference on *Computational Models in Systems Biology*, New York, NY, 2010, ACM, pp. 23–32.
8. A. DEGANI, M. HEYMANN, *Formal verification of human-automation interaction*, Human Factors, **44**, *1*, pp. 28–43, 2002.
9. G. DOHERTY, M. MASSINK, G. FACONTI, *Reasoning about interactive systems with stochastic models*, in C. Johnson, (Editor), Proceedings of the 8th International Workshop on *Interactive Systems – Design, Specification, and Verification*, Springer-Verlag, London, United Kingdom, 2001, pp. 144–163.
10. A. DUGUID, S. GILMORE, M. GUERRIERO, J. HILLSTON, L. LOEWE, *Design and development of software tools for Bio-PEPA*, in A. Dunkin, R. Ingalls, E. Yucesan, M. Rossetti, R. Hil, and B. Johansson (Editors), Proceedings of the Winter Simulation Conference, 2009, pp. 956–967.
11. M. B. DWYER, V. CARR, L. HINES, *Model checking graphical user interfaces using abstractions*, in M. Jazayeri and H. Schauer (Editors), Proceedings of the 6th European Software Engineering Conference, Springer-Verlag, New York, NY, 1997, pp. 244–261.
12. N. GEISWEILLER, *Documentation of EMPEPA*, 2006. Retrieved on January 23, 2014, from http://empepa.sourceforge.net/empepa-doc.pdf.
13. P. GLYNN, *A gsmp formalism for discrete event systems*, Proceedings of the IEEE, **77**, *1*, pp. 14–23, 1989.
14. E. HAHN, H. HERMANNS, B. WACHTER, L. ZHANG, *INFAMY: An infinite-state markov model checker*, in A. Bouajjani and O. Maler (Editors), Proceedings of the 21st International Conference on *Computer Aided Verification*, Springer, Berlin, Germany, 2009, pp. 641–647.
15. H. HERMANNS, U. HERZOG, J.-P. KATOEN, *Process algebra for performance evaluation*, Theoretical Computer Science, **274**, 1, pp. 43–87, 2002.
16. H. HERMANNS, J.-P. KATOEN, *Automated compositional Markov chain generation for a plain-old telephone system*, Science of Computer Programming, **36**, 1, pp. 97–127, 2000.
17. J. HILLSTON, *A Compositional Approach to Performance Modelling*, Cambridge University Press, Cambridge, United Kingdom, 1996.
18. J. HILLSTON, *Fluid Flow Approximation of PEPA models*, Proceedings of the Second International Conference on *Quantitative Evaluation of Systems*, IEEE Computer Society Press, 2005, pp. 33–42.

19. M. KWIATKOWSKA, G. NORMAN, D. PARKER, *Stochastic Model Checking*, in M. Bernardo and J. Hillston (Editors), *Formal Methods for Performance Evaluation*, Springer-Verlag, Berlin, 2007, pp. 220–270.
20. M. KWIATKOWSKA, G. NORMAN, D. PARKER, *PRISM 4.0: Verification of Probabilistic Real-time Systems*. In G. Gopalakrishnan and S. Qadeer (Editors), Proceedings of the 23rd International Conference on *Computer Aided Verification*, Springer-Verlag, Berlin, 2011, pp. 585–591.
21. Y. LIU, R. FEYEN, O. TSIMHONI, *Queuing Network-Model Human Processor (QN-MHP): A computational architecture for multitask performance in human-machine systems*, ACM Transactions on Computer-Human Interaction, **13**, pp. 37–70, 2006.
22. I. MACKENZIE, *Fitts' law as a research and design tool in human-computer interaction*, Human Computer Interaction, **7**, pp. 91–139, 1992.
23. M. MASSINK, D. LATELLA, A. BRACCIALI, M. HARRISON, J. HILLSTON, *Scalable context-dependent analysis of emergency egress models*, Formal Aspects of Computing, **24**, *2*, pp. 267–302, 2012.
24. D. NAVARRE, P. PALANQUE, F. PATERNO, C. SANTORO, R. BASTIDE, *A tool suite for integrating task and system models through scenarios*, in C. Johnson (Editors), Proceedings of the 8th International Workshop on *Interactive Systems – Design, Specification, and Verification*, Springer-Verlag, London, 2001, pp. 88–113.
25. R. NELSON, *Probability, Stochastic Processes, and Queueing Theory*, Springer-Verlag, New York, NY, 1995.
26. J. NORRIS, *Markov chains*, Cambridge University Press, Cambridge, 1998.
27. F. PATERNO, *Model-Based Design and Evaluation of Interactive Applications*, Springer-Verlag, London, 2000.
28. F. PATERNO, G. FACONTI, *On the use of LOTOS to describe graphical interaction*, in A. Monk, D. Diaper, and M. Harrison (Editors), Proceedings of the 7th Conference of the British Computer Society Human Computer Interaction Specialist Group, Cambridge, MA, 1992, Harvard University Press, pp. 155–173.
29. P. REISNER, *Formal grammar and human factors design of an interactive graphics system*, IEEE Transactions on Software Engineering, **SE–7**, *2*, pp. 229–240, 1981.
30. L. SU, H. BOWMAN, P. BARNARD, B. WYBLE, *Process algebraic modelling of attentional capture and human electrophysiology in interactive systems*, Formal Aspects of Computing, **21**, *6*, pp. 513–539, 2009.
31. A. SWAIN, H. GUTTMANN, *Handbook of human reliability analysis with emphasis on nuclear power plant applications - final report*, Technical Report, Division of Facility Operations, Office of Nuclear Regulatory Research, US Nuclear Regulatory Commission, Washington, DC, 1983. Retrieved on January 23, 2014, from http://pbadupws.nrc.gov/docs/ML0712/ML071210299.pdf.
32. M. TER BEEK, G. FACONTI, M. MASSINK, P. PALANQUE, M. WINCKLER, *Resilience of Interaction Techniques to Interrupts: A Formal Model-Based Approach*, in T. Gross, J. Gulliksen, P. Kotze, L. Oestreicher, P. Palanque, R. Prates, and M. Winckler (Editors), Proceedings of the 12th IFIP TC13 – Conference on *Human-Computer Interaction*, Springer-Verlag, Berlin, 2009, pp. 494–509.
33. M. TER BEEK, S. GNESI, D. LATELLA, M. MASSINK, M. SEBASTIANIS, G. TRENTANNI, *An Application of Stochastic Model Checking in the Industry: User-Centered Modeling and Analysis of Collaboration in Thinkteam®*, in S. Gnesi and T. Margaria (Editors), *Formal Methods for Industrial Critical Systems: A Survey of Applications*, John Wiley & Sons, Hoboken, NJ, 2013, pp. 179–203.
34. M. TRIBASTONE, A. DUGUID, S. GILMORE, *The PEPA Eclipse Plugin*, Performance Evaluation Review, **36**, *4*, pp. 28–33, 2009.
35. C. WU, Y. LIU, *Queuing network modeling of transcription typing*, ACM Transactions on Computer-Human Interaction, **15**, *1*, pp. 1–45, 2008.
36. H. YOUNES, *Ymer: A statistical model checker*, in K. Etessami and S. Rajamani (Editors), Proceedings of the 17th International Conference on *Computer Aided Verification*, Springer-Verlag, Berlin, 2005, pp. 429–433.

Received January 23, 2014