Towards quantitative evaluation of stochastic pharmacy workflows

Herbert, Luke Thomas; Sharp, Robin

Published in:
Proceedings of the 23rd Nordic Workshop Programming Theory

Publication date:
2011

Towards quantitative evaluation of stochastic pharmacy workflows

Luke Herbert
Technical University of Denmark
Lyngby, Denmark
lthhe@imm.dtu.dk

Robin Sharp
Technical University of Denmark
Lyngby, Denmark
robin@imm.dtu.dk

1 Introduction

European hospitals are being placed under growing pressure to deliver efficiency gains due to an increasing elderly population and ever tighter financial constraints. A significant proportion of the treatment performed relies on the adaptive application of various intravenous medicine which is mostly in liquid form and which is prepared in central pharmacies serving a number of departments. Due to strict safety requirements within pharmacies, the process of preparation of fluid medicine for intravenous use is; labour intensive, inefficient[3], error-prone[1, 6] and poses health risks to medical staff. Current product preparation workflows rely on production techniques largely abandoned by other industries; including duplicating inventory, inflexible batch processing, and over production where doses are often prepared in anticipation of a need or demand that may end up not being realised which contributes to waste[7].

A solution to these issues has been sought in the automation of the medication management and dispensing processes. Deployment of systems for this purpose has delivered significant improvements in safety for both staff and patients[2, 10], and new pharmacy capabilities such as customised medicine have become feasible[2]. However, the technology has proven disruptive to traditional pharmacy processes and many potential efficiency gains have yet to be realised.

1.1 Contribution

In this paper, we present the first steps towards addressing these issues by developing a method to model and analyse pharmacy and automated workflows and their interaction so as to be able to accurately provision an effective automated solution. This is achieved by extending a well developed modelling formalism (BPMN)[5], in use in healthcare, to include timing and stochastic data (section 2). We present an algorithm for the conversion of this formalism into a format amenable to *stochastic model checking*, which allows for the calculation of a wide range of system properties (section 3). This work will serve as a basis for the development of software tools implementing the method and further theoretical work aimed at automatic synthesis of process configurations (section 4).

2 BPMN

Business Process Model and Notation (BPMN)[5] is a graphical notation for specifying business processes. The primary goal of BPMN is to provide a notation that is readily understandable by all business users. BPMN's ability to serve as a standardized bridge between business process design and implementation has lead to widespread adoption in the healthcare industry[16, 14] where its design goals have allowed for precise description of hospital workflows. Specific case studies[13, 15, 12] have underlined this, developing models of various complex hospital workflows rapidly and allowing effective manual restructuring of process flows.

2.1 Business Process Diagrams (BPD)

Modelling a workflow in BPMN involves composing a number of BPMN elements into a single *business process diagram* (BPD). For the purposes of this extended abstract we will consider BPDs restricted to a minimal subset of BPMN elements with two extensions to allow for timing and stochastic branching. Although this omits traditional data-based control flow, it is sufficient to illustrate a method to perform performance analysis of the resulting models using techniques from stochastic model checking.

Definition 1 (Minimal BPD)**.** *A minimal BPD is a tuple* $\mathcal{BPD} = (\mathbf{O}, \mathcal{F})$ *where* $\mathbf{O}$ *is a set of nodes (corresponding to BPMN objects) and* $\mathcal{F} \subseteq \mathbf{O} \times \mathbf{O}$ *is an edge relation (corresponding to BPMN flows). The set* $\mathbf{O}$ *can be partitioned into 7 disjoint subsets;* start *events* $\mathbf{E^S}$*,* end *events* $\mathbf{E^E}$*,* trigger *events* $\mathbf{E^T}$*, catch events* $\mathbf{E^C}$*,* $\mathbf{G^B}$ *a set of branch gateways where each element* $g \in \mathbf{G^B}$ *has an associated* rate distribution function R_g*,* $\mathbf{G^J}$ *a set of join gateways which merge control paths and* $\mathbf{T}$ *a set of tasks, where each element* $t \in \mathbf{T}$ *has an associated stochastic time delay expressed as a normal distribution function* $N_t(\mu_t, \sigma_t^2)$*.*

A $\mathcal{BPD}$ describes a number of business processes, where for each process $\mathcal{F}$ defines a directed graph with nodes which are elements of $\mathbf{O}$. For each node $o \in \mathbf{O}$ the input nodes of o are $\mathsf{in}(o) = \{x \in \mathbf{O} | x \mathcal{F} y\}$ and output nodes of o are $\mathsf{out}(o) = \{y \in \mathbf{O} | x \mathcal{F} y\}$. Branch gateways assign a rate to each outflow indicating how often the flow of control takes this path and thus controlling execution. This definition of *BPDs* allows for graphs which are unconnected, do not have start or end elements or various other properties which place them outside what is permitted in BPMN models. We therefore define:

Definition 2 (Well-formed minimal BPD)**.** *A* $\mathcal{BPD}$ *is well-formed if the following conditions hold:*

$$
\begin{array}{ll}
\textbf{\textit{WF1:}}\ \forall e \in \mathbf{E^S} : \mathsf{in}(e) = \varnothing \wedge |\mathsf{out}(e)| = 1 & \textbf{\textit{WF2:}}\ \forall e \in \mathbf{E^E} : \mathsf{out}(e) = \varnothing \wedge |\mathsf{in}(e)| = 1 \\
\textbf{\textit{WF3:}}\ \forall t \in \mathbf{T} : |\mathsf{out}(t)| = 1 \wedge |\mathsf{in}(t)| = 1 & \textbf{\textit{WF4:}}\ \forall g \in \mathbf{G^B} : |\mathsf{in}(g)| = 1 \wedge |\mathsf{out}(g)| > 1) \\
\textbf{\textit{WF5:}}\ \forall g \in \mathbf{G^B} : \sum_{\forall x_i \in \mathsf{out}(g)} R_g(x_i) = 1 & \textbf{\textit{WF6:}}\ \forall g \in \mathbf{G^J} : |\mathsf{in}(g)| > 1 \wedge |\mathsf{out}(g)| = 1) \\
\textbf{\textit{WF7:}}\ \forall o \in \mathbf{O}, \exists (s,e) \in \mathbf{E^S} \times \mathbf{E^E} : s \mathcal{F}^* o \wedge o \mathcal{F}^* e &
\end{array}
$$

where $\mathcal{F}^*$ *is the reflexive transitive closure of* $\mathcal{F}$*.*

The first two conditions **WF1** and **WF2** simply state that start and end states do not have respectively in or out flows and are followed/preceded by a single state. **WF3** ensures tasks do not branch control flow. **WF4 WF6** ensure that split and join of control flows actually split/join flow. **WF5** ensures that the rates of all branches are defined and hence no branch gateway can be reached from which a further choice is not possible. Finally **WF7** ensures that all objects lie on a path from a start to an end event. We will only consider *well-formed minimal BPDs*. However, it should be noted that this language has features that cover the vast majority of the core BPMN constructs with only data based control flow being absent. Many elements of data-based control flow can, however, be simulated in well-formed minimal BPDs using message passing and dummy processes.

2.2 BPMN Semantics

BPMN is a visual notation and while the BPMN specification[5] provides extensive syntactic rules, the semantics of BPMN is only given in narrative form using a somewhat inconsistent terminology. A number of papers have undertaken the task of providing formal semantics in the form of Petri nets[4], Business Process Execution Language (BPEL),[11] and Communicating Sequential Processes (CSP) [17].

In the work introduced in this paper we adopt the method for deriving a CSP semantics for a BPMN fragment given in [17] to our extended subset of BPMN. The basic idea of determining the semantics of BPMN is as follows: an abstracted BPMN syntax is expressed in the Z notation and then a semantic function converts this to a parallel composition of CSP processes corresponding to states in the diagram. These processes are themselves built up from smaller predefined CSP processes used as building blocks. This development is verbose but quite straightforward and can be used in our case without dramatic modification other than accommodating timing, in the form of a single global clock, and a slight modification of non-deterministic choice to accommodate stochastic rates.

3 Stochastic Model Checking

The main goal with this work is to be able to perform stochastic model checking of BPMN models. Specifically we wish to derive properties of the form:

- **Transient and steady-state probabilities** e.g. the probability that the system operational at time instant t or the overall probability that it is operational.
- **Timing, occurrence and ordering of events** e.g. the probability that a failure of component *B* (if it occurs) happens before any failure of component *A*.

- **Reward-based properties** e.g. the throughput of the system, i.e. the expected steady-state rate of job completion.
- **Best- and worst-case scenarios** e.g. the best-case instantaneous availability of the system at time t, starting from any initial configuration.

3.1 Conversion

Conversion of *well-formed minimal BPDs* to a specific model checking format follows broadly the process used for a wide range of model checkers as all require a formal language input. In this case we will outline our conversion to the PRISM language format[8].

We begin by decomposing a BPD into processes. We then traverse each process from its unique start node to various end nodes building a graph-like data structure in a fashion similar to[11]. Branching in the simple cases presented here maps directly to the PRISM language.

We deal with tasks $t \in \mathbf{T}$ by creating two states for a task: one before and one after the task. The first state functions as a branching gateway with a number of edges generated to the second sate which functions as a join gateway. The number of edges generated can be chosen during the conversion process by dividing the distribution into the required number of intervals, each edge has a time delay equal to the centre of the interval and a branch probability given as $p = Pr[a < X < b]$ where a and b are the bounds of the interval. (see figure 1)

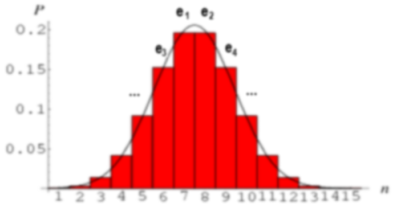

Figure 1: Task edge generation

It should be noted that our conversion process allows for a great deal of tuning, making it possible to produce models of varying complexity and with a wide range of annotations.

4 Conclusions

In this brief introduction we have extended the BPMN modelling formalism, widely used in the modelling of business operations, to allow for the recording of variable task timings and stochastically branching control flow. We have outlined a means to determine the semantics of such models and outlined how these models can be converted to a format suitable for verification by model checking. This abstract omits many details of the methods being developed and seeks to demonstrate results for a very limited subset of BPMN. It should be stressed that a fuller presentation of this work would present a complete semantics for a larger timed stochastic BPMN fragment, which consequently also would allow for a more extensive description of the method of conversion to a model checkable system description. This work is ongoing, and the theoretical developments described are accompanied by the development of software tools making use of the PRISM model checker[9].

4.1 Future work

The ultimate goal of this work is to investigate the following types of synthesis problems within the context given.

- Given a timed BPMN workflow, including probabilistic branching, calculate the next sequence of actions to be taken for all states in the systems to obtain the optimum of some parameter. (E.g. Given a medical robot interacting with pharmacists required to perform a given list of tasks in a minimum amount of time, what action should it take at each stage depending on the outcome of its own, unpredictable, operations.).
- Given a timed BPMN workflow, and a new process to be added to this workflow calculate the optimal configuration of the new combined workflow, with respect to some property of system, such as time. (E.g. How would be it best to reorganise a pharmacy workflow once a robotic system was introduced).
- Given an existing timed BPMN workflow and a multi-set of new processes that may be added, find the optimal choice from this multi-set to achieve the smallest/largest value of a parameter of interest. (E.g. Determine what selection of robotic sub-modules interacting with an existing workflow will achieve a reduction in drug production time/cost).

References

[1] C. A. Bond, C. L. Raehl, and T. Franke. Medication errors in united states hospitals. *The Journal of Human Pharmacology and Drug Therapy*, 21(9):1023–1036, September 2001.

[2] J. Carmenates and M. R. Keith. Impact of automation on pharmacist interventions and medication errors in a correctional health care system. *American Journal of Health-System Pharmacy*, 59(9):779–783, May 2001.

[3] A. Colquhoun. Could automation improve efficieny and help pharmacies with cost saving? *The Pharmaceutical Journal*, 285:587–591, November 2010.

[4] R. M. Dijkman, M. Dumas, and C. Ouyang. Formal semantics and analysis of bpmn process models. 2007.

[5] O. M. Group. *Business Process Model and Notation (BPMN) 2.0*. Object Management Group, Needham MA, USA, 2011.

[6] P. Y. Han, I. D. Coombes, and B. Green. Factors predictive of intravenous fluid administration errors in australian surgical care wards. *Quality and safety in health care*, 14:179–184, 2004.

[7] B. L. Hintzen, S. J. Knoer, C. J. V. Dyke, and B. S. Milavitz. Effect of lean process improvement techniques on a university hospital inpatient pharmacy. *American Journal of Health-System Pharmacy*, 66(22), November.

[8] M. Kwiatkowska, G. Norman, and D. Parker. Prism: probabilistic model checking for performance and reliability analysis. *SIGMETRICS Perform. Eval. Rev.*, 36:40–45, March 2009.

[9] M. Kwiatkowska, G. Norman, and D. Parker. PRISM 4.0: Verification of probabilistic real-time systems. In G. Gopalakrishnan and S. Qadeer, editors, *Proc. 23rd International Conference on Computer Aided Verification (CAV'11)*, volume 6806 of *LNCS*, pages 585–591. Springer, 2011.

[10] S. Oswald and R. Caldwell. Dispensing error rate after implementation of an automated pharmacy carousel system. *American Journal of Health-System Pharmacy*, 64(13):1427–1431, July 2007.

[11] C. Ouyang, M. Dumas, and A. H. M. T. Hofstede. Pattern-based translation of bpmn process models to bpel web services. *International Journal of Web Services Research (JWSR)*, 5(1):42–62, 2007.

[12] J. Puustjärvi and L. Puustjärvi. Automating the coordination of electronic prescription processes. In *2006 8th International Conference on e-Health Networking, Applications and Services (Healthcom 2006)*, pages 147–151, August 2006.

[13] J. Puustjärvi and L. Puustjärvi. Automating the dissemination of information entities to healthcare professionals. In B. Papasratorn, W. Chutimaskul, K. Porkaew, and V. Vanijja, editors, *Advances in Information Technology*, volume 55 of *Communications in Computer and Information Science*, pages 123–132. Springer Berlin Heidelberg, 2009.

[14] A. A. Rad, M. Benyoucef, C. E. Kuziemsky, and A. A. Rad. An evaluation framework for business process modeling languages in healthcare. *J. Theor. Appl. Electron. Commer. Res.*, 4:1–19, August 2009.

[15] M. G. Rojo, E. Rolon, L. Calahorra, F. O. Garcia, R. P. Sanchez, F. Ruiz, N. Ballester, M. Armenteros, T. Rodriguez, and R. M. Espartero. Implementation of the business process modelling notation (bpmn) in the modelling of anatomic pathology processes. In *Proceedings of the 9th European Congress on Telepathology and 3rd International Congress on Virtual Microscopy*, volume 3 (Suppl 1), London, UK, July 2008. BioMed Central Ltd.

[16] E. Rolón, F. García, F. Ruíz, M. Piattini, and L. Calahorra. Healthcare process development with bpmn. In S. R. Cruz-Cunha M. M., Tavares A. J., editor, *Handbook of Research on Developments in E-Health and Telemedicine: Technological and Social Perspectives*, pages 1024–1047. Facultad de Ingeniería, Universidad de Talca, Talca, Chile, 2010.

[17] P. Y. Wong and J. Gibbons. A process semantics for bpmn. In *Proceedings of the 10th International Conference on Formal Methods and Software Engineering*, ICFEM '08, pages 355–374, Berlin, Heidelberg, 2008. Springer-Verlag.