AN SPA PERFORMANCE MODEL OF A PRODUCTION CELL

D.R.W. HOLTON

Department of Computing, The University of Bradford, Bradford, BD7 1DP, UK. email: `D.R.W.Holton@comp.brad.ac.uk`

J.P.N. GLOVER

Department of Cybernetics, The University of Reading, Reading, UK. email: `cybjpng@cyber.rdg.ac.uk`

ABSTRACT

Stochastic process algebras (SPA) are a relatively new technique for constructing analytical models of the performance of systems; to date very little work has been done on applying them in automation. SPA are based on classical process algebras such as CCS, and inherit the virtues of being expressive, compositional, and formal. This paper demonstrates the use of an SPA by taking a specific production cell and constructing a performance model.

KEYWORDS: performance modeling, stochastic process algebras, production cell design

INTRODUCTION

The performance modeling community has long advocated modeling the performance of systems during the design phase (see [1]). Problems with the performance of a proposed system can be identified at an early stage in the design, allowing remedial action to be taken. This approach is desirable since the costs of changing a design increase rapidly the closer the system is to an implementation.

This aim of this work is to introduce PEPA [2], a general-purpose performance modeling technique, and to demonstrate its use in modeling an existing surface mount production cell [3]. Previous investigations [4] have generated an algebraic model in SCCS which allowed analysis of the transient behaviour of the cell due to faults as well as scheduled events such as maintenance. However transient analysis does not allow the feasibility of alternative cell configurations to be conveniently assessed. It is the steady-state behaviour which dictates viability, so a modeling technique allowing the solution of an appropriate cost function is required. Stochastic process algebras [5] such as PEPA are a recent modeling technique closely related to SCCS which allow the analysis of steady-state performance; to date the only work done on applying them in automation is [6, 7].

The remainder of this paper gives a description of the production cell modeled, followed by an introduction to PEPA. The model of the production cell follows, and the performance results obtained are presented. The paper concludes with an appraisal of the approach.

THE PRODUCTION CELL

The manufacturing application under consideration is a surface mount PCB production cell based around pick and place robots and a Zone Convection Re-flow (ZCR) oven. The specification of these components comes directly from a project funded jointly by industry and the EPSRC.[1] The production cell consists of a series of stand-alone machines which communicate in order to synchronise the passage of PCBs along the line. Firstly a loading machine presents unpopulated PCBs to a screen printer where the appropriate solder screen is pasted to the board. The PCB is inspected then passed to a component placement station consisting of one or more pick and place robots. After a further inspection stage the populated board is passed to the ZCR oven to reflux the solder paste. This reflux process is achieved by passing the PCB through a number of temperature controlled zones. The speed at which PCBs are conveyed through the zones defines the reflux thermal profile.

DESCRIPTION OF PEPA

A PEPA model specifies a stochastic process, which can be shown to be a continuous time Markov process. It is based on CCS [8], with annotations allowing the calculation of performance characteristics. The major design objectives for PEPA include:

Parsimony: it is desirable to provide few elements in the language, making it easy to reason about models and to give the modeler flexibility. PEPA provides components and activities, corresponding to states and transitions in the underlying stochastic model.

Formal definition: an exact description of the interpretation of all expressions allows the notion of equivalences to be defined. As a consequence of this it is possible to interchange equivalent components without changing the characteristics of a system, permitting model simplification. This ameliorates the state space explosion problem, common to all performance modeling techniques.

Compositionality: this key feature of SPA in general encourages the construction of models from simple components. Individual components can be developed and investigated in isolation. These isolated components are then composed to define the system being modeled.

A PEPA model describes a system as a collection of components which engage, singly or in cooperation, in activities. Components are the active units in a system, and activities denote the behaviour of a component. PEPA provides a quantification of time based on real numbers plus an unspecified time, $\top$, denoting a passive activity.

[1] "Flexible Distributed Control of Manufacturing Systems using Local Operating Networks" GR/J 64047.

The rate of an activity is represented by a random variable drawn from a negative exponential distribution. A description of the syntax of PEPA is given below.

Prefix: The expression (α, r) denotes an activity consisting of an *action* α and a rate r. The prefix combinator is the basic operator which combines activities to describe the behaviour of a components—the expression $(\alpha, r).P$ first engages in the activity (α, r) then behaves as component P.

Choice: Alternative behaviours are denoted by the choice combinator. The component $(\alpha, r_1).P + (\beta, r_2).Q$ behaves as either P or Q, depending on whether (α, r_1) or (β, r_2) completes first.

Constant: The provision of constants allows names to be given to components. It also allows recursive definitions such as $P \stackrel{def}{=} (\alpha, r).P$, allowing components with (potentially) infinite behaviour to be specified.

Cooperation: Cooperation allows components to be composed in parallel. $P \underset{S}{\bowtie} Q$ denotes that P and Q are composed in parallel and only synchronise when both perform actions from the cooperation set S.

A tool [9] exists to support the calculation of performance metrics from a PEPA model. It generates the Markov process from the PEPA model and solves it either symbolically, if the model is small, or numerically. Performance measures are calculated by associating *rewards* with activities in the model, for example if the throughput of a production cell is required then the activities which indicate that an item has been produced would have a reward associated with them. Similarly if the utilisation of a given machine in the cell is required then activities associated with the use of the machine can have an associated reward. It is possible to calculate numerous performance indicators simultaneously, allowing a detailed examination of the performance characteristics of a system to be performed efficiently.

PERFORMANCE MODELING THE PRODUCTION CELL

This section describes the model of the cell, followed by an indication of how the activity rates were calculated. Finally the performance metrics calculated from the model are presented.

The Model

The production cell is defined at the top level by a composition of components describing how the individual machines in the cell interact.

$$
\begin{aligned}
\mathit{Cell} \stackrel{def}{=} \; & \mathit{Loader} \underset{\{\mathit{Load}\}}{\bowtie} \mathit{SPrinter} \underset{\{\mathit{Deliver}\}}{\bowtie} \mathit{PrintInsp} \\
& \underset{\{\mathit{ToPP}\}}{\bowtie} \mathit{PickPlace} \underset{\{\mathit{OutputBoard}\}}{\bowtie} \mathit{InspConv} \underset{\{\mathit{ToZCR}\}}{\bowtie} \mathit{ZCR}
\end{aligned}
$$

This states that the *Loader* cooperates with the *SPrinter* via the action *Load*, the *SPrinter* cooperates with the *PrintInsp* via the *Deliver* action, and so on. The definition of each component is given below, along with a short commentary on each.

$$\mathit{Loader} \stackrel{def}{=} (\mathit{Load}, LT).\mathit{Loader}$$

The definition of the *Loader* component reflects the simplicity of this device. It performs the *Load* action with a rate of LT, meaning that a new board is loaded approximately every $1/LT$ seconds (the mean of a negative exponential distribution with parameter r is $1/r$). The loader then repeats this behaviour indefinitely.

$$\mathit{SPrinter} \stackrel{\mathit{def}}{=} (\mathit{Load}, \top).(\mathit{Deliver}, PT).\mathit{SPrinter}$$

The screen printer initially waits passively for a board to be loaded, indicated by the rate of the *Load* action being the unspecified time, $\top$. As the *Load* action is performed in cooperation with the *Loader*, it will not be completed until both components are ready to cooperate. This ensures that the loader and screen printer are synchronised when the transfer of a board takes place. The screen printer then delivers the board to the next machine in the cell. The rate PT incorporates both the time taken to print onto the board and the time taken to deliver it.

$$\mathit{PrintInsp} \stackrel{\mathit{def}}{=} (\mathit{Deliver}, \top).\Big((\mathit{ToPP}, PAR).\mathit{PrintInsp} + (\mathit{Reject}, PRR).\mathit{PrintInsp}\Big)$$

The print inspector receives a board and either rejects it if faulty or passes it to the next machine. As above, the internal activity of checking the board is omitted. The value of $PAR/(PAR + PRR)$ gives the probability that a board will be accepted, and the probability that a board is rejected is found in a similar way.

$$\mathit{PickPlace} \stackrel{\mathit{def}}{=} (\mathit{ToPP}, \top).(\mathit{OutputBoard}, IT).\mathit{PickPlace}$$

The pick and place robot receives boards, places components and delivers them. It would be possible to give a more detailed model, incorporating the number of placement heads and the effect of speed on population failure rate. However this level of detail is considered inappropriate and may result in an overly-complex Markov process. It is in cases such as this where the compositional nature of SPA is particularly useful. A detailed model of a robot could be built and analysed in isolation to find the rate at which it operated. A simplified model could then be constructed, proved to be equivalent and substituted in the model of the production cell.

$$\begin{aligned} \mathit{InspConv} \stackrel{\mathit{def}}{=}\; & (\mathit{OutputBoard}, \top). \\ & \Big((\mathit{ToZCR}, IAR).\mathit{InspConv} + (\mathit{Discard}, IRR).\mathit{InspConv}\Big) \end{aligned}$$

The system which inspects the boards after population is very similar to the print inspector.

$$\begin{aligned} \mathit{Zone}_1 &\stackrel{\mathit{def}}{=} (\mathit{ToZCR}, \top).(\mathit{Cook}_1, CT).\mathit{Zone}_1 \\ \mathit{Zone}_i &\stackrel{\mathit{def}}{=} (\mathit{Cook}_{i-1}, \top).(\mathit{Cook}_i, CT).\mathit{Zone}_i \qquad 1 < i < n \\ \mathit{Zone}_n &\stackrel{\mathit{def}}{=} (\mathit{Cook}_{n-1}, \top).(\mathit{Dispose}, CT).\mathit{Zone}_n \\ \mathit{ZCR} &\stackrel{\mathit{def}}{=} \mathit{Zone}_1 \underset{\{\mathit{Cook}_1\}}{\bowtie} \mathit{Zone}_i^{n-2} \underset{\{\mathit{Cook}_{n-1}\}}{\bowtie} \mathit{Zone}_n \end{aligned}$$

A more detailed model of the oven is given to illustrate the descriptive power of PEPA and the ease with which models can be composed. The physical nature of the oven lends itself to an elegant formulation of the model in terms of an indexed family of components. This allows a number of components which have substantially the same behaviour to be defined concisely. In this case a number is associated with each zone which is used as the index into the family of components. Each member of the family receives a PCB and passes it to the next member at an appropriate rate. At either end of the oven slightly different behaviour is required. At the front the PCB is received from the inspection system, and at the end the PCB is removed from the cell. The definitions of the components reflect this behaviour. The model of the oven is given by composing the individual zones, where the notation $Zone_i^{n-2}$ is used as shorthand to save writing out the composition of the components indexed by i (standing for $n-2$ copies of the composition of component $Zone_i$, with $1 < i < n$).

Calculating the Rates

Many of the rates required were calculated in a straightforward manner from knowledge of the machines in the cell. For example it was known that the loader took eight seconds to present a board, so the rate LT is $1/8$. The rates required for the inspection stations required more careful analysis. For example it is known that approximately that one in thirty boards are rejected by the print inspection station. A suitable value for PRR is therefore $\frac{1}{30} \times TRT$, where TRT is the rate at which boards move through the inspection station. Similarly $PAR = \frac{29}{30} \times TRT$.

Performance Results

From experience, cell designers consider the pick and place robot to be the heart of a production cell, determining throughput and production costs. The performance model can be used to confirm this qualitative analysis with quantitative results—see figure 1.

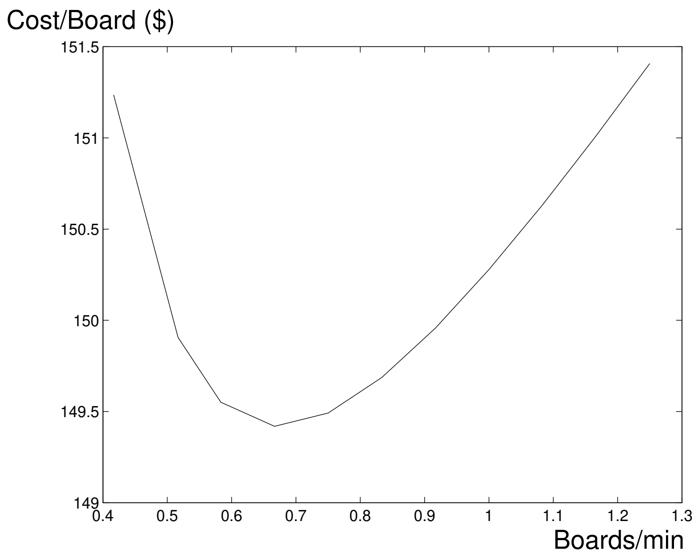

Figure 1: Performance Results

This shows the cost of producing a board against the speed at which the robot operates. A simplistic cost function was used, based on a fixed cost per unit time to run the cell, a production run of a particular size, and the assumption that the number of boards which fail the inspection is directly related to the speed at which the robot operates. A more realistic cost function with additional complexity could be used without requiring any modification to the PEPA model. Similarly, model parameters other than the robot speed could be varied to investigate their effect on the cost function.

CONCLUSIONS

The main aim of this work has been achieved—it has been shown that is is possible to create a PEPA specification of a production cell and to investigate its performance. However the expressive power of the language initially lead to an overly detailed formulation of the model, which included actions internal to a component which did not contribute to its performance. Appropriate simplification was required to remove these features in order to yield an equivalent model which could be solved for the desired performance measures.

The method yielded quantitative results which confirmed qualitative statements about the cell's performance and profitability, giving confidence in the cell specification. The real power of the method becomes evident when the complexity of the cell exceeds the capacity for a designer to reason qualitatively. However the onus is still on the designer to perform the appropriate analysis of the model.

REFERENCES

[1] C. Harvey. Performance engineering as an integral part of system design. *BT Technology Journal*, 4(3):143–147, July 1986.

[2] J. Hillston. *A Compositional Approach to Performance Modeling.* PhD thesis, University of Edinburgh, 1994. Available as technical report CST-107-94.

[3] K. Warwick. Flexible distributed control of manufacturing systems using local-operating networks. Technical report, Department of Cybernetics, University of Reading, 1995. Second 9-monthly report for the EPSRC DIP Project, Grant Reference GR/J 64047.

[4] J.P.N. Glover, G.T. Foster, and K. Warwick. Parallel Discrete Event Simulation of Production Lines using Local Operating Networks. In *Proceedings of IFAC/IFORS/IMACS Symposium on Large Scale Systems*, 1995.

[5] S. Gilmore and J. Hillston, editors. *Process Algebra and Performance Modeling*, volume 38(7) of *The Computer Journal*, December 1995.

[6] S. Gilmore, J. Hillston, D.R.W. Holton, and M. Rettelbach. Specifications in a Stochastic Process Algebra for a Robot Control Problem. *International Journal of Production Research*, December 1995.

[7] D.R.W. Holton. A PEPA Specification of an Industrial Production Cell. *The Computer Journal*, 38(7):542–551, 1995. ISSN: 0010-4620.

[8] R. Milner. *Communication and Concurrency.* Prentice Hall International, 1989.

[9] S. Gilmore and J. Hillston. The PEPA Workbench: A Tool to Support a Process Algebra-based Approach to Performance Modelling. In G. Haring and G. Kotsis, editors, *Proceedings of the Seventh International Conference on Modelling Techniques and Tools for Computer Performance Evaluation*, volume 794 of *LNCS*, pages 353–368. Springer-Verlag, 1994.