A PEPA Specification of an Industrial Production Cell

D.R.W. Holton

Department of Computing Science, The University of Aberdeen
Aberdeen, AB9 2UE, UK

This paper describes the use of the stochastic process algebra PEPA to model and evaluate the performance of an industrial production cell. An initial specification of the production cell is is developed and transformed to allow performance measures of the cell to be found. The reasons for making these transformations are given, and performance measures for the transformed specification are presented.

1. INTRODUCTION

Designing a production cell is a complicated task since there are many factors to be considered. The requirements of most manufacturing processes dictate that particular machines are required, for instance a machining task will require the use of lathe, but issues such as how materials are moved between machines and the sensing requirements of the cell are often not as well defined. Formal specification can be a valuable aid to a cell designer as it allows a range of options for the configuration of the cell to be explored in a precise setting—in particular sensing requirements can be clarified during the development of a specification as it often becomes evident what information about the state of the cell is required to ensure that it operates effectively. If a library of specifications of common machines is available to the designer then experimentation with the configuration of the cell would be greatly eased, allowing for a wider range of options to be explored. If the specification of the cell can be augmented with information related to the performance of each machine, allowing the designer to calculate performance measure such as the throughput of the cell, then the specification becomes more valuable to the designer because of the extra information that it provides. For example it would be possible for the designer to demonstrate that any constraints (such as minimum throughput) may be satisfied, or that those constraints were unrealistic. It may also be possible (though difficult!) to find the rates required to optimise the throughput of the cell given a set of constraints on the cost of setting up and operating the cell.

The main aim of this work is to investigate whether the stochastic process algebra PEPA [1] may be used to specify and investigate the performance of an industrial production cell. To date the published case studies have concentrated on networks and distributed systems (an exception is [2]), so it seemed worthwhile to attempt a case study in a different field, though it may be argued that production cells are an example of distributed systems.

The cell specified was used as the basis for one of the major case studies of the KorSo project [3], and is described in detail in [4]. There is a graphical simulation of the cell (see figure 1 for a screen-shot) which can be used to visualise how the cell operates. Since the simulator can be interfaced to any program which reads from standard input and writes to standard output, it is possible to implement specifications of the cell controller.

Further aims of this work include:

- determining how easily a specification of the cell may be developed, and seeing if extensions to PEPA are suggested by the case study; and
- investigating whether the performance of the cell predicted by the specification is reasonable, and also to investigate the feasibility of optimising the throughput of the cell by varying the speed at which components operate.

The next section of this paper describes the production cell. Following this the specification of the cell is given, then the transformations required to allow performance measures to be calculated are described. Performance characteristics of the transformed specification are then presented. The final section includes some conclusions and directions for future work.

2. THE PRODUCTION CELL

The task of the production cell is to forge metal plates (or *blanks*) in a press, and is based upon a cell in use in a metal processing plant near Karlsruhe. It consists of the following components:

feed belt: this is used to introduce the blanks to the cell from some central repository;

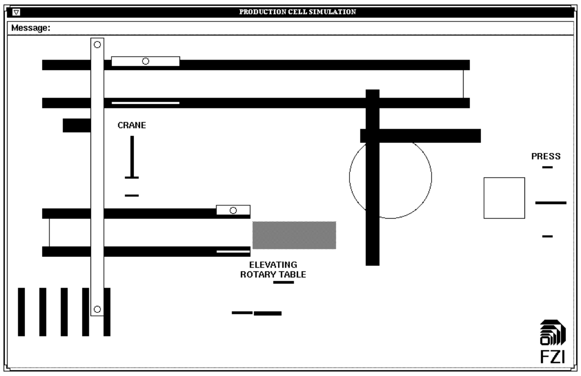


FIGURE 1. Plan view of the production cell

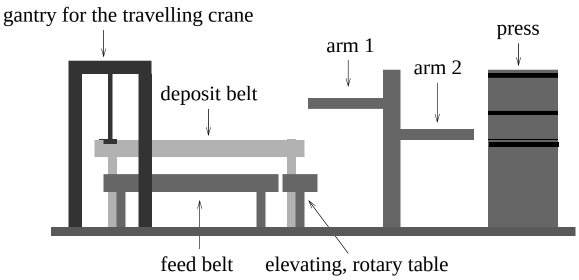


FIGURE 2. Side view of the production cell

elevating rotary table: blanks reach the end of the workcell and are passed onto this component, from where they may be picked up and fed into the next part of the process;

robot: this has two arms mounted on a common base which can rotate. The arms, which are mounted at different heights, can translate horizontally and have electro-magnetic end-effectors;

press: this is the 'active' component of the production cell which performs the processing of the blanks;

deposit belt: after blanks have been processed by the press, they are placed here to be removed from the production cell;

travelling crane: once the blanks reach the end of the deposit belt they are picked up by the travelling crane and moved to another part of the factory.

Figures 1 and 2 give plan and side views of the production cell.

Each component in the workcell has motors or *actuators* which are used to control the device, and *sensors* which are used to report its state. These will be described in more detail when the specification of each component is developed.

From the point of view of a blank, the sequence of events which occurs is:

1. it is placed on the feed belt and transported to the elevating rotary table;
2. the table moves into position so the blank can be picked up by the robot;
3. the robot moves the blank to the press, where it is forged;
4. the press moves to the unloading position and the blank is picked up by the robot;
5. the robot transports the blank to the deposit belt;
6. when the blank reaches the end of the belt it is picked up by the travelling crane and moved to its ultimate destination.

In the system being modelled, the ultimate destination of a blank is the feed belt, so this cycle repeats indefinitely. In a more realistic setting the blank would be moved to the next production phase.

3. THE SPECIFICATION

The main choice that had to be made when specifying the production cell was how to coordinate the components. The options were to specify a cell controller agent which embodied a plan for processing blanks (c.f. [2]) or to specify how adjacent components were coordinated. The latter was chosen because it was felt to be more in the spirit of the compositional nature of PEPA—only local interactions between components exist, which makes the effect of changing the specification of an agent obvious.

The following conventions are used in this specification:

- μ this indicates the rate at which internal commands such as $belt1\_start$ can be completed;
- ξ this indicates the rate at which synchronisation between machines in the cell can occur;
- $\top$ as usual this indicates that the agent is passive with respect to a synchronisation, i.e. that it will perform that activity as soon as the agent or agents it cooperates with are ready.

Another convention used is that when a handshake between machines necessitates that they cooperate more than once, the same name is used for all the activities. This is to suggest that there would be a single physical connection between the machines.

In the following sections each component of the cell is specified, and these agents are composed to specify the behaviour of the complete production cell.

3.1. Feed Belt

The feed belt has a single actuator which may be turned on and off to start and stop the belt, and a single photoelectric sensor used to determine if a blank is at the end of the belt. When the blank breaks the light beam the sensor turns on and stays on until the blank has moved past the sensor.

The specification of the belt agent is given below:

$$Belt_0 \stackrel{\text{def}}{=} (belt1\_start, \mu).Belt_1$$
$$Belt_i \stackrel{\text{def}}{=} (blank\_add, \mu).(sensor_{On}, F_T). \quad 0 < i < N$$

$$
\begin{aligned}
& ((\mathit{ready\_to\_put}, \top).(\mathit{sensor}_{\mathit{Off}}, F_B). \\
& (\mathit{ready\_to\_put}, \xi).\mathit{Belt}_{i+1} \\
+\ & (\mathit{belt1\_stop}, \mu).(\mathit{ready\_to\_put}, \top). \\
& (\mathit{belt1\_start}, \mu).(\mathit{sensor}_{\mathit{Off}}, F_B). \\
& (\mathit{ready\_to\_put}, \xi).\mathit{Belt}_{i+1}) \\
\mathit{Belt}_N \stackrel{\mathrm{def}}{=}\ & (\mathit{can\_accept}, \xi).(\mathit{sensor}_{\mathit{On}}, F_T). \\
& ((\mathit{ready\_to\_put}, \top).(\mathit{sensor}_{\mathit{Off}}, F_B). \\
& (\mathit{ready\_to\_put}, \xi).\mathit{Belt}_N \\
+\ & (\mathit{belt1\_stop}, \mu).(\mathit{ready\_to\_put}, \top). \\
& (\mathit{belt1\_start}, \mu).(\mathit{sensor}_{\mathit{Off}}, F_B). \\
& (\mathit{ready\_to\_put}, \xi).\mathit{Belt}_N)
\end{aligned}
$$

This agent is slightly complicated by the fact that there is a finite supply of blanks in the simulated production cell. This means that the number of blanks in circulation must be counted and, once they have all been introduced, blanks which have already been processed are allowed to go through the cell again.

The actions via which this agent cooperates with adjacent agents are *ready_to_put* and *can_accept*; the rest denote internal actions which cause the belt to change state (start or stop) or indicate information being fed back from the machine (e.g. sensor_{on} indicates that the sensor at the end of the belt has switched on). This is typical of all the agents in this specification—agents are mostly concerned with controlling machines and the interactions between agents are minimised.

3.2. Elevating Rotary Table

The purpose of the elevating rotary table is to receive a blank from the feed belt and perform the appropriate translation and rotation so the robot arm can pick up the blank and place it correctly in the press.

This component has two actuators, one of which raises and lowers the table and the other which rotates the table. It has three sensors for determining whether the table is at the lower or upper position and to report the angle through which the table has been rotated.

The behaviour of the table is rather dull—initially it is positioned so it can accept a blank from the feed belt; once a blank has been received the table moves so the robot can pick the blank up; once the blank has been removed by the robot it returns to the starting position, ready for another cycle.

$$
\begin{aligned}
\mathit{Table} \stackrel{\mathrm{def}}{=}\ & \mathit{PutPosn}.(\mathit{ready\_to\_put}, \xi). \\
& (\mathit{ready\_to\_put}, \top).\mathit{PickPosn}. \\
& (\mathit{ready\_to\_pick}, \xi).(\mathit{ready\_to\_pick}, \top). \\
& \mathit{Table} \\
\mathit{PutPosn} \stackrel{\mathrm{def}}{=}\ & (\mathit{table\_downward}, \mu).(\mathit{table\_left}, \mu). \\
& ((\mathit{bottom}, T_T).(\mathit{table\_stop\_v}, \mu). \\
& (\mathit{table\_rot}_0, T_R).(\mathit{table\_stop\_h}, \mu) \\
+\ & (\mathit{table\_rot}_0, T_R)(\mathit{table\_stop\_h}, \mu). \\
& (\mathit{bottom}, T_T).(\mathit{table\_stop\_v}, \mu)) \\
\mathit{PickPosn} \stackrel{\mathrm{def}}{=}\ & (\mathit{table\_upward}, \mu).(\mathit{table\_right}, \mu). \\
& ((\mathit{top}, T_T).(\mathit{table\_stop\_v}, \mu). \\
& (\mathit{table\_rot}_{50}, T_R).(\mathit{table\_stop\_h}, \mu) \\
+\ & (\mathit{table\_rot}_{50}, T_T).(\mathit{table\_stop\_v}, \mu). \\
& (\mathit{top}, T_T).(\mathit{table\_stop\_v}, \mu))
\end{aligned}
$$

3.3. Robot

The robot is used to transport blanks to and from the press. It consists of two arms attached to a common base, each with an electro-magnetic end effector. It has actuators for rotating the robot, for extending and retracting the arms, and for activating and deactivating the end effectors. It has three sensors which report the angle of rotation of the robot and the extension of each arm.

The robot has the most complicated behaviour of any of the devices. This is reasonable as it has to interact with three other devices. The robot first picks up a blank from the table, loads it into the press and returns to collect the next blank from the table. When it has received a new blank, the second arm is used to remove the processed blank from the press and this is moved to the deposit belt. The first arm then loads the press with a new blank and the cycle repeats.

$$
\begin{aligned}
\mathit{Robot}_0 \stackrel{\mathrm{def}}{=}\ & \mathit{InitRobot}.\mathit{Robot} \\
\mathit{InitRobot} \stackrel{\mathrm{def}}{=}\ & (\mathit{arm}_1\mathit{\_backward}, \mu). \\
& (\mathit{safe\_posn}_{\mathit{arm}_1}, A1_{T2}). \\
& (\mathit{arm}_1\mathit{\_stop}, \mu). \\
& (\mathit{arm}_2\mathit{\_backward}, \mu). \\
& (\mathit{safe\_posn}_{\mathit{arm}_2}, A2_{T3}). \\
& (\mathit{arm}_2\mathit{\_stop}, \mu).\mathit{Rot}_1\mathit{Table}. \\
& \mathit{FromTable}.\mathit{Rot}_1\mathit{Press}. \\
& \mathit{LoadPress}.\mathit{Rot}_1\mathit{Table} \\
\mathit{Robot} \stackrel{\mathrm{def}}{=}\ & \mathit{FromTable}.\mathit{Rot}_2\mathit{Press}. \\
& \mathit{UnloadPress}.\mathit{Rot}_2\mathit{Belt}. \\
& \mathit{ToDBelt}.\mathit{Rot}_1\mathit{Press}. \\
& \mathit{LoadPress}.\mathit{Rot}_1\mathit{Table}.\mathit{Robot} \\
\mathit{Rot}_1\mathit{Table} \stackrel{\mathrm{def}}{=}\ & (\mathit{robot\_right}, \mu).(\mathit{at\_table}_{\mathit{arm}_1}, R_1). \\
& (\mathit{robot\_stop}, \mu) \\
\mathit{Rot}_1\mathit{Press} \stackrel{\mathrm{def}}{=}\ & (\mathit{robot\_left}, \mu).(\mathit{at\_press}_{\mathit{arm}_1}, R_4). \\
& (\mathit{robot\_stop}, \mu) \\
\mathit{Rot}_2\mathit{Belt} \stackrel{\mathrm{def}}{=}\ & (\mathit{robot\_left}, \mu).(\mathit{unload\_blank}, \xi + \delta). \\
& (\mathit{at\_belt}_{\mathit{arm}_2}, R_3).(\mathit{robot\_stop}, \mu) \\
\mathit{Rot}_2\mathit{Press} \stackrel{\mathrm{def}}{=}\ & (\mathit{robot\_left}, \mu).(\mathit{at\_press}_{\mathit{arm}_2}, R_2). \\
& (\mathit{robot\_stop}, \mu) \\
\mathit{FromTable} \stackrel{\mathrm{def}}{=}\ & (\mathit{ready\_to\_pick}, \top).(\mathit{arm}_1\mathit{\_forward}, \mu). \\
& (\mathit{pick\_posn}_{\mathit{arm}_1}, A1_{T1}).(\mathit{arm}_1\mathit{\_stop}, \mu). \\
& (\mathit{arm}_1\mathit{\_mag\_on}, \mu).(\mathit{ready\_to\_pick}, \xi).
\end{aligned}
$$

$$
\begin{aligned}
& (arm_1\_backward, \mu). \\
& (safe\_posn_{arm_1}, A1_{T2}).(arm_1\_stop, \mu) \\
UnloadPress \stackrel{\text{def}}{=} \; & (unload\_blank, \top).(arm_2\_forward, \mu). \\
& (pick\_posn_{arm_2}, A2_{T1}).(arm_2\_stop, \mu). \\
& (arm_2\_mag\_on, \mu). \\
ToDBelt \stackrel{\text{def}}{=} \; & (DBelt\_ready, \top).(arm_2\_forward, \mu). \\
& (put\_posn_{arm_2}, A2_{T2}).(arm_2\_stop, \mu). \\
& (arm_2\_mag\_off, \mu).(DBelt\_ready, \xi). \\
& (arm_2\_backward, \mu). \\
& (safe\_posn_{arm_2}, A2_{T3}).(arm_2\_stop, \mu) \\
LoadPress \stackrel{\text{def}}{=} \; & (load\_blank, \top).(arm_1\_forward, \mu). \\
& (put\_posn_{arm_1}, A1_{T2}).(arm_1\_stop, \mu). \\
& (arm_1\_mag\_off, \mu).(arm_1\_backward, \mu). \\
& (safe\_posn_{arm_1}, A1_{T2}).(arm_1\_stop, \mu). \\
& (load\_blank, \xi)
\end{aligned}
$$

3.4. Press

It is the job of the press to process the blanks, where processing amounts to forging the blank by compressing it. To achieve this the press has two horizontal plates, the lower of which can be translated in the vertical direction. An actuator is used to perform this translation. The lower plate can come to rest in one of three positions; the lower position which allows a blank to be removed from the press, the middle position which allows a blank to be loaded into the press, and the upper position where forging occurs. The press has three sensors, reporting which of these three positions the plate is in.

The behaviour of the press is very simple: it moves to the loading position and waits until a blank has been loaded, forges it, moves the blank to the position from where it can be removed and repeats this cycle.

$$
\begin{aligned}
Press_0 \stackrel{\text{def}}{=} \; & (press_{top}, \mu).Top2Middle.Press \\
+ \; & (press_{middle}, \mu).Press \\
+ \; & (press_{bottom}, \mu).Bottom2Middle. \\
& Press \\
Press \stackrel{\text{def}}{=} \; & (load\_blank, \xi).(load\_blank, \top). \\
& Middle2Top.Top2Bottom. \\
& (unload\_blank, \xi).(unload\_blank, \top). \\
& Bottom2Middle.Press \\
Top2Middle \stackrel{\text{def}}{=} \; & (press\_downward, \mu). \\
& (press_{middle}, P_{m2t}).(press\_stop, \mu) \\
Middle2Top \stackrel{\text{def}}{=} \; & (press\_upward, \mu). \\
& (press_{top}, P_{m2t}).(press\_stop, \mu) \\
Top2Bottom \stackrel{\text{def}}{=} \; & (press\_downward, \mu). \\
& (press_{bottom}, P_{t2b}).(press\_stop, \mu) \\
Bottom2Middle \stackrel{\text{def}}{=} \; & (press\_upward, \mu). \\
& (press_{middle}, P_{b2m}).(press\_stop, \mu)
\end{aligned}
$$

3.5. Deposit Belt

The deposit belt is very similar to the feed belt—it also has one effector for starting and stopping the belt and one sensor to indicate when a blank reaches the end of the belt.

It was decided to allow the deposit belt to contain more than one blank, which meant that this agent is not identical to the feed belt. The most obvious way to achieve this is to treat the belt as a bounded buffer, however while this would be perfectly adequate it was decided to investigate a more complicated way of specifying this device. The basic idea is to split the belt into three regions, each of which is controlled by its own agent. Blanks are received by the first region and passed on to the final via the second.

$$
\begin{aligned}
DBelt_0 & \stackrel{\text{def}}{=} (belt2\_start, \mu).DBelt \\
DBelt & \stackrel{\text{def}}{=} Front_1 \underset{L_1}{\bowtie} Middle_0 \underset{L_2}{\bowtie} End \\
Front_1 & \stackrel{\text{def}}{=} (DBelt\_ready, \xi).Front_2 \\
& \quad + WaitForBelt.Front_1 \\
Front_2 & \stackrel{\text{def}}{=} (DBelt\_ready, \top).Front_3 \\
& \quad + WaitForBelt.Front_2 \\
Front_3 & \stackrel{\text{def}}{=} (front\_to\_middle, D_B + \delta).Front_1 \\
& \quad + WaitForBelt.Front_3 \\
Middle_0 & \stackrel{\text{def}}{=} (front\_to\_middle, \top).Middle_1 \\
Middle_j & \stackrel{\text{def}}{=} (front\_to\_middle, \top).Middle_{j+1} \\
& \quad + (middle\_to\_end, \xi).Middle_{j-1} \\
& \quad + WaitForBelt.Middle_j \\
Middle_M & \stackrel{\text{def}}{=} (middle\_to\_end, \top).Middle_{M-1} \\
& \quad + WaitForBelt.Middle_M \\
End & \stackrel{\text{def}}{=} (middle\_to\_end, \top).End \\
& \quad + Unload.End \\
Unload & \stackrel{\text{def}}{=} (sensor_{On}, \top).(sensor_{Off}, D_B). \\
& \quad (belt2\_stop, \mu).(wait, \xi) \\
& \quad (blank\_ready, \xi).(blank\_ready, \top). \\
& \quad (belt2\_start, \mu).(go, \xi) \\
WaitForBelt & \stackrel{\text{def}}{=} (wait, \top).(go, \top) \\
L_1 & = \{front\_to\_middle, wait, go\} \\
L_2 & = \{middle\_to\_end, wait, go\}
\end{aligned}
$$

What has been gained by this extra complexity? The first thing to note is that this specification demonstrates a technique for synchronising several agents on a particular activity. This is a common requirement in physical systems and represents an 'emergency stop' button—when the red button is pressed all components of the device must stop. However this feature alone would not justify the extra complexity. The idea for splitting a conveyor belt into different regions came from considering how to model a workpiece moving through several phases in a manufacturing process, for example through a zone convection re-flow oven. In such a device a workpiece spends given amounts of time in the various zones before passing to the next, and it is therefore important to model the transitions of pieces between zones.

3.6. Travelling Crane

The purpose of the travelling crane is to move the blank from the deposit belt to its final destination—in this case it is back to the feed belt. It has three actuators; one to move the crane laterally between the belts, one to change the height of the end effector, and one to activate and deactivate the electromagnet attached to the end effector. It also has three sensors; one to report the height of the end effector, one which reports when the crane is above the feed belt and one which reports when it is above the deposit belt.

The behaviour of the crane is straightforward. It is initially above the deposit belt with the electromagnet switched off. Once a blank is ready the end effector is lowered and the electromagnet switched on, the end effector is then raised and the crane moves to the feed belt. When the feed belt indicates that it is ready to accept a blank the end effector is lowered, the electromagnet switched off, the end effector is raised again and the crane moves back to the deposit belt.

$$
\begin{aligned}
Crane_0 &\stackrel{\mathrm{def}}{=} MagUp.(crane\_mag\_off, \mu).\\
&\quad ToBelt2.Crane\\
Crane &\stackrel{\mathrm{def}}{=} (blank\_ready, \top).MagDown.\\
&\quad (crane\_mag\_on, \mu).(blank\_ready, \xi).\\
&\quad MagUp.ToBelt1.(can\_accept, \top).\\
&\quad MagDown.Crane_0\\
MagDown &\stackrel{\mathrm{def}}{=} (crane\_lower, \mu).(mag_d, C_V).\\
&\quad (crane\_stop\_v, \mu)\\
MagUp &\stackrel{\mathrm{def}}{=} (crane\_lift, \mu).(mag_u, C_V).\\
&\quad (crane\_stop\_v, \mu)\\
ToBelt1 &\stackrel{\mathrm{def}}{=} (crane\_to\_belt1, \mu).(over\_belt1, C_H).\\
&\quad (crane\_stop\_h, \mu)\\
ToBelt2 &\stackrel{\mathrm{def}}{=} (crane\_to\_belt2, \mu).(over\_belt2, C_H).\\
&\quad (crane\_stop\_h, \mu)
\end{aligned}
$$

3.7. The Production Cell

The individual agents described above need to be composed to define the behaviour of the complete production cell.

$$
\begin{aligned}
Workcell &\stackrel{\mathrm{def}}{=} Robot_0 \underset{S_1}{\bowtie} ((Belt_0 \underset{S_2}{\bowtie} Table \parallel DBelt_0)\\
&\quad \underset{S_3}{\bowtie} Crane_0 \parallel Press_0)\\
S_1 &= \{ready\_to\_pick, unload\_blank,\\
&\quad DBelt\_ready, load\_blank\}\\
S_2 &= \{ready\_to\_put\}\\
S_3 &= \{blank\_ready, can\_accept\}
\end{aligned}
$$

This rather curious composition reflects the structure of the workcell. The innermost expression composes the feed belt, the table and the deposit belt, though it is only the feed belt and the table which cooperate. This composite agent cooperates with the crane, as the crane needs to cooperate with both belts, and this is composed with the press. Finally the robot synchronises with this bracketed expression. A diagram showing which agents cooperate is given in figure 3.

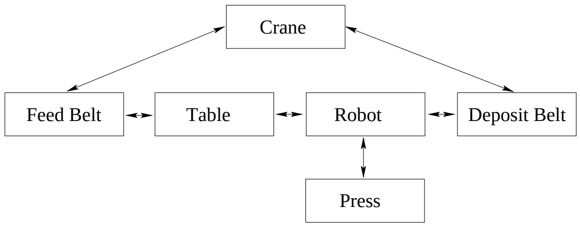


FIGURE 3. Schematic view of the production cell

4. TRANSFORMING THE SPECIFICATION

PEPA is supported by a tool [5] which allows a user to generate performance measures from a specification. The specification developed above was submitted to the tool in order to evaluate the performance of the cell, but it was found that the derivation graph of the specification was so large that it was effectively impossible to compute. This lead to a number of simplifications being made to the specification.

There were two distinct types of problems with the specification:

- the corresponding transition graph had an enormous number of nodes;
- when subdividing behaviours, some agents which were not closed were introduced;
- there were nodes in the transition graph which were visited very infrequently (usually once), or were associated with activities which had very high rates, and therefore had low probabilities associated with them which caused the infinitesimal generator matrix to be ill-conditioned.

The last problem was only revealed after the first and second were addressed. In the rest of this section the transformed agents are presented and the major changes explained.

4.1. Feed Belt

In common with all the agents in the original specification, a large proportion of the feed belt's actions are internal and have very high rates, and therefore do not contribute greatly to determining the *performance* of the cell. This agent originally had a number of transient states—$Belt_0$ was used to initialise the feed belt and $Belt_i$ counted the number of blanks introduced into the workcell. The initialisation was redundant once the internal actions were removed and counting the number of blanks was thought to be artificial, as a real workcell would effectively have an infinite supply of blanks. Similarly accepting blanks from the crane to be recirculated around the cell was unrealistic. The choice between behaviours in $Belt_N$ was also redundant once the internal actions had been removed. The simplified version of the belt agent is therefore:

$$\begin{aligned} Belt \stackrel{\text{def}}{=}\ & (B\_sensor_{On}, F_T).(ready\_to\_put, \top). \\ & (B\_sensor_{Off}, F_B).(ready\_to\_put, \xi).Belt \end{aligned}$$

4.2. Elevating Rotary Table

Similar simplifications were made to this agent, in particular the rotation and translation of the table were considered as a single movement with the rate calculated from the slower operation.

$$\begin{aligned} Table \stackrel{\text{def}}{=}\ & (move\_table, T_T).(ready\_to\_put, \xi). \\ & (ready\_to\_put, \top).(move\_table, T_T). \\ & (ready\_to\_pick, \xi).(ready\_to\_pick, \top).Table \end{aligned}$$

4.3. Robot

The major change to the robot agent was removing the initialisation behaviour which ensured the press was loaded and the arm moved back to the table. This did not alter greatly the definition of the agent, but had an important effect on the behaviour of the press.

$$\begin{aligned} Robot &\stackrel{\text{def}}{=} (ready\_to\_pick, \top).FromTable \\ FromTable &\stackrel{\text{def}}{=} (pick\_posn_{arm_1}, A1_{T1}). \\ & \quad (ready\_to\_pick, \xi). \\ & \quad (safe\_posn_{arm_1}, A1_{T2}).Rot_2Press \\ Rot_2Press &\stackrel{\text{def}}{=} (at\_press_{arm_2}, R_2).UnloadPress \\ UnloadPress &\stackrel{\text{def}}{=} (unload\_blank, \top). \\ & \quad (pick\_posn_{arm_2}, A2_{T1}).Rot_2Belt \\ Rot_2Belt &\stackrel{\text{def}}{=} (unload\_blank, \xi).(at\_belt_{arm_2}, R_3). \\ & \quad ToDBelt \\ ToDBelt &\stackrel{\text{def}}{=} (DBelt\_ready, \top). \\ & \quad (put\_posn_{arm_2}, A2_{T2}). \\ & \quad (DBelt\_ready, \xi). \\ & \quad (safe\_posn_{arm_2}, A2_{T3}).Rot_1Press \\ Rot_1Press &\stackrel{\text{def}}{=} (at\_press_{arm_1}, R_4).LoadPress \\ LoadPress &\stackrel{\text{def}}{=} (load\_blank, \top).(put\_posn_{arm_1}, A1_{T2}). \\ & \quad (safe\_posn_{arm_1}, A1_{T2}).(load\_blank, \xi). \\ & \quad Rot_1Table \\ Rot_1Table &\stackrel{\text{def}}{=} (at\_table_{arm_1}, R_1).Robot \end{aligned}$$

4.4. Press

Other than removing the initialisation and internal commands, the behaviour of the press had to reflect that the robot no longer initially loaded it with a blank. The simplest way to do this was for the press to allow itself to be unloaded before it accepted a new blank, meaning that the robot will initially unload an empty press, but this will not affect the steady state performance of the cell.

$$\begin{aligned} Press \stackrel{\text{def}}{=}\ & (unload\_blank, \xi).(unload\_blank, \top). \\ & (move\_press, P_{b2m}).(load\_blank, \xi). \\ & (load\_blank, \top).(move\_press, P_{m2t2b}).Press \end{aligned}$$

4.5. Deposit Belt

Introducing cooperation between agents causes the transition graph to grow enormously, therefore it was decided to completely re-write this agent as a bounded buffer.

$$\begin{aligned} DBelt_1 &\stackrel{\text{def}}{=} (DBelt\_ready, \xi).(DBelt\_ready, \top). \\ & \quad (delay, D_B).DBelt_2 \\ DBelt_2 &\stackrel{\text{def}}{=} (DBelt\_ready, \xi).(DBelt\_ready, \top). \\ & \quad (delay, D_B).DBelt_3 \\ & \ + (D\_sensor_{On}, D_T).(D\_sensor_{Off}, D_B). \\ & \quad (blank\_ready, \xi).(blank\_ready, \top).DBelt_1 \\ DBelt_3 &\stackrel{\text{def}}{=} (D\_sensor_{On}, D_T).(D\_sensor_{Off}, D_B). \\ & \quad (blank\_ready, \xi).(blank_r eady, \top).DBelt_2 \end{aligned}$$

4.6. Travelling Crane

The only extra change required for this agent was to modify the rate of the *can_accept* action from $\top$ to ξ, since this agent no longer cooperates with the feed belt.

$$\begin{aligned} Crane \stackrel{\text{def}}{=}\ & (mag_u, C_V).(over\_belt2, C_H). \\ & (blank\_ready, \top).(mag_d, C_V). \\ & (blank\_ready, \xi).(mag_u, C_V). \\ & (over\_belt1, C_H).(can\_accept, \xi). \\ & (mag_d, C_V).Crane \end{aligned}$$

4.7. The Production Cell

The composition of the agents was slightly simplified since the crane and the feed belt no longer cooperate. A diagram showing which agents cooperate is given in figure 4

$$\begin{aligned} Workcell &\stackrel{\text{def}}{=} Robot \underset{S_1}{\bowtie} ((Belt \underset{S_2}{\bowtie} Table) \parallel Press \\ & \quad \parallel (DBelt_1 \underset{S_3}{\bowtie} Crane)) \\ S_1 &= \{ready\_to\_pick, unload\_blank, \\ & \quad DBelt\_ready, load\_blank\} \\ S_2 &= \{ready\_to\_put\} \\ S_3 &= \{blank\_ready\} \end{aligned}$$

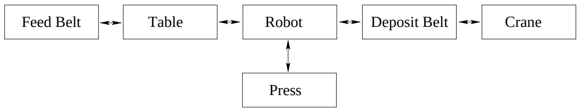


FIGURE 4. Schematic view of the production cell

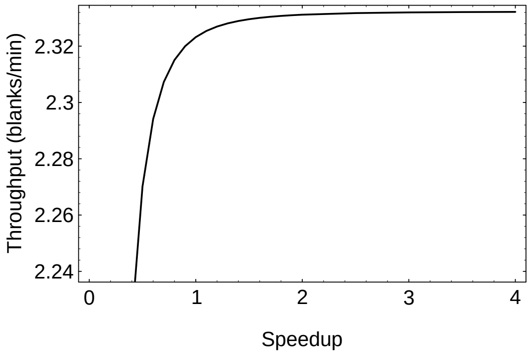
5. PERFORMANCE OF THE CELL

Performance figures were calculated from the transformed specification using the PEPA workbench to compile the specification to a transition system, which was then transformed into a Markov chain. The Markov chain was solved to allow steady state properties to be calculated—for this paper the *throughput* of the cell and the *utilisation* of the press were the quantities calculated. Throughput was defined to be the number of blanks the cell could produce in a minute. The press utilisation was defined to be the percentage of time it spent doing useful work. It is typical in production cell design to try to determine the utilisation of particular components of a cell as it is desirable to ensure that the components which do the majority of the processing of the raw materials are utilised as much as possible, particularly since these tend to be the most expensive machines in the cell.

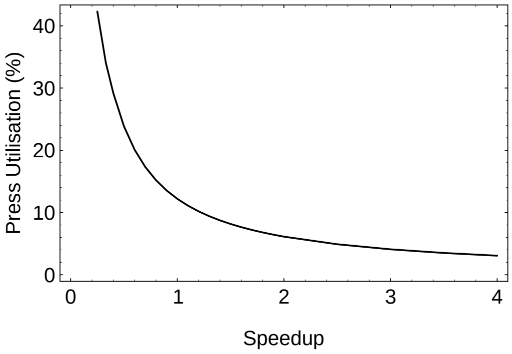
The performance figures are based on timings taken from the simulation of the workcell. The values assigned to the rate parameters are given in table 1. The experiments take the form of investigating the effect of speeding up a given component, or number of components, on the throughput of the cell and the press utilisation. A component is speeded up by multiplying each of its rate parameters by a speedup factor.

The first experiment was to determine the throughput of the cell and the press utilisation given the figures in table 1. The throughput was 2.32 blanks/minute, and the utilisation of the press was 12.2%. While the controller described by the specification given here has not yet been implemented, the throughput of the cell with the supplied controller was very similar.

The next set of experiments was to investigate how changing the speed of the press effects the performance of the cell, since the press is the central component. Figure 5(a) shows that the difference in throughput between slowing the press down by a factor of two and speeding it up by a factor of four is minimal, indicating that the press is not a bottle-neck in this production process. However the utilisation of the press decreases significantly as the speed of the press increases, which can be explained by the fact that it would take longer for a slower press to perform its task, therefore it press would spend less time waiting to be loaded or unloaded by the robot. This would indicate that it would be possible to use a relatively slow press in the production cell with no significant difference in the throughput.



(a)



(b)

FIGURE 5. Effect of speeding up the press

Since it is apparent that the press spends much of its time waiting to be loaded or unloaded, the next set of experiments investigated the effect of changing the speed of the robot. Figure 6 shows that there is a significant cost of slowing the robot down, but that speeding it up makes little difference to the throughput and the press utilisation. This indicates that the bottle-neck in the production process exists elsewhere.

The effect of speeding up the other components in the workcell was investigated, and it was found that there was little to gain from speeding up any of them apart from the crane, as can be seen in figure 7. Figure 7(a) shows that the increase in throughput was significantly greater than that experienced when speeding up the other components, and that this increase only tails off after the speed of the crane had been doubled. Figure 7(b) shows that this increase in throughput is not at the expense of press utilisation, so it would appear to be desirable to provide a faster crane for the workcell.

The experiments with the crane prompted the investigation of speeding up both the press and the crane. Figure 8(a) indicates that this proved not to increase the throughput of the cell significantly, while 8(b) shows that it has a detrimental effect on the utilisation of the

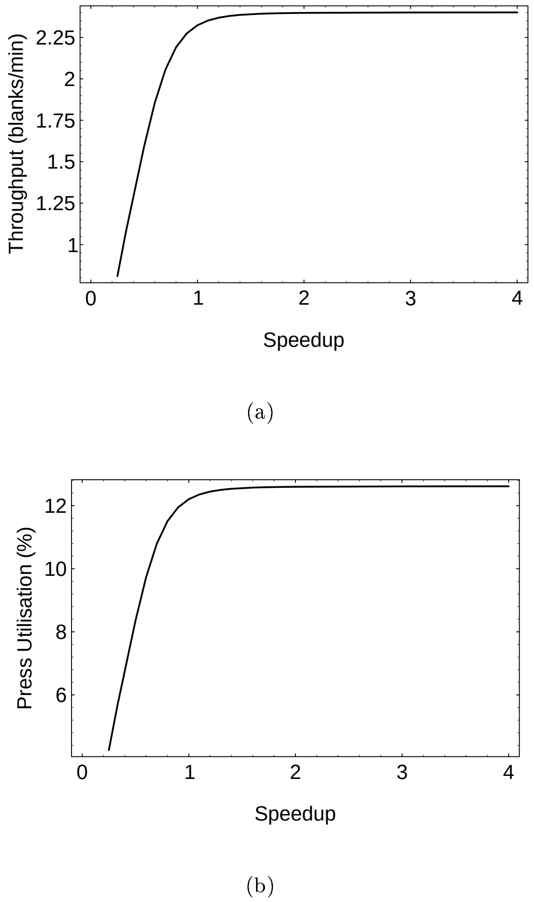


FIGURE 6. Effect of speeding up the robot

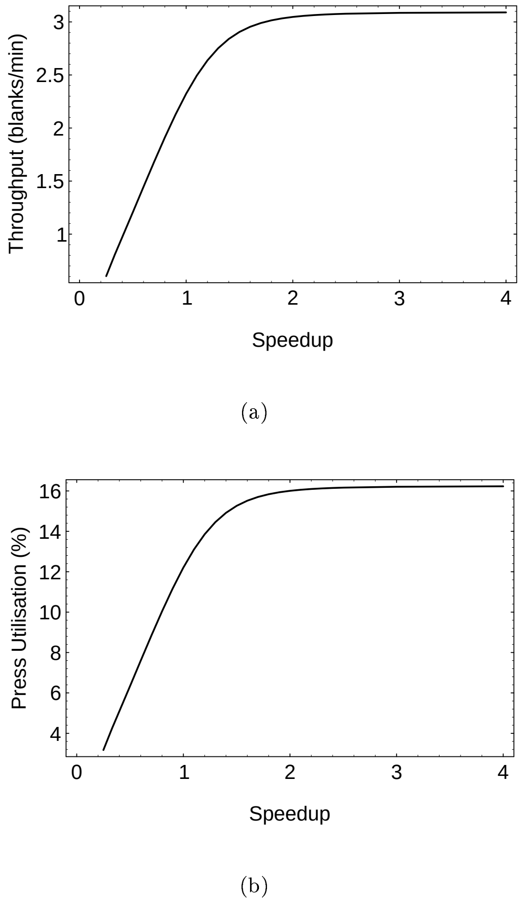


FIGURE 7. Effect of speeding up the crane

press. This can be interpreted as meaning that as the speed of the press increases and blanks are transported out of the cell more quickly by the crane, the press spends more and more time waiting to be loaded or unloaded, indicating that it is the robot causing a bottleneck.

This finding motivated the final set of experiments presented here, where both the robot and the press are speeded up. Figure 9 shows that speeding up these components has a beneficial effect on the throughput and press utilisation, and that the improvements achieved are by far the most significant discovered. It appears that the most significant bottle-neck was the rate at which blanks were removed from the cell, and once this was alleviated it was possible to increase the rate at which blanks were fed to the press in order to further improve the cell throughput and the press utilisation. The fact that both graphs flatten off as the speedup increases indicates that the speed at which other components operate starts to reduce these gains, but this only occurs at unrealistic values of the speedup of the crane and robot.

6. CONCLUSIONS AND FUTURE WORK

6.1. Conclusions

The main aim of this work has been achieved—it has been shown that it is possible to create a PEPA specification of a production cell and to investigate the performance of that cell. The initial specification was flawed in two respects: because of a lack of understanding of the form that PEPA agents may take, specifically that all agents must be closed; and secondly in trying to mix the specification of functional aspects of the cell with performance aspects. From the latter point it can be seen that the level of abstraction with which a system should be described is somewhat different to that which might be expected by a specifier familiar with process algebras. For example internal activities with a very high rate, such as commands to start or stop a conveyor belt, were removed from the initial specification since they contributed little to understanding the performance of the cell, but contributed to making the analysis of the specification intractable. This means that a specifier would have to take care to include only those aspects which are important for determining performance characteristics, but the subsequent specifica-

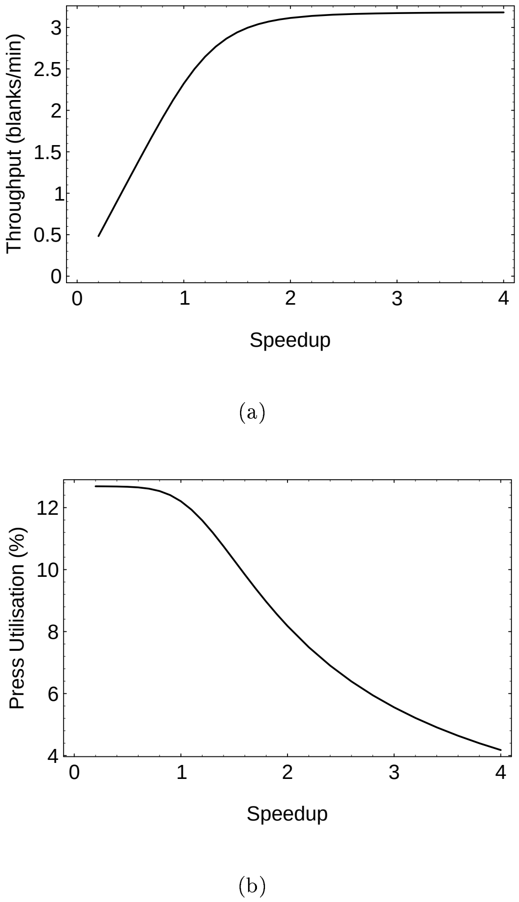


FIGURE 8. Effect of speeding up the press and crane

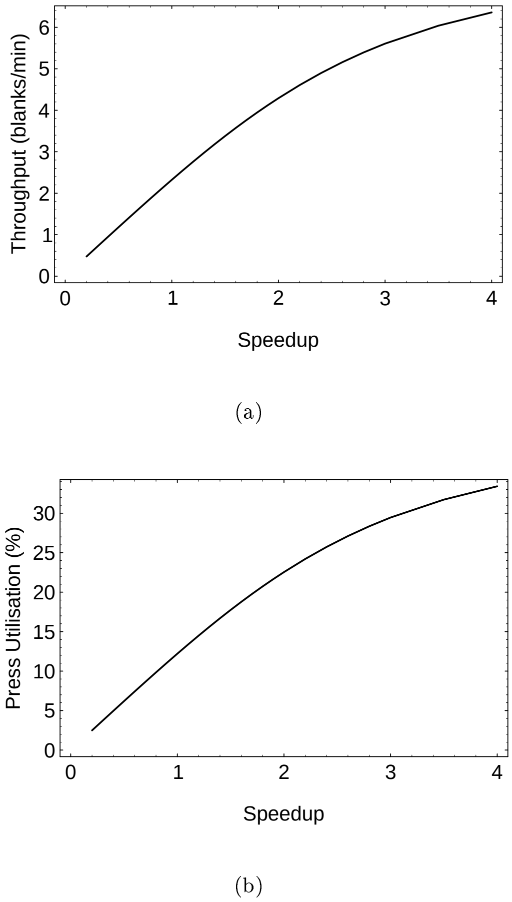


FIGURE 9. Effect of speeding up the robot and crane

tion may not be a suitable basis for an implementation. A different approach would be to designate such activities as instantaneous and for tools which calculate the transition system corresponding to the specification to omit these activities.

A further aim was to investigate whether this case study suggested extensions to PEPA to make modelling this type of system more convenient. The first observation is that sequential composition of agents would allow specifications to be written in a more natural style. For example the definition of the *Robot* agent in section 3.3. is rather more readable than that in 4.3. It should be possible to define sequential composition in such a way that those agents which incorporate it can be transformed to equivalent ones without sequential composition, so leaving the core semantics of the language unchanged.

Another language feature that would be useful is the *disabling* operator found in LOTOS [6]. This allows an agent to be disrupted by another agent (i.e. control passes to the second agent) if any of the initial events of the second agent could occur. This would increase the clarity of the specification of failures in, or interruptions to, an agent. An example of where this would be useful in the specification of this cell is in the deposit belt agent (section 3.5.). The *End*, *Front* and *Middle* agents could be disabled by an *Unload* agent, ensuring that no further blanks could be placed on the conveyor belt while it was halted. However it is not immediately obvious how this feature would be incorporated in PEPA.

Notwithstanding the above comments, it was possible to investigate the performance of the cell and discover some interesting information about which machines limited the throughput—it was originally thought that the press would be the cause of bottlenecks, but this was proved wrong and the surprising result that it was sufficient to speed up the press and the crane in order to improve significantly the throughput of the cell and the press utilisation was obtained. In this respect the case study has been a success.

6.2. Future work

Now that a specification has been produced which yields performance measures, this work can continue in a number of directions. Firstly it would be desirable to analyse the specification to determine liveness and safety properties of the controller it describes. It should be possible to determine liveness properties from the model as it stands—the performance experiments show that the cell does produce processed blanks, and that it is possible to determine how well utilised individual components are. Safety properties, such as determining that blanks are never dropped in unsafe places, may require that the specification is transformed to a standard process algebra (by removing all performance information) and analysed used standard techniques. Following on from this analysis the specification should be implemented and interfaced to the simulation, demonstrating that the controller described can usefully control the cell. It would also be desirable to control the physical cell as a simulation is always an approximation to a real system.

Another useful direction would be to investigate how to optimise the rates of the individual machines to maximise the throughput at the minimum cost. The main difficulties here lie in:

- the computational complexity of the task; and
- determining the cost of operating a given machine at a particular speed.

The complexity of optimising a cell is related to the number of components—the cell described in this paper has only seven components, so the optimisation task is tractable, but it should be noted that the size of the search space grows exponentially with the number of components.

Determining the cost of operating a machine at a given given speed is quite complex. In the case that a particular machine can operate at variable speeds the information required would include factors such as how its failure rate changes as the speed of the machine varies, and perhaps how the running costs vary with speed. For types of machines which run at a fixed speed the consequence of requiring a faster machine might be buying a more expensive model, meaning that capital costs would need to be factored into the optimisation procedure.

7. ACKNOWLEDGEMENTS

I am very grateful to Jane Hillston and Stephen Gilmore for introducing me to PEPA, for guiding my first faltering steps with the language and for the work Stephen especially put in on dragging this specification kicking and screaming though the PEPA workbench, which allowed me to calculate the performance figures given in section 5.

TABLE 1. Table showing values assigned to rate parameters

Parameter	Value (sec^{-1})
F_T	0.503524672708963
F_B	0.857632933104631
T_R	0.521104742053153
T_T	0.353232073472271
R_1	0.232774674115456
R_2	2.17256362507759
R_3	0.407355679702048
R_4	0.724187875025864
$A1_{T1}$	1.16083347843752
$A1_{T2}$	0.633181897329555
$A2_{T1}$	0.648735319687359
$A2_{T2}$	1.63530586106703
$A2_{T3}$	0.464475281577689
P_{b2m}	0.965250965250965
P_{m2t2b}	0.317258883248731
D_B	0.713266761768902
D_T	0.34965034965035
C_H	0.141924496168039
C_V	0.372300819061802
ξ	100

REFERENCES

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

[2] S.T. Gilmore, J.E. Hillston, D.R.W. Holton, and M. Rettelbach. Specifications in a Stochastic Process Algebra for a Robot Control Problem. Technical Report ECS-CSG-7-94, Edinburgh University, 1994. Accepted for publication by the **International Journal of Production Research.**

[3] M. Broy and S. Jähnichen, editors. *Korrekte Software durch formale Methoden.* Technische Universität Berlin, 1993.

[4] Claus Lewerentz and Thomas Lindner. *Case Study "Production Cell": A Comparative Study in Formal Software Development.* FZI-Publication, 1994.

[5] 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.

[6] ISO. *Information Processing Systems—Open Systems interconnection—LOTOS, A Formal Description Technique based on the Temporal Ordering of Observational Behaviour.* ISO, 1987.