Extended Petri-Net Modeling for Re-manufacturing Line Simulation

Gábor Erdős Paul Xirouchakis
CAD/CAM laboratory,Swiss Federal Institute of Technology at Lausanne, Switzerland
EPFL-ICAP-LICP phone:(+41-21) 693 2930, fax (+41-21) 693 3553
{*Gabor.Erdos, Paul.Xirouchakis*}*@epfl.ch*

Abstract

We address the problem of building, simulating and improving the performance of a remanufacturing line computer model. We present a parametric Petri net model of a remanufacturing line: Places represent precedence constraints on operations while transitions model remanufacturing operations. Basic parameters of the Petri net model are the token type (its color), the capacity of places and the priority and firing duration of transitions. We use different token colors to model the different states of a product through the test and repair loop. The capacity parameter is used to model the grouping of products into boxes, that facilitate transportation. The priority of operations is used to resolve conflicts among alternative operations. We simulate the operation of the parametric Petri net model using the software POSES++ for a semi-automated telephone remanufacturing line test case. From the simulation experiments of the case study it turns out that throughput performance is not very sensitive to the duration of testing, cleaning or packing and optimal performance can be achieved when the ratio of disassembly to reassembly duration is close to one. We extend our previous base model to include the effects of also feeding the line with new covers and electronic circuits to avoid line starvation.

1. Introduction

There is a growing need in the industry to plan the end of life phase of products. Among many reasons are the costs associated with the treatment of products at the end of their life and the increasing awareness of environmental impacts of discarded products. Planning the re-manufacturing process cause specific problems which are uncommon in current process planners. The re-manufacturing process is essential for material and components isolation, since the objective of manufacturers is to increase the reusable or recyclable components and subassemblies, while reduce the disposed materials. Along with these two goals, the awareness of environmental impacts, is becoming an important factor in design for re-manufacturing too.

There are basically two different ways for tackling these requirements. First is the planning of the re-manufacturing process. This can be either predictive like in [2, 7, 9, 11]or adaptive (re-planning) like in [6, 10]. The second approach is the evaluation of the end of life impact of products, as overviewed in [1, 3, 5].

This work develops a Petri Net based evaluation tool for re-manufacturing processes. We start with a real world example and build its simulation model. Subsequently we perform several simulation experiments with different sets of model parameters and we study the remanufacturing line throughput performance in terms of the remanufactured telephones per unit time. We extend our previous base model [4] to include the effects of also feeding the line with new covers and electronic circuits to avoid line starvation. The computer simulations show that this extended model is capable to balance the re-manufacturing line even in case of different failure ratios.

2. Re-manufacturing Line Model

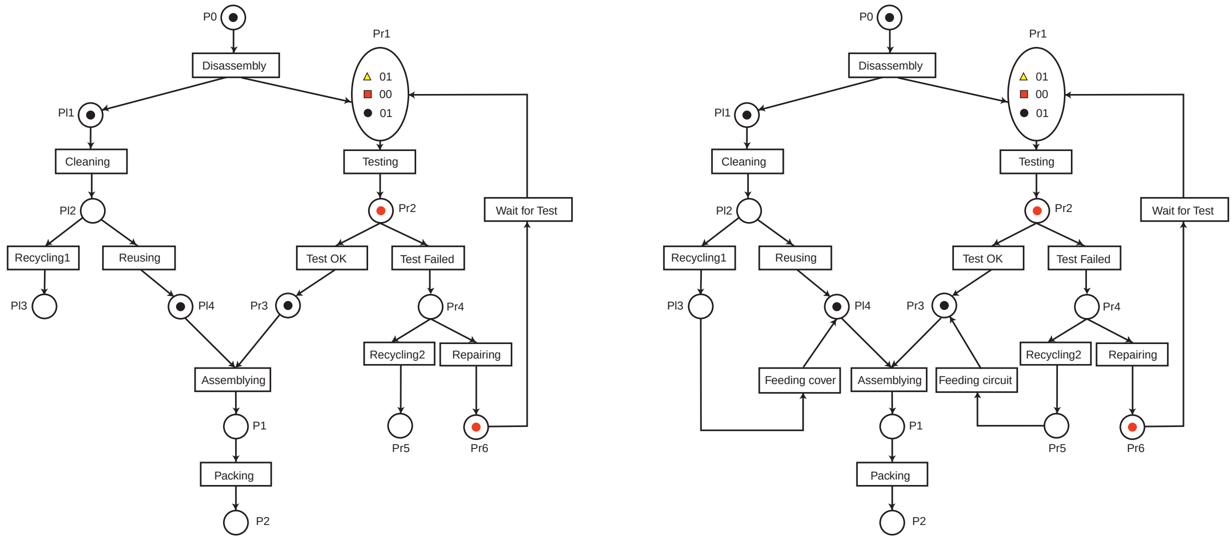
The modelled re-manufacturing line is a semi-automated telephone disassembly/reassembly line. This process has the following main steps:

1. Disassemble the telephone into plastic cover and electrical board components.
2. Test the electrical board
   - If the test is OK send to reassemble
   - If the test fails send the board to repair, or if the board has failed on the test already two times then send it to recycling
3. Clean the plastic cover part
   - If the cleaned part is reusable, then send to reassemble
   - If the cleaned part is not reusable , than send to recycling
4. Reassemble the telephone.

5. Pack the reassembled telephone into boxes

2.1. Base Model

To model the re-manufacturing process we propose the construction of a Petri Net model. A Petri Net consists of places and transitions, which are linked to each other by, directed arcs. Input arcs are directed from places to transitions, while output arcs are directed from transitions to places. Places represent precedence conditions while transitions represent process operations. A transition has certain number of input and output places representing the pre-conditions and post-conditions of the event, respectively. The presence of a token in a place is interpreted as satisfying the truth of the precedence condition associated with the place. Tokens represent and carry information. Sophisticated Petri nets, like colored Petri nets can attach features to tokens, which simplifies the topology. It can be proved that these featured Petri nets can be realized with simple Petri nets also, which use only ”black” tokens.



(a) (b)

Figure 1: Base and Extended Petri-Net Model

The POSES++ program [8] allows us to use the following main features:

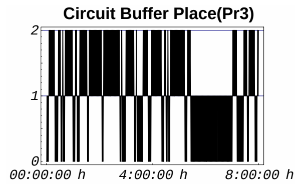
- Defining featured tokens: black (i.e. tokens that carry no information), integer type, string type, structure type, etc.
- Defining the capacity of places - the maximum number of tokens a place can incorporate simultaneously.
- Defining the access modes of places - How the tokens are to enter the places waiting queues, and in what sequence the queues will be searched and the tokens removed when the associated transitions fire. In this paper we have selected the random access mode for the test case example.
- Timed transitions.
- Defining the funcionality features of the transitions, like the priority, transition time duration, etc.
- Assigning priorities for competing transitions - As long as no priorities are given, the random number generator decides on an equal chance basis which of the competing transitions is to be investigated first.
- Implementing function calls, when transitions fire. This functionality is utilized in the test case example to dynamically calculate the priorities of the conflicting transitions (*Recycling1*, *Reusing*) and (*TestOK*, *TestFailed*).

The Petri-Net model of the base model is displayed in Figure 1.a. This model has the following features:

Every place accommodates black tokens except *Pr1*, *Pr2*, *Pr4*, and *Pr6* which are specified of the integer type to model the number of times that a token enters the test loop. The reason is that, the electrical boards that fail on the test (defective boards) can be send to repair only a fixed number of times; therefore after each repair, the ”color” of the token changes until it satisfies the input requirements of the *Recycling2* transition, which are has token color requirements of integer type with a value equal to two. As shown in Figure 1, we use the following three token colors: ● for black tokens, △ for tokens that failed the test the first time and ■ for tokens that failed the test the second time. These three token types correspond to the values 0,1 and 2 of the integer token type.The model uses parametric values for capacity of places;priorities between competing transitions like *TestOk* or *TestFailed* and *Recycling1* or *Reusing*; transition duration times for every transition.

Parameter Table

Disassembly time	Cleaning time	Testing time	Repairing time	Assembly time	Packing time	Circuit feeding time	Cover feeding time
115 sec	50 sec	40 sec	110 sec	60 sec	60 sec	n. a.	n. a.



Simulation Results

Assembled telephones	TestOK	Recycling1	Recycling2
	TestFailed	Reusing	Repairing
247	249	2	1
	51	247	51

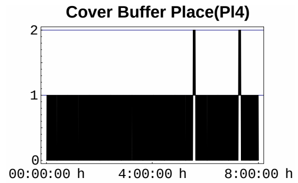


Figure 2: Experiment 1- Original parameter set with no refeeding

The capacities of places *P0*, *Pr5* and *P2* are considered infinite since they represent infinite capacity buffers. For places *Pl3*, *Pl4*, *Pr3* and *Pr4* the capacity is considered infinite because otherwise the statistical distribution of tokens in the preceding conflicting transitions could not be easily implemented. The capacity of place P1 is selected as 120 to model a finite capacity of this buffer (equal to the number of telephones in a box).

The model has two conflicting transition pairs (*Recycling1*, *Reusing* and, *TestOK*, *TestFailed*), where one predicate place (*Pl2* or *Pr2* respectively) can fire either one of the conflicting transitions. To model the probabilistic nature of firing we introduced a statistical function for deciding, which transition will fire. Since the POSES++ program provide us only with the priority feature to resolve the conflicting transitions, we wrote a simple C routine, which sets the priorities on-line. There is one important requirement for using the conflict resolving routine.

2.2. Extended Model

The output places of the transition pairs should have infinite capacity, otherwise the decision, which transition to fire, would depend on the number of tokens of the output places.

The transportation time is not considered explicitly in these experiments but it is taken as part of the assigned duration times in transitions.

We extend our base model in order to include the effects of also feeding the line with new covers and electronic circuits to avoid line starvation. The base model has been constructed based on the real re-manufacturing line. Its parameters are tuned to match the real parameters [4]. This model is capable to model hypothetical changes, since their parameters are matched against the real example. Therfore the result of the simulations are expected to be more accurate. The figure of the extended model is displayed in Figure 1.b.

The two new transition feed the line with new cover boxes and circuits, whenever they are recycled. We wanted to investigate the effect of the feeding when the *TestOk/TestFailed* ratio is changing.

3. Experiments

We use different parameter values during the simulation experiments, in order to determine the behavior of the system. All experiments have been conducted for a simulation time of 8 hours. The results are shown in tables and in graphs. The graphs show the occupancy of bottleneck places (Pl4 and Pr3) during the time of simulation. The first experiment (Figure 2)shows the performance of the re-manufacturing line based on the tuned parameters. The recycled cover boxes and circuits are not re-feeded, but the line is balanced as the history diagrams of the two bottleneck places show. This balanced performance is the result of the *TestOk/TestFailed* ratio which has been set to roughly 3:1. When this ratio is changed - we set this ratio to 2:1 - the bottleneck places filled up quickly and the re-manufacturing line shows an unbalanced performance even if the total number of re-assembled telephones does not reduce considerably (see in Figure 3).

Parameter Table

Disassembly time	Cleaning time	Testing time	Repairing time	Assembly time	Packing time	Circuit feeding time	Cover feeding time
115 sec	50 sec	40 sec	110 sec	60 sec	60 sec	n. a.	n. a.

Simulation Results

Assembled telephones	TestOK	Recycling1	Recycling2
	TestFailed	Reusing	Repairing
239	240	8	10
	131	241	121

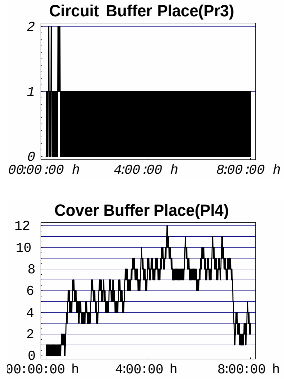


Figure 3: Experiment 2- Modified *TestOK/TestFailed* ratio with no refeeding

The base model is extended by introducing the refeeding transitions to avoid the line starvation in case of variable *TestOk/TestFailed* ratio. At first we allowed *slow* refeeding of the recycled parts. This means that each recycled cover box and circuit are replaced by a new one. The feeding time has been set to 240 sec. From the result of the experiment (see Figure 4) one can see, that the occupacy of the two bottleneck places are much less during the simulation. Also the total number of re-assembled telephones basically does not change comparing to the first experiment (see Figure 2).

Parameter Table

Disassembly time	Cleaning time	Testing time	Repairing time	Assembly time	Packing time	Circuit feeding time	Cover feeding time
115 sec	50 sec	40 sec	110 sec	60 sec	60 sec	240 sec	240 sec

Simulation Results

Assembled telephones	TestOK	Recycling1	Recycling2
	TestFailed	Reusing	Repairing
249	231	3	18
	146	246	128

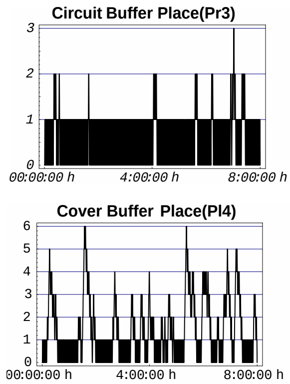


Figure 4: Experiment 3- Modified *TestOK/TestFailed* ratio with slow refeeding

The performance of the line can be further improved by allowing *faster* refeeding as it is shown in Figure 5. The refeeding time in this case is 30 sec. The circuit buffer place (Pr3) is utilised optimally in this experiment, but the cover buffer place (Pl4) shows slight starvations. The refeeding transitions in this way can not balance completely the line starvations. To fix eliminate the waiting of the circuit we trird to refeed both circuits and cover boxes only if there are no reused ones in the buffer places. The result of the this simulation(Figure 6) shows that this change has no effect on the performance. We tried to change the type of the transition in order to avoid circuit starvation of the re-manufacturing line. The new type of transition refeed two pieces of new circuit and cover box for each recycled ones. We expected that this will solve the line starvation, but as the experiment shows in Figure 7 the performance of the line degraded instead of improving. Even if the total number of re-assembled telephones are slightly increased the utilisation of the bottleneck places are much more intensive than in the previous cases. Also introducing the double feeding causes cover box starvation instead of the circuit starvation of the previous experiment.

Parameter Table							
Disassembly time	**Cleaning time**	**Testing time**	**Repairing time**	**Assembly time**	**Packing time**	**Circuit feeding time**	**Cover feeding time**
115 sec	50 sec	40 sec	110 sec	60 sec	60 sec	30 sec	30 sec

Simulation Results			
Assembled telephones	**TestOK** / **TestFailed**	**Recycling1** / **Reusing**	**Recycling2** / **Repairing**
248	239	14	9
	111	235	102

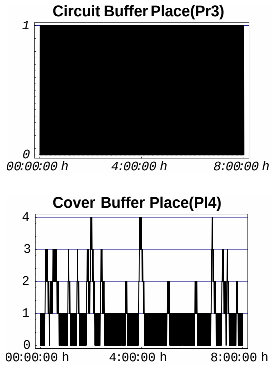


Figure 5: Experiment 4- Modified *TestOK/TestFailed* ratio with fast refeeding

Parameter Table							
Disassembly time	**Cleaning time**	**Testing time**	**Repairing time**	**Assembly time**	**Packing time**	**Circuit feeding time**	**Cover feeding time**
115 sec	50 sec	40 sec	110 sec	60 sec	60 sec	30 sec	30 sec

Simulation Results			
Assembled telephones	**TestOK** / **TestFailed**	**Recycling1** / **Reusing**	**Recycling2** / **Repairing**
249	238	1	12
	110	248	98

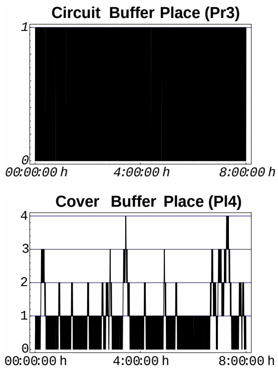


Figure 6: Experiment 5- Modified *TestOK/TestFailed* ratio with fast refeeding if there is no reused part

Parameter Table							
Disassembly time	**Cleaning time**	**Testing time**	**Repairing time**	**Assembly time**	**Packing time**	**Circuit feeding time**	**Cover feeding time**
115 sec	50 sec	40 sec	110 sec	60 sec	60 sec	30 sec	30 sec

Simulation Results			
Assembled telephones	**TestOK** / **TestFailed**	**Recycling1** / **Reusing**	**Recycling2** / **Repairing**
253	243	4	16
	137	245	121

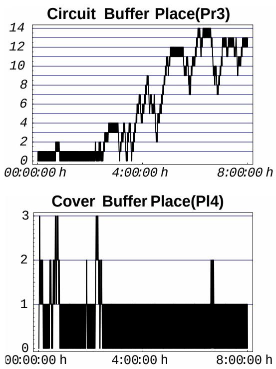


Figure 7: Experiment 6- Modified *TestOK/TestFailed* ratio with double refeeding

4. Conclusion

We extend our previous base model of an existing telephone re-manufacturing line by including refeeding transitions of circuit and cover boxes. We also investigated the effect of these transitions on the performance of the re-manufacturing line in case of changing TestOk/TestFailed ratio. From the results of the simulations we arrive to the conclusions that the refeeding transitions can improve the performance of the line, but it can not solve the line starvation completely. We also tried to double feeding the line, but it turned out from the experiments that this modification degrades the performance of the re-manufacturing line.

References

[1] B Alting, L.and Legarth. Life cycle engineering and design. *Annals of CIRP*, 44(2):41–48, 1995.

[2] E. Arai and K. Iwata. Cad system with product assembly/disassembly planing. *Robotics and Computer-Integrated Manufacturing*, 12(11):41–48, 1992.

[3] H. J. Bullinger, J. Warschat, J. Bopp, and Wörner K. Approaches to product life-cycle cost estimation in concurrent engineering. *Proceedings of the 3rd International Conference on Concurrent Engineering and Electronic Design Automation*, pages 118–122, April 1996.

[4] Gábor Erdős and Paul Xirouchakis. Petri-net modeling for re-manufacturing line simulation. In *Proceedings of the First International Workshop on Intelligent Manufacturing Systems*, pages 351–361, Lausanne,Switzerland, April 1998.

[5] K. Feldmann and O Meedt. Determination and evaluation of the optimal end of life strategy for products based on simulation of disassembly and recycling. In *Proceedings of 4th International Seminar on Life Cycle Engineerig*, pages 145–156, Berlin,Germany, June 1997.

[6] D. Geiger and E. Zussman. Probabilistic reactive disassembly planning. *Annals of the CIRP*, 35(1):9–12, 1996.

[7] L. Laperriere and H.A. ElMaraghy. Planning of products assembly/disassembly. *Annals of the CIRP*, 41(1):5–9, 1991.

[8] GPC mbH. *POSES++ Online Manual*, 1997.

[9] P. Xirouchakis and D. Kiritsis. A generic petri net modell for disassembly planning process. In , *Proc. TMCE'96*, Budapest,Hungary, May 1996.

[10] E.F. Zuelch, G.and Schiller and M. Schneck. Adaptive dynamic process plans- a basic for a disassembly information system. In *Proc. of the CIRP Int. Seminar on recycling*, pages 245–253, 1995.

[11] E. Zussman and M. Meltser. Planning and re-planning strategy of disassembly processes. In *Proceedings of 4th International Seminar on Life Cycle Engineerig*, pages 283–292, Berlin,Germany, June 1997.