Simulation of a disassembly-to-order system

Ying Tang*, Ludivig J. Ungewitter, Tobi Mann, and Tosh Kakar

*Dept. of Electrical & Computer Engineering
Rowan University
Glassboro, NJ 08028
Email: tang@rowan.edu

Dept. of Computer Science & Computer Engineering
Pacific Lutheran University
Tacoma, WA 98445

Abstract

This paper focuses on the project of design and simulation of a disassembly-to-order system that provides a unique "hands-on and minds-on" research experience for undergraduate students. This project is completed by a multidisciplinary group of faculty and students from Electrical & Computer Engineering at Rowan University and Computer Science and Computer Engineering at Pacific Lutheran University (PLU). In such a system, the disassembly of discarded products is processed to satisfy certain demands for parts and/or materials, while economic and environmental goals are achieved. Two scenarios in the system are analyzed and their performance is compared.

1. Introduction

Due to the increased awareness of the state of environment by consumers and manufacturers, End-of-life (EOL) options for discarded products and materials have become emerging areas of engineering research [4]. This necessitates a certain level of partial or complete disassembly depending on the type and level of demand for used products, components and /or materials [1]. Since disassembly paths and termination goals are not necessarily fixed, automation of disassembly is extraordinarily difficult. Moreover, the current disassembly systems exhibit a high degree of inflexibility towards variations in demands. Regardless of product condition, a used product is usually completely disassembled through a fixed process flow. Therefore, it is necessary to develop an efficient disassembly process plan from a system perspective to minimize both labor and environmental costs. In our previous work, a systematic approach to disassembly line design is proposed, which executes a disassembly-to-order model rather than disassembly-to-stock [2]. Disassembly, being a labor-intensive operation, is expensive to deploy. Hence, it is critical to conduct performance analysis before a process plan is applied to the shop floor.

This project addresses this issue by designing and simulating a disassembly-to-order system. Murdock Undergraduate Research Program at PLU supports part of the project and the other part is finished over the course of two-semester Engineering Junior/Senior Clinic at Rowan University as part of their design sequence. The rest of the paper is organized as follows. Section 2 gives a brief introduction of the research environment at two universities. The concept of disassembly lines is then presented in

Section 3. Section 4 specifies simulation design. The system performance in two scenarios is evaluated in Section 5, and the conclusion is given in Section 6.

2. Collaboration Environment to Foster Undergraduate Research

There is no better way to learn science and engineering than by actually doing it by yourself. To give students in the Natural Science division at Pacific Lutheran University the chance to explore some of the questions that currently intrigue researchers, the division has established a coordinated undergraduate research program. Students in this program work closed with a faculty supervisor, conducting scientific research for 10 weeks during the summer. Weekly seminars are held to promote interactions among students and faculty from all natural science departments, who are working on diverse projects. These seminars help build a community of learners and establish a research culture in the division. It is under this circumstance that this project was first started.

The Engineering Clinic is the hallmark of the Rowan Engineering program. It is a series of courses taken each semester by every engineering student. In the Engineering Clinic, which is based on the medical school model, students and faculty from all four engineering departments work side-by-side on multidisciplinary/interdisciplinary laboratory experiments, design projects, applied research, and product development. While each clinic course has a specific theme, the underlying concept of engineering design pervades throughout. The clinic progression gives us a way to systematically develop our students as collaborative designers. Freshmen Engineering Clinic introduces design through reverse engineering; at the sophomore level, students learn structured design and get their first, open-ended project experience. Students in the Junior and Senior Engineering Clinic work in multidisciplinary design teams on projects of progressive complexity. The four-year, 20-credit (Freshman 2+2, Sophomore 4+4, Junior 2+2, Senior 2+2) Engineering Clinic sequence offers students the opportunity to incrementally learn the science and art of design by continuously applying the technical skills they have obtained in traditional coursework. This project is continuously finished as a Junior/Senior multidisciplinary design project.

3. The concept of disassembly lines

A disassembly line is configured as a sequence of workstations, each with one or more machines/workers [2]. Because of uncontrolled variability in terms of product type and condition, a preliminary inspection is performed off-line in a disassembly-to-order system. In this procedure, a disassembly Petri Net (DPN) model is used to derive a disassembly sequence for each used product with EOL consideration [2-4]. The concept of value gain is introduced to get the trade-off between disassembly cost and benefit gained [2-3]. After the off-line inspection, discarded products with the same disassembly path and the final disassembled parts are stored into different input inventories. When a demand comes, an input inventory is chosen to provide the source for this demand, where the total number of products needed is smaller and the total value gain is higher. Then, based on the system status and the demand information, a disassembly line is configured and dedicated to disassemble these products. As soon as the demand is satisfied, its corresponding disassembly line can be revised or dismissed, and then the groups of equipment/resources are reassigned to other disassembly lines. Fig 1 gives overall flow chart for the disassembly line design.

4. Simulation Design for a disassembly-to-order system

To simulate a disassembly-to-order system and provide the performance evaluation for the disassembly line design method, a simulation package written in Java is developed. The software architecture includes the following modules shown in Fig. 2:

- Database
- Primary Inspection Unit
- Input Inventory

- Output Inventory
- Factory

A. Database

Database consists of three parts. Product Database provides the relevant information of each discarded product, including EOL and cost values in its DPN. Order Database keeps order information, including its size, due date, and priority. Machine information is saved in Machine Database, which includes machine functionality, machine class types, and machine pool size. The data is real-time populated through *ProductInfo*, *ValueGenerator*, and *Machine* classes.

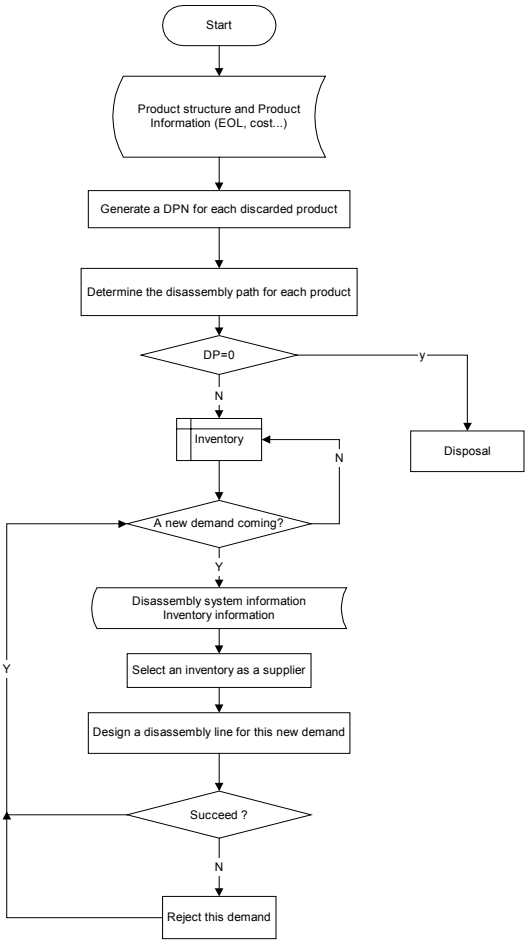

Fig.1: The flow chart of a disassembly line configuration

B. Primary Inspection Unit

Based on product information from Database, a hash table is used to implement a DPN structure in the primary inspection unit. The disassembility for each product is first checked, and then its optimal disassembly path is obtained. To fulfill these tasks, five classes are defined, where *Node, NodePointer, LeafNode, and PetriNet* classes are for the DPN modeling, and *PrimaryInspectionUnit* for the optimal disassembly planning.

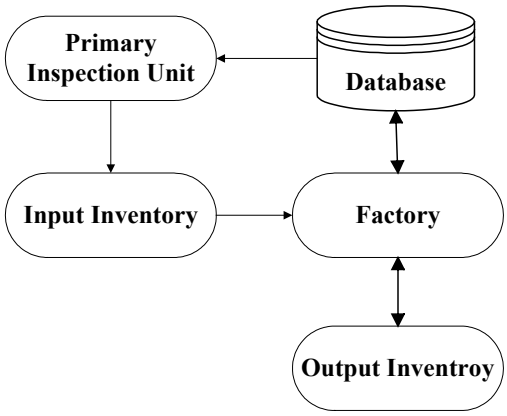

Fig. 2: The simulation software architecture

C. Input Inventory/Output Inventory

Based on primary inspection results, products are stored into corresponding input inventory slots. Their information is held there through *InspectedProduct and InventoryHelper* classes. Output Inventory model simply collects disassembled components before they are shipped out.

D. Factory

In the Factory module, three major classes are defined to implement disassembly line design algorithm in a disassembly shop floor. They are listed as follows:

- MachinePool: keeps information about machine availability.
- Workstation: keeps information of a workstation, including the number of machines, machine types etc.; conducts processes in this workstation; and adjusts its speed by adding or removing machines.
- DisassemblyLine: configures a disassembly line for a certain demand and handles its actual processes.

Fig. 3 shows the whole class structure for the simulation software. Since Java is an object-oriented language, these modules can be easily modified and used to simulate a large variety of conditions in a disassembly-to-order system. Furthermore, Java is platform independent that further extends the portability of the software.

5. Performance Evaluation

Using this software, two scenarios in a disassembly-to-order system are simulated at the same time to disassemble batches of personal computers. In the traditional case, demands are successively processed according to their order priorities. In the proposed case, our disassembly line methodology is applied to the system. Whenever a demand comes, the system configures a disassembly line on the fly based on the inventory information and system status, and adjusts the speed of workstations to maintain line balance.

During the implementation of this software, the input data is randomly populated into the Database in the following fashion:

- Orders enter the system with the fixed due date and different order size. The order size has a uniform distribution between [10000, 40000];
- In a DPN, the EOL value of each product, subassembly, and component, and the cost value of each disassembly task are both randomly chosen with a uniform distribution, which have the range [-2, 1] and [0.001, 0.1], respectively;

- The process time for each disassembly task has a uniform distribution between [0.1, 5.5]; and
- Users can define the number of workstations and machine types in the module, and change the number of machines in each machine class through the interface shown in Fig. 4. In this example, the system has four workstations and four types of machines.

To show the difference between these two scenarios, the slack time (ST), earliness (ET) and tardiness (TT) definitions are introduced as follows:

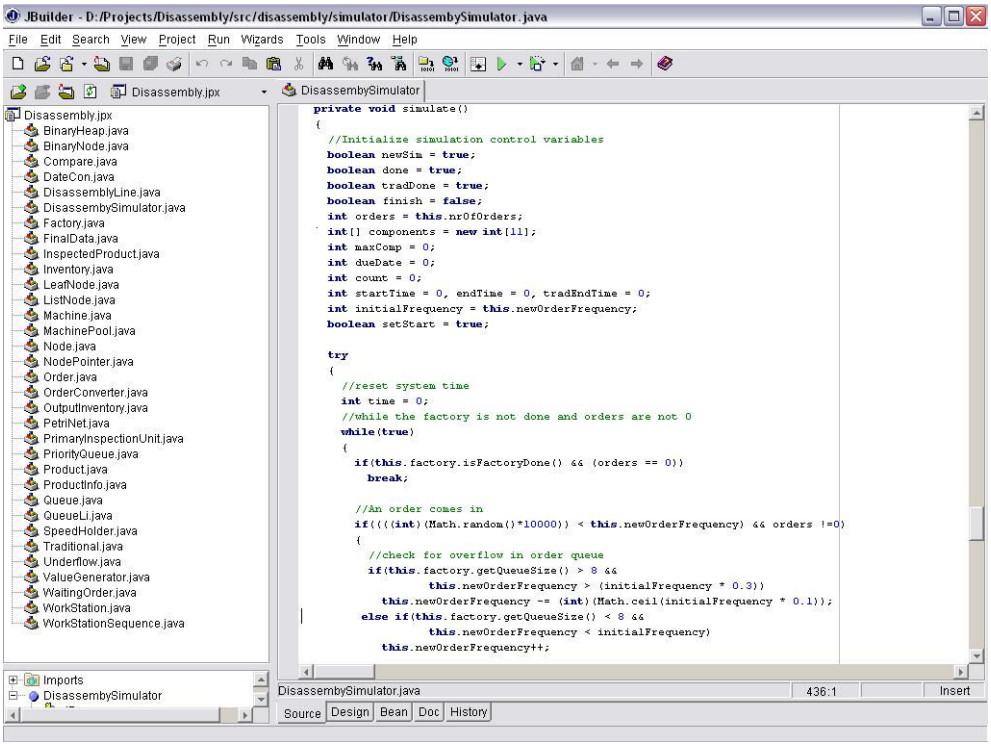

Fig. 3. Simulation software workspace for a disassembly-to-order system

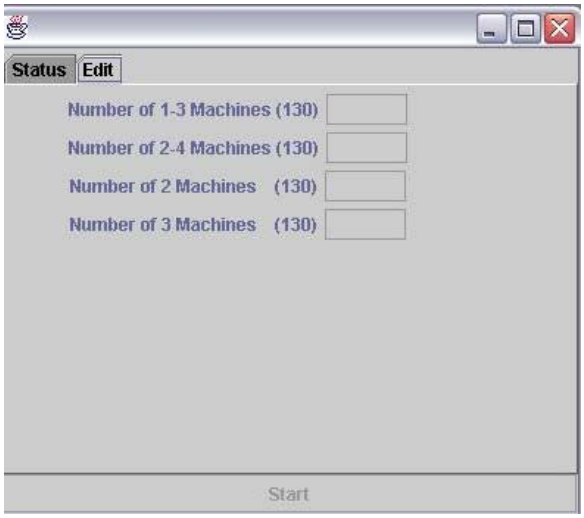

Fig. 4 Interface to modify the number of machines in each machine class

ST = Due(w) - Com(w)
ET = *Max* {ST, 0}
TT = *Max* {-ST, 0}

where Due (w) is the due date for the w^{th} order, and Com(w) the completion time for the w^{th} order. Finally, two alternatives in real-time procedures are compared through a set of experimental simulation.

In the first set of experiments, as the number of demands changes from 10 to 100, the total value gain of the system and the speeds of disassembly lines are obtained and compared between the traditional and the proposed cases shown in Figs. 5 and 6. Since the uncertainty of product structure and condition is considered in the preliminary inspection, the disassembly system is much more profitable in terms of disassembly costs and revenues.

Fig. 5: Value gain comparison between the proposed and the traditional cases

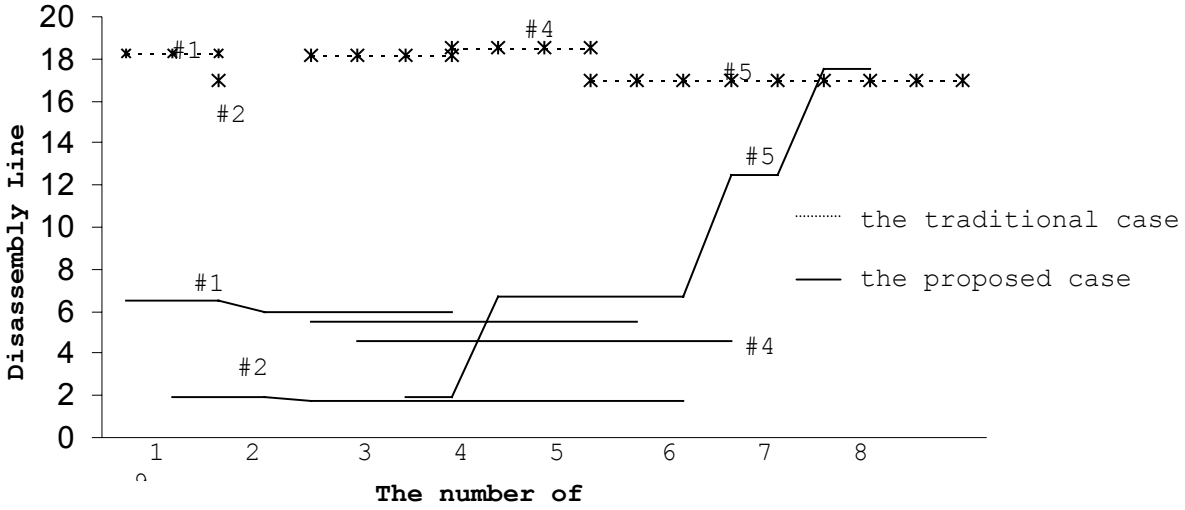

Fig. 6 Speeds of disassembly lines for different cases when the system processes five orders (#1-#5 are referred to orders 1-5)

For simplicity, only the case when the system deals with 5 orders is considered in Fig. 6. The corresponding data for these orders are listed in Table 1. In the traditional scenario, demands are successively processed in the system. All the resources in the shop floor are fully dedicated to one demand at one time. In the proposed case, the disassembly line design algorithm helps the system to concurrently handle multiple demands. Speeds of disassembly lines are adjusted to maintain line balance, according to their condition, information of the corresponding demands and system capacity. The key

issue in this design is to keep the speed of a disassembly line to be as close as possible to its desired speed. The desired speed of a disassembly line is the minimal speed that ensures its corresponding order to be finished on time.

Table 1: The input data for five orders

Order	Order date	Due date	Order Size	
1	01/01/00	01/04/00	Hard Drive: 20200 Display Adapter: 20100 Supply Unit: 19600 Case: 24300	CD-ROM: 32400 Network Adapter: 33200 Memory Extension: 22300 Cables: 23700
2	01/01/00	01/07/00	Motherboard: 15985 Hard Drive: 18200 Network Adapter: 15200 Case: 14000 Cables: 14000	Disk Driver: 11924 Audio Card: 20110 Supply Unit: 17600 Memory Extension: 12300
3	01/03/00	01/08/00	Disk Driver: 21600 Display Adapter: 24300 Supply Unit: 31800 Case: 26300,	CD-ROM: 35400 Audio Card: 19200 Memory Extension: 36400 Cables: 22100
4	01/03/00	01/08/00	Motherboard: 27900 Hard Driver: 16200 Audio Card: 30800 Supply Unit: 26800 Case: 27400	Disk Driver: 17700 CD-ROM: 17300 Network Adapter: 23100 Cables: 29300
5	01/03/00	01/09/00	Motherboard: 44400 Hard Drive: 29600 Supply Unit: 38400 Case: 64600,	Disk Driver: 36700 CD-ROM: 67900 Display Adapter: 78100 Cables: 52800

In the second set of experiments, the earliness and tardiness are compared between these two scenarios, where the number of orders increases from 20 to 100. Based on the simulation results shown in Fig. 7, the proposed case always has smaller earliness than the traditional case. Consequently, a smaller output inventory reduces the storage cost in the proposed scenario. There is no pattern visible regarding the tardiness.

4. Conclusion

This work develops simulation software using Java to analyze a disassembly-to-order system. The system performance is evaluated and compared between two scenarios. In the proposed scenario, the methodology proposed in our previous work is found to be effective in maintaining a better trade-off between disassembly costs and revenues and reducing inventory costs. Students participated in this project gain invaluable experience in disassembly research and have a better understanding of Java programming. More specifically, the multidisciplinary collaboration between two universities provides a unique paradigm in engineering education, where a group of individuals work together on inquiry, exploration, and sharing of knowledge. It further promotes undergraduates a strong interest in post-graduate education. The two student authors who participated in the project are currently studying in the graduate school.

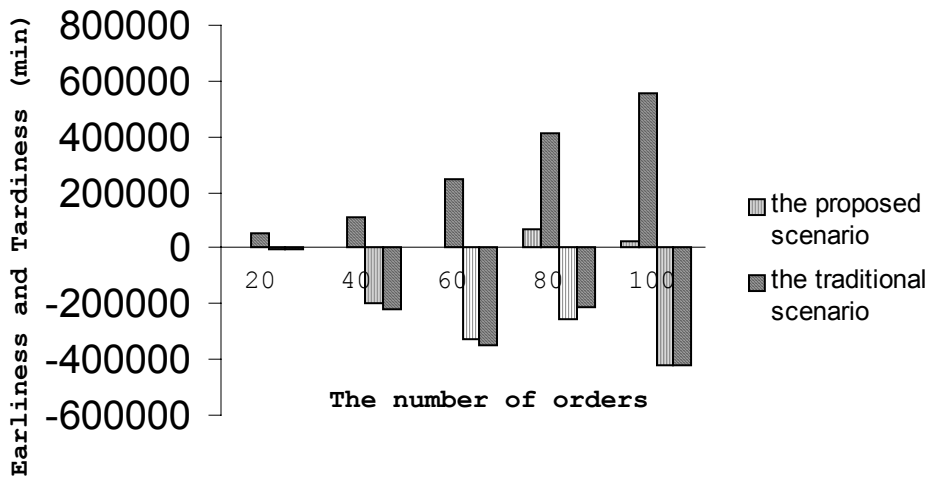

Fig. 7 Earliness and tardiness comparison between the proposed and the traditional cases

Reference

[1]. Gupta, S. M. and Kongar, E., "A disassembly-to-order system," *Proc. Of IV SIMPOI/POMS,* Guaruja/SP, Brazil, Aug. 11-14, 2001, pp. 331-338.

[2]. Tang, Y., Zhou, M. C. and Caudill, R., "A systematic approach to disassembly line design," *Proc. of IEEE Int. Conf. on Electrical & Environment,* Denver, CO, May 7-9, 2001, pp. 173-178.

[3]. Tang, Y., Zhou, M. C. and Caudill, R., "An Integrated Approach to Disassembly Planning and Demanufacturing Operation," *IEEE Trans. on Rob. & Aut.*, 17(6), December 2001, 773-784.

[4]. Zussman, E. and Zhou, M.C., "Design and implementation of an adaptive process planner for disassembly processes," *IEEE Trans. on Rob. & Aut*., Vol. 16, No. 2, April, 2000, pp. 171-179.

Ying Tang is Assistant Professor of Electrical and Computer Engineering at Rowan University, Glassboro, Nj. She received the B.S. and M.S. degrees from the Northeastern University, P. R. China, in 1996 and 1998, respectively, and Ph. D degree from New Jersey Institute of Technology, Newark, NJ, in 2001. Her research interests include modeling and design of computer-integrated systems, Petri nets, Networking and communication, and FPGA design.

Ludivig J. Ungewitter is currently a graduate student of Computer Science department at Columbia University.

Tobi Mann is currently a graduate student of Computer Science and Engineering department at Washington University in St. Louis.

Tosh Kakar is Assistant Professor of Computer Science and Computer Engineering at Pacific Lutheran University.