January 01, 2003

A multi-kanban model for disassembly

Surendra M. Gupta
Northeastern University

Gun Udomsawat
Northeastern University

Sagar V. Kamarthi
Northeastern University

Laboratory for Responsible Manufacturing

Bibliographic Information

Udomsawat, G., Gupta S. M. and Kamarthi, S. V., "A Multi-Kanban Model for Disassembly", ***Proceedings of the 2003 Northeast Decision Sciences Institute Conference***, Providence, Rhode Island, pp. 169-171, March 27-29, 2003.

Copyright Information

Contact Information

Dr. Surendra M. Gupta, P.E.
Professor of Mechanical and Industrial Engineering and
Director of Laboratory for Responsible Manufacturing
334 SN, Department of MIE
Northeastern University
360 Huntington Avenue
Boston, MA 02115, U.S.A.

(617)-373-4846 ***Phone***
(617)-373-2921 ***Fax***
gupta@neu.edu ***e-mail address***

http://www.coe.neu.edu/~smgupta/ ***Home Page***

A MULTI-KANBAN MODEL FOR DISASSEMBLY

Gun Udomsawat, Northeastern University, Boston, MA 02115, (617)-373-7635, gudomsaw@coe.neu.edu
Surendra M. Gupta*, Northeastern University, Boston, MA 02115, (617)-373-4846, gupta@neu.edu
Sagar V. Kamarthi, Northeastern University, Boston, MA 02115, (617)-373-3070, sagar@coe.neu.edu
(***Corresponding author)

ABSTRACT

In this paper we demonstrate how kanbans can be used in a disassembly line setting. Disassembly line is quite different from assembly line in terms of material movement, demand arrival and inventory level fluctuation. We discuss these differences and introduce the concept of a multi-kanban mechanism to cope with them. With the help of a simulation model (developed using the ARENA® software), we show that the modified kanban mechanism is in fact an effective tool for a disassembly line setting. An example is presented to illustrate the concept.

INTRODUCTION

In recent years, the continuous growth in consumer waste has seriously threatened the environment. Realizing this, many countries have passed regulations that force manufacturers to take back the used products from consumers so that the components and materials recovered from the products may be reused and/or recycled. Disassembly plays an important role in a product recovery environment. A disassembly line is perhaps the most suitable setting for disassembly of products in large quantities.

Just-in-time philosophy evolved from a number of principles such as the elimination of waste, reduction of production cost, total quality control and recognition of employees' abilities and has been implemented in several types of production system for many decades. Some advantages of JIT include its simplicity in production scheduling, reduced burden on operators, ease of identification of parts by the kanbans attached to the containers and substantial reduction in paper work. In this paper we demonstrate how kanbans can be implemented in a disassembly line setting.

Disassembly line is quite different from assembly line in terms of material movement, demand arrival and inventory level fluctuation. For example, not only can the demand arrive at the last station, it can also arrive at any of the stations in the system. In fact, there are many other complicating matters that need to be considered to implement the concept of kanbans in such an environment. We discuss these complications and introduce the concept of a multi-kanban mechanism to cope with this environment. With the help of a simulation model (developed using the ARENA® software [10]), we show that the modified kanban mechanism is in fact an effective tool for a disassembly line setting. An example is presented to illustrate the concept.

LITERATURE REVIEW

Hopp and Spearman [9] describe the basic mechanism of kanban control in one-card and two-card environments. Gupta and Al-Turki [4], [5], [6] and Gupta *et al*. [7] propose the concept of the flexible kanban system (FKS) in various settings involving uncertainties.

Among emerging studies of disassembly, disassembly line has become the subject of recent interest [2]. Korugan and Gupta [11] suggest an adaptive way of implementing kanbans to a single-stage hybrid system. More information on disassembly and product recovery can be found from Brennan *et al*. [1], Gungor and Gupta [3], Gupta and McLean [8], Lee *et al*. [12], Moyer and Gupta [13] and Tang *et al*. [14].

DISASSEMBLY LINE AND ITS COMPLICATIONS

Disassembly line can be described as a series of workstations operating in a sequence to disassemble the end-of-life (EOL) products to meet certain demand for subassemblies and/or components. Depending on the type of EOL, it may enter the disassembly line at any station, not just the first station. Similarly, depending on what is demanded, the demand could occur at any station, not just the last station. The arrival of EOL product and the occurrence of demand are the two crucial differences that set a disassembly line and an assembly line apart. These are also the reason for the disorderly fluctuations in inventory levels in a disassembly line as compared to an assembly line. The inventory control mechanism must therefore be carefully addressed in order to control these fluctuations.

MULTI-KANBAN MECHANISM

The multi-kanban mechanism proposed here employs many different kanbans that correspond to various components and subassemblies in the system. The kanbans are used to control the inventory level of the system and initiate the disassembly process at workstations. Figure 1 illustrates a typical disassembly line.

1. Kanban Types

There are two basic types of kanbans in the system: *component kanbans* and *subassembly kanbans*. A component kanban is attached to a disassembled component that is placed in the component buffer of the workstation where it is disassembled. Similarly, a disassembly kanban is attached to a residual subassembly that is placed in the subassembly buffer of the workstation where it was separated from the component. A component placed in a component buffer can be retrieved by an external demand. When authorized, a subassembly placed in the subassembly buffer is routed for disassembly to the next workstation based on its disassembly sequence.

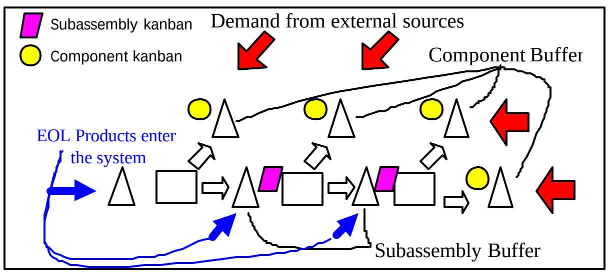

Figure 1. Multi-Kanban in Disassembly

At the first workstation, products arrive only from outside sources. However, at any other workstation *i*, where 1< *i* £ *N*-1, there are two possible types of arrivals. The first type is a subassembly that arrives from an upstream workstation, called *internal subassembly*. There is always a subassembly kanban attached to an internal subassembly. The second type is a product (or subassembly) that arrives from outside sources, called *external subassembly*. There is no kanban attached to an external subassembly. This is also true of the products arriving from external sources to the first workstation. As long as there is an external product or subassembly available at an input buffer, the system will process it first before processing any available internal subassembly. This will avoid unnecessary pulling of an internal subassembly from an upstream workstation. Thus, the number of kanbans attached to internal subassemblies will remain constant throughout the process.

2. Kanban Routing Mechanism

Consider workstation *j*, where 1£ *j* £ *N*-1. When a demand for component *j* arrives at the component buffer of workstation *j*, one unit of component *j* is retrieved and the component kanban *j* attached to it is routed to the most desirable workstation. The procedure for determining the most desirable workstation to route component kanban *j* is given below. (Note that this procedure is not applicable to component kanbans *N*-1 and *N*. In both cases the kanbans are routed to the input buffer of the last workstation).

A component kanban originating from workstation *j* will be routed to a workstation *i*, where 1£ *i* < *j*, or workstation *j* depending on the availability and the desirability of the subassembly that contains component *j*.

Routing component kanban *j* to workstation *i*, where 1£ *i* £ (*j*-1), will result in an immediate separation of component *j* from component *i*. Thus, the only subassembly located at the input buffer of workstation *i* that would be useful is a subassembly that contains only components *i* and *j*. If this type of subassembly exists in the input buffer of workstation *i*, then workstation *i* is qualified. Similarly, if there is at least one subassembly in the input buffer of workstation *j*, then workstation *j* is qualified.

Next, we need to select the most desirable workstation to route component kanban *j* among the qualified ones such that, if chosen, will cause the least amount of extra inventory in the system. Choosing workstation *i* will increase the inventory level of component *i* by an additional unit. Thus, the best workstation *i* is the one that is most starving for its component. By checking the backorder level for demand *i*, we could determine the most starving workstation. If there is a tie, select the most downstream workstation.

Choosing workstation *j* will create a residual subassembly that will be further disassembled at downstream workstations. If workstation *j* is chosen, then a proper subassembly must be chosen to disassemble. For example, if a backorder exists at the component buffer of workstation *k*, where *j*< *k* £ (*N*-1), then, if available, we might try disassembling a subassembly that contains only components *i* and *k*. If more than one workstation qualify as starving workstations, then the one that is most starving among them is chosen. If there is a tie, select the most downstream workstation.

We can now compare the starving levels of workstations *i* and *j*. If the highest starving level of workstation *i* is greater than or equal to the highest starving level of workstation *j* then we will route the component kanban *j* to workstation *i*, otherwise, we will route it to workstation *j*.

Note that whenever an external subassembly is available, it will always be chosen first. Internal subassemblies will only be used when no external subassembly of the desired kind is available. Subassembly kanbans are routed in a fashion similar to component kanbans.

LINE DESCRIPTION AND ASSUMPTIONS

We consider a disassembly line with *N*-1 workstations. The inputs to the line are products (raw materials) with structures that are composed of various combinations of up to *N* components from a list of *N* different components. Any combination of components is possible. However, no more than one unit of any given component can be present in a product. The input location for a product depends on its configuration. The input location for a product is the most upstream workstation that is used to disassemble a component in that product. Only one type of component is disassembled at a given workstation except at the last workstation where component N-1 and component N are separated.

We assume that each component takes the same amount of time to disassemble. At each workstation, there are two types of output buffers. The componennt disassembled at a workstation, s_i, is placed in the *component buffer*, B_i. The rest of the subassembly is routed to the *subassembly buffer*, B'_i. The subassembly buffer becomes the input buffer for the subsequent workstation to further disassemble the subassembly according to its disassembly sequence (Figure 1).

There are multiple sources of demands. A demand can occur at any workstation. The demand at a given workstation is always for the component that is disassembled at that workstation. Regardless of the configuration, products must be disassembled in a predefined sequence from the first component to the last component. When a particular component is demanded, it is retrieved from the output component buffer, B_i, of the workstation where it is disassembled. We allow only single-unit retreival at a time. If there is no component available at the component buffer, the demand waits there in the form of a backorder.

EXPERIMENTATION AND RESULTS

In order to test the proposed mechanism, we experimented the kanban mechanism using ARENA® simulation software [10]. We considered four workstations and EOL products consisting of unto at most five different components, viz., A, B, C, D and E.

We divided the experimentation into two parts. The first part was designed to determine how the kanban mechanism deals with fluctuation in demand and the second part was designed to determine how effective is the kanban mechanism in controlling the inventory level of the system.

Part 1

In this part, we varied the demand rates. We analyzed the response of the system to these changes by observing the backorder queue lengths at the component and subassembly buffers. We learned that as the demand rate increases, both the average component and subassembly backorder queue lengths also increase. As the number of kanbans is increased, the system responds favorably by decreasing the level of both component and subassembly backorder queue lengths.

Part 2

In this part, we varied the demand rates for the component in workstation 3. As we decrease the demand rate for component C, the average inventory level of component C that has kanban attached to it is approximately the same as base level. However, the overflow inventory level increases due to the imbalance in demands. On the other hand, if we increase the demand rate for component C, the kanban-attached inventory level diminishes. The overflow doesn’t exist anymore in this case. We also found out that higher base kanban level result in shorter backorder queues for all components.

REFERENCES

[1] Brennan L., Gupta S. M. and Taleb K. N., "Operations Planning Issues in an Assembly/Disassembly Environment", *International Journal of Operations and Production Management*, Vol. 14, No. 9, 57-67, 1994.

[2] Gungor A. and Gupta S. M., “Disassembly Line in Product Recovery”, *International Journal of Production Research*, Vol. 40, No. 11, 2569-2589, 2002.

[3] Gungor A. and Gupta S. M., “Issues in Environmentally Conscious Manufacturing and Product Recovery: A Survey”, *Computer and Industrial Engineering*, Vol. 36, No. 4, 811-853, 1999.

[4] Gupta S. M. and Al-Turki Y. A. Y., “Adapting Just-In-Time Manufacturing Systems to Preventive Maintenance Interrupts”. *Production Planning and Control*, Vol. 9, No. 4, 349-359, 1998.

[5] Gupta S. M. and Al-Turki Y. A. Y., "An Algorithm to Dynamically Adjust the Number of Kanbans in a Stochastic Processing Times and Variable Demand Environment", *Production Planning and Control*, Vol. 8, No. 2, 133-141, 1997.

[6] Gupta S. M. and Al-Turki Y. A. Y., "The Effect of Sudden Material Handling System Breakdown on the Performance of a JIT System", *International Journal of Production Research*, Vol. 36, No. 7, 1935-1960, 1998.

[7] Gupta S. M., Al-Turki Y. A. Y. and Perry R. F., "Flexible Kanban System", *International Journal of Operations and Production Management*, Vol. 19, No. 10, 1065-1093, 1999.

[8] Gupta S. M. and McLean C. R., "Disassembly of Products", *Computers and Industrial Engineering*, Vol. 31, 225-228, 1996.

[9] Hopp W. J. and Spearman M. L., “Factory Physics”, Second Edition, *McGraw-Hill*, New York, 2001.

[10] Kelton D. W., Sadowski R. P. and Sadowski D. A., “Simulation with Arena®”, WCB, McGraw-Hill, New York, 1998.

[11] Korugan A. and Gupta S. M., “Adaptive Kanban Control Mechanism for A Single-stage Hybrid System”, Proceedings of the 2001 SPIE Conference, No. 4569, 175-181, 2002.

[12] Lee D.-H., Kang J.-G. and Xirouchakis P., “Disassembly Planning and Scheduling: Review and Further Research”, Journal of Engineering Manufacture, Vol. 215, No. B5, 695-709, 2001.

[13] Moyer L. and Gupta S. M., "Environmental Concerns and Recycling/ Disassembly Efforts in the Electronics Industry",

Journal of Electronics Manufacturing, Vol. 7, No. 1, 1-22, 1997.

[14] Tang Y., Zhou M., Zussman E., and Caudill R, "Disassembly Modeling, Planning, and Application: A Review", Proceedings of the 2000 IEEE International Conference on Robotics and Automation, San Francisco, California, April, pp. 2197-2202., 2000.