L R m

Laboratory for Responsible Manufacturing

Bibliographic Information

Moore, K. E., Gungor, A. and Gupta, S. M., "Disassembly Petri Net Generation in the Presence of XOR Precedence Relationships", ***Proceedings of the 1998 IEEE International Conference on Systems, Man and Cybernetics***, La Jolla, California, October 11-14, pp. 13-18, 1998.

Copyright Information

(c) 1998 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other users, including reprinting/ republishing this material for advertising or promotional purposes, creating new collective works for resale or redistribution to servers or lists, or reuse of any copyrighted components of this work in other works.

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***

Disassembly Petri Net Generation in the Presence of XOR Precedence Relationships

K. E. Moore
Information and Decision Systems
ALPHATECH, Inc.
50 Mall Road
Burlington, MA 01803, USA

A. Gungor and S. M. Gupta
Laboratory for Responsible Manufacturing
Dept. of MIME, 334 Snell Engineering
Northeastern University, 360 Huntington Ave.
Boston, MA 02115, USA

ABSTRACT

A disassembly process plan (DPP) is a sequence of disassembly tasks which begins with a product to be disassembled and terminates in a state where all the parts of interest are disconnected. Disassembly process planning is critical for minimizing the resources invested in disassembly and maximizing the level of automation of the disassembly process and the quality of the parts (or materials) recovered. In this paper, we propose an algorithm which automatically generates a disassembly PN (DPN) from a geometrically-based disassembly precedence matrix (DPM). This algorithm can be used to generate DPPs for products which contain simple AND, OR, complex AND/OR, and XOR relationships. The resulting DPN can be analyzed using the reachability tree method to generate all feasible disassembly process plans (DPPs), and cost functions can be used to determine the optimal DPP. An example is used to illustrate the procedure.

1. INTRODUCTION

As a result of regulatory and consumer pressures, there has been an increasing emphasis on environmentally conscious manufacturing (EnviCoM). EnviCoM involves the entire life cycle of products, from conceptual design to final delivery, and ultimately to the end-of-life (EOL) disposal of the products, such that environmental standards and requirements are satisfied. A major element of EOL is product recovery which includes recycling and remanufacturing. Both recycling and remanufacturing involve product disassembly in order to retrieve the desired parts and/or subassemblies.

Disassembly may be defined as a systematic method for separating a product into its constituent parts, components, subassemblies, or other groupings. Disassembly may be partial (some components are not fully disassembled) or complete (the product is fully disassembled). Disassembly process planning is critical in minimizing resources (e.g., time and money) invested in disassembly and maximizing the level of automation of the disassembly process and the quality of the parts (or materials) recovered.

A disassembly process plan is a sequence of disassembly tasks which begins with a product to be disassembled and terminates in a state where all of the parts of interest are disconnected (i.e., it includes partial and complete disassembly). We are interested in generating optimal or near-optimal DPPs, which minimize the cost of disassembly (assuming some level of disassembly is required) or obtain the best cost/benefit ratio for disassembly. In this paper, we present an algorithm that automatically generates feasible DPPs from a geometrically-based DPM. The latter can be generated from a CAD representation of the product.

In the next section, we present a review of the literature on disassembly process planning. In Section 3, we describe our technical approach and present an example to illustrate the methodology. In Section 4, we present the algorithm for generating the DPN for products with XOR precedence relationships. Section 5 summarizes the work presented here.

2. LITERATURE REVIEW

Disassembly process planning is a new area of research and there is a relatively small number of papers. One of the first known papers was published in 1991 and uses a branch-and-bound approach to minimize total disassembly cost [1]. At each stage in the disassembly, the algorithm selects a part to disassemble which has the lowest total disassembly cost. This approach generates the optimal DPP when costs are constant. Several papers utilize AND/OR graphs. For example, [2] generates the AND/OR graph from movement and interference matrices; the AND/OR graph is then analyzed to generate all feasible DPPs. References [3-5] generate disassembly process graphs (DPGs) from AND/OR graphs. The DPG incorporates cost and revenue data, which are used to stop DPG generation when further disassembly is no

0-7803-4778-1/98 $10.00 © 1998 IEEE

longer profitable. The major drawback to this approach is that it is exhaustive. A graph theoretical approach is proposed in [6]. The graph is based on the geometric information for the product and is analyzed to generate all feasible disassembly steps. The resulting tree can be used to identify an efficient, though not necessarily optimal, sequence of steps to disassemble a specific part.

PNs have been shown to be very useful in assembly process planning [7]. To date, however, very little has been done to apply PNs to disassembly. In the first paper to use PNs in disassembly, [8] propose a method for generating PNs from a series of precedence tables, similar to the method used in [9]. The resulting PN is analyzed using the reachability method to generate all feasible DPPs. No optimization is done and the approach is exhaustive.

One of the limitations present in all of the above papers is that the disassembly precedence relationships are limited to simple AND and simple OR relationships; none contain complex AND/OR or XOR relationships. An *AND relationship* exists between components c_1 and c_2 in relation to c_3, if both c_1 and c_2 must be removed prior to c_3. An *OR relationship* exists between parts c_1 and c_2 in relation to c_3, if either c_1 or c_2 must be removed prior to c_3. A *complex AND/OR relationship* exists between parts c_1, c_2, and c_3, in relation to c_4, if c_1 along with either c_2 or c_3 must be removed prior to c_4. An *XOR relationship* exists between parts c_1, c_2, and c_3, in relation to c_4, if c_1 along with c_2 or c_3, but not both, must be removed prior to c_4 (i.e., if c_1 and c_2 are removed, then c_3 must remain until c_4 is removed and, likewise, if c_1 and c_3 are removed, then c_2 must remain until c_4 is removed). [10, 11] are the only papers to address the complex AND/OR relationship. In this paper, we extend the work of [10, 11] to XOR relationships.

APPROACH

In our research, we are developing a method based on PNs to generate near-optimal DPPs. The methodology involves the following steps: 1) Analyze the product to generate a disassembly precedence matrix (DPM) representing the physically-based disassembly constraints [12]; (2) Generate the disassembly PN (DPN) from the DPM; and (3) Generate near optimal DPPs from the DPN. In this paper, we describe the algorithm for generating a near-optimal DPP. We have successfully applied this approach to the case of complex AND/ORs [10, 11].

Throughout this paper, we use the product shown in Fig. 1 to illustrate our approach. The example product consists of seven elements (five parts and two joining elements). As shown, part 1 is attached to the fixture (the shaded base and clamps). The XOR relationship for this product exists between parts 3, 4, and 5. Either part 3 or part 4 (but not both) must be removed prior to part 5; part 5 cannot be left hanging in the air. For this discussion, we only consider movement in two dimensions; however, this approach can be extended to the three-dimensional case without loss of generality. Movement can be in direction d, $d = \{x, -x, y, -y\}$.

Disassembly Precedence Matrix

The DPM represents the geometrically-based precedence relationships between the components of the product. We recognize the following types of precedence relationships: AND, OR, complex AND/OR, and XOR. Due to the nature of the geometric constraints, groups of OR constraints are in the same direction, d. We can now define the DPM, $B = [b_{ij}]$, $i, j = 1, \ldots, k$ (k is the number of parts) as:

$$b_{ij} = \begin{Bmatrix} 1, & \text{part } i \text{ AND precedes part } j \\ d, & \text{part } i \text{ OR precedes part } j \\ -1, & \text{part } i \text{ XOR precedes part } j \\ 0, & \text{otherwise} \end{Bmatrix}. \qquad (1)$$

The DPM can be generated automatically from a CAD representation of the product [12]. The DPM for the product appears in Fig. 2.

Petri Nets

Petri nets (PNs) are a graphical and mathematical technique useful for modeling concurrent, asynchronous, distributed, parallel, nondeterministic, and stochastic systems. PN models can be analyzed to determine both their qualitative and quantitative properties. PNs have recently emerged as a promising approach for modeling manufacturing systems, and have been used for assembly process planning [7]. As discussed above, very little has been done to apply PNs to disassembly.

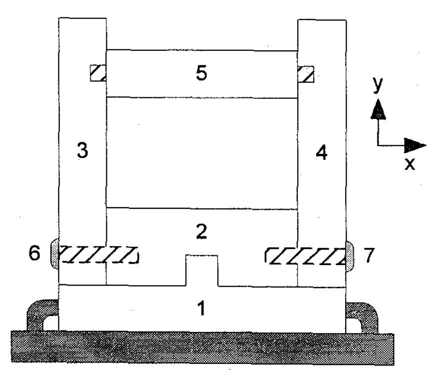

Fig. 1. Sample Product.

$$\begin{array}{c|c}
k & \begin{matrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 \end{matrix} \\
\begin{matrix} 1\\2\\3\\4\\5\\6\\7 \end{matrix} &
\begin{bmatrix}
0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & -1 & 0 & 0 \\
1 & 0 & 0 & 0 & -1 & 0 & 0 \\
1 & 1 & 0 & 0 & 0 & 0 & 0 \\
1 & 1 & 1 & 0 & 0 & 0 & 0 \\
1 & 1 & 0 & 1 & 0 & 0 & 0
\end{bmatrix}
\end{array}$$

Fig. 2. DP for sample product.

A PN is defined as a 5-tuple, $PN = (P, T, A, W, M_0)$, where $P = \{p_i\}$ is a finite set of places, $i = 1,\ldots, m$; $T = \{t_j\}$ is a finite set of transitions, $j = 1,\ldots, n$; $A \subseteq \{P \times T\} \cup \{T \times P\}$ is a set of directed arcs; $W: A \rightarrow \{1, 2,\ldots\}$ is a set of weight function on arcs; $M_0: P \rightarrow \{0, 1,\ldots\}$ is the initial marking; and $P \cap T = \varnothing$ and $T \cup P > \varnothing$. The net structure may be represented compactly by $A = [a_{ij}]$, where

$$a_{ij} = \begin{Bmatrix} -w(i,j), & \text{arc goes from } p_i \text{ to } t_j; \\ w(i,j), & \text{arc goes from } t_j \text{ to } p_i; \\ 0, & \text{otherwise.} \end{Bmatrix}. \tag{2}$$

A marking, M_q, denotes the current state of a PN, after the q^{th} transition firing. A marking changes when a transition *fires*. We define the set functions I of input places and O of output places for a transition t_j, as $I(t_j) = \{p \mid (p, t_j) \subset A\}$ and $O(t_j) = \{p \mid (t, p_j) \subset A\}$, respectively. A transition t_j is enabled in a marking M_q iff $M_q(p) \geq w(p, t_j)\ \forall\ p \in I(t_j)$, where $w(p, t_j)$ is the weight of the arc from p to t_j. When a transition fires, the new marking is defined as $M_{q+1} = \{M_q - w(p, t_j)\ \forall\ p \in I(t_j);\ M_q + w(t_j, p)\ \forall\ p \in O(t_j);\ M_q, \text{otherwise}\}$. We write $M_q[t_j\rangle M_{q+1}$ to say that M_{q+1} is reachable from M_q by firing t_j; $M_q[s\rangle M_r$ means that M_r is reachable from M_q by firing the sequence of transitions specified by s.

GENERATING THE DISASSEMBLY PETRI NET

In this section, we describe the algorithm for automatically generating a DPN from the DPM for products containing XOR precedence relationships. In the following discussion $B_{i\cdot}$ represents row i of B, $B_{\cdot j}$ represents column j of B, and $\mathbf{0}$ is a zero vector In addition, we define the place and transition notation for the DPN as follows: p_b represents the beginning place (the presence of a product to disassemble), p_i represents the preconditions to removing part i, p_f represents the (reduced) preconditions for completion of the disassembly, t_b is a logical transition relating p_b to any part which has no AND precedents, t_i represents the disassembly of part i, and t_f is a logical transition representing the completion of the disassembly. Additional place and transition notation will be introduced as necessary.

Observations on the Disassembly Precedence Matrix

We begin by making several observations about the DPM. These observations are explained below.

(H1) No Precedents: Column j contains only zeros ($B_{\cdot j} = \mathbf{0}$): Component c_j has no precedents; the presence of a product to disassemble is a sufficient condition for the removal of c_j. Hence, p_j is an output of t_b and the sole input to t_j. The related arc weights are unity: $w(p_b, t_b) = w(t_b, p_j) = w(p_j, t_j) = 1\ \forall\ \{j \mid B_{\cdot j} = \mathbf{0}\}$.

(H2) No Antecedents: Row i contains only zeros ($B_{i\cdot} = \mathbf{0}$). The removal of c_i is not precedent to the removal of any other component; however, it is a precondition of complete disassembly. Hence, t_i is an input to p_f, $w(t_i, p_f) = 1$, and $w(p_f, t_f) =$ the number of zero rows in B.

(H3) AND Precedents: $B_{\cdot j}$ contains one or more 1s ($b_{ij} = 1$, for some i). The removal of c_i precedes the removal of c_j. This is an AND precedent, i.e., the removal of all c_i $\{i \mid b_{ij} = 1\}$ is precedent to the removal of c_j. The arc weights are $w(t_i, p_j) = 1$ and $w(p_j, t_j) =$ the number of 1s in $B_{\cdot j}$.

(H4) OR Precedents: $B_{\cdot j}$ contains ds where no two ds are the same ($\forall\ b_{hj}, b_{ij} \in D$, $b_{hj} \neq b_{ij}$). The removal of c_i ($b_{ij} = d$) is OR precedent in direction d to the removal of c_j. At least one c_i ($b_{ij} \in D$) must be removed prior to removing c_j. Let po_j represent the set of OR conditions; $w(t_i, po_j) = 1$. Since only one OR condition must be met to disassemble c_j, $w(po_j, t_j) = 1$. To ensure that only one OR condition fires t_j, let p_j be a place with an input arc from t_b and an output arc to t_j where $w(t_b, p_j) = w(p_j, t_j) = 1$. Since the remaining OR conditions are required for complete disassembly, introduce an arc from po_j to t_f; where $w(po_j, t_f) =$ one less than the total number of OR precedent conditions to c_j.

(H5) AND Within OR Precedents: $B_{\cdot j}$ contains ds where at least two ds are equal ($\exists\ b_{hj}, b_{ij} \in D \mid b_{hj} = b_{ij}$ and $h \neq i$). The removal of a *set* of components $\{c_i \mid b_{ij} = d\}$ is OR precedent in direction d to the removal of c_j; i.e., all c_i's with the same d w.r.t. c_j must be removed to satisfy a single OR condition. This is an AND precedence within an OR precedence group. This case combines (H3) and (H4). Place $pa_{j,d}$ represents the AND conditions within the d^{th} OR precedent group for c_j; a logical transition $ta_{j,d}$ represents completion of the set of AND conditions within the d^{th} OR precedent group for c_j; and $w(t_i, pa_{j,d}) = 1$ and $w(pa_{j,d}, ta_{j,d}) =$ the number of AND conditions within the d^{th} OR precedent group for c_j. From (H4), we include po_j and p_j, where $w(ta_{j,d},$

$po_{j,d}) = w(t_b, p_j) = w(p_j, t_j) = 1$, and $w(po_j, t_f)$ = one less than the total number of OR precedent groups to c_j.

(H6) Complex AND/OR Precedents: $B_{\cdot j}$ contains ds and 1s ($B_{\cdot j} = \{b_{ij} \mid b_{ij} = \{0, 1, d\}$). Removal of the *set* of components $\{c_i \mid b_{ij} = 1\}$ along with *at least one* of the *sets* of components $\{c_h \mid b_{hj} = d\}$ is precedent to the removal of c_j. This represents the complex AND/OR precedence relationship. We simply replace pc_j and its input and output arcs with p_j, where p_j represents the AND precedent conditions for removing c_j. The definition of p_j, its input transition(s), and input and output arc weights are the same as for (H3). (H6) combines (H3), (H4), and (H5).

(H7) XOR Precedents: $B_{\cdot j}$ contains -1s ($B_{\cdot j} = \{b_{ij} \mid b_{ij} = \{0, -1\}$). Removal of one and only one c_i ($b_{ij} = -1$) is precedent to the removal of c_j. Since only one c_i ($b_{ij} = -1$) can be removed prior to c_j, we need to prevent any other c_h ($b_{ij} = -1, i \neq h$) from being removed prior to c_j. Let pc_j represent the control for the XOR precedents; $w(pc_j, t_i) = 1 \ \forall i \ \{b_{ij} = -1\}$. To establish the initial control, we introduce an arc from t_b to pc_j where $w(t_b, pc_j) = 1$; to allow the remaining XOR precedent parts to be disassembled, we introduce an arc from t_b to pc_j where $w(t_f, pc_j)$ = the number of XOR conditions to j less one. Since we do not know which XOR condition will be met first and since the remaining XOR conditions are required for complete disassembly, we introduce px_j and use it in a manner analogous to po; i.e., $w(px_j, t_j) = 1$, $w(t_i, px_j) = 1 \ \forall i \ \{b_{ij} = -1\}$, and $w(px_j, t_f)$ = one less than the total number of XOR precedent conditions to c_j.

Algorithm to Generate the DPN with XORs

Let AG_j be the set of AND precedents to j, O_j be the set of directions for which there exist OR precedence group to j, $OG_{j,d}$ be the OR precedent group to j in direction d, XG_j be the set of components XOR precedent to j, and NZ a vector of binary variables indicating whether B_i contains non-zero entries.

$$nz_i = \begin{cases} 1, & B_{i\cdot} = \mathbf{0} \\ 0, & \text{otherwise} \end{cases}.$$

The algorithm to generate the DPN A appears in Fig. 3. Fig. 4 shows the DPN generated by this algorithm for the sample product.

Step 1: Initialize Variables.

$P = \{p_b, p_i, p_f\}$, $T = \{t_b, t_j, t_f\}$, $i, j = 1$ to k;
$A = (k + 2) \times (k + 2)$ matrix (initial size),
where $a_{ii} = -1$ for $i = 1$ to $k + 2$, $a_{k+1,k+2} = 1$;
$AG_j = O_j = OG_{j,d} = XG_j = \{\varnothing\}, j = 1$ to k, $d \in \{D\}$,
$nz_i = 1$, $i = 1$ to k.

Step 2: Complete T, P, and A.

Step 2.1: Scan $B_{\cdot j}$ for places associated with AND, OR, and XOR precedence groups.

If $b_{ij} = 1$, add i to AG_j, set $nz_i = 0$.

If $b_{ij} = d$, add i to $OG_{j,d}$, add d to O_j, add $po_{j,d}$ to P, set $nz_i = 0$.

If $b_{ij} = -1$, add i to XG_j, add pc_j and px_j to P, set $nz_i = 0$.

Step 2.2: Generate arcs in A for places and transitions associated with $B_{\cdot j}$.

Step 2.2.1: Examine AG_j.

If $|AG_j| = 0$, $a(p_j, t_b) = 1$

If $|AG_j| > 0$, $a(p_j, t_j) = -|AG_j|$,
$a(p_j, t_i) = 1$, for $i \in AG_j$.

Step 2.2.2: Examine $O_{j,d}$ and O_j.

If $|O_j| > 0, a(po_j, t_j) = -1$
$a(po_j, t_f) = -(|O_j| - 1)$.

If $|O_{j,d}| = 1$, for $d \in O_j$,
$a(po_j, t_i) = 1$, for $i \in O_{j,d}$.

If $|O_{j,d}| > 1$, for $d \in O_j$
$T = \{T \cup ta_{j,d}\}$, $P = \{P \cup pa_{j,d} \cup po_j\}$
$a(po_j, ta_{j,d}) = 1$, $a(pa_{j,d}, ta_{j,d}) = -|O_{j,d}|$
$a(pa_{j,d}, t_i) = 1$ for $i \in O_{j,d}$.

Step 2.2.3: Examine XG_j.

If $|XG_j| > 0$, $P = \{P \cup pc_j \cup px_j\}$
$a(pc_j, t_b) = 1$, $a(pc_j, t_f) = (|XG_j| - 1)$
$a(pc_j, t_i) = -1 \ i \in XG_j$, $a(px_j, t_i) = 1 \ i \in XG_j$
$a(px_j, t_j) = -1$, $a(px_j, t_f) = -(|XG_j| - 1)$.

Step 3. Finalize A.

Step 3.1: Scan NZ to generate arcs for parts with no antecedents.

$a(p_f, t_i) = nz_i$

Step 3.2: Sum NZ to generate arcs for final place.

If $\sum_{i=1}^{k} nz_i > 0$, $a(p_f, t_f) = -\sum_{i=1}^{k} nz_i$

If $\sum_{i=1}^{k} nz_i = 0$, $a(p_f, t_b) = 1$

Fig. 3. Algorithm for Generating DPN with XOR.

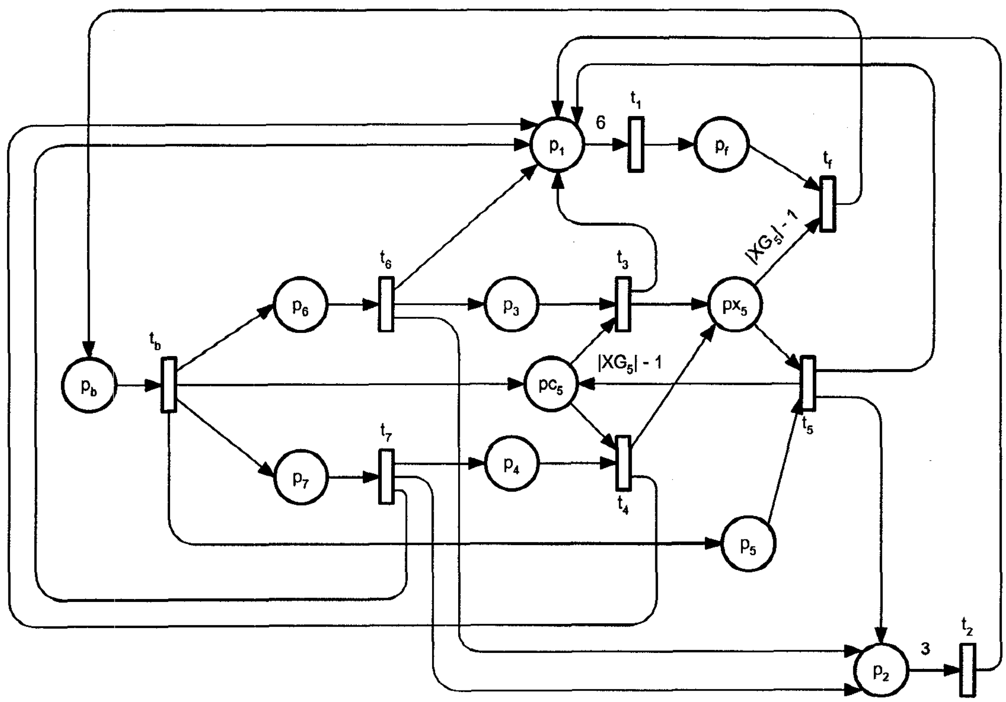

Fig. 4. DPN for Sample Product.

Properties of the DPN

With a small number of restrictions, this algorithm guarantees that the resulting DPN is live, bounded, and reversible. These restrictions include:

- In complex AND/OR cases, a part cannot be OR precedent in more than one direction.
- We do not consider direction in the case of XOR .
- We assume that there are a finite number of parts, that the parts can be removed without destructive disassembly, parts are removed individually, and parts are removed in a single direction.

The proof of these properties, which is too long for inclusion here, is based on the work of [13, 14]. The proof begins with a set of submodels that capture the observations made on the DPN. Each submodel is supplemented by p_b, t_b, p_f, and t_f, resulting in a live, bounded, and reversible DPN submodel. The resulting models are merged via shared paths, and then reduced by deleting redundant paths. The rules for merging and reduction preserve the desired behavior. This is simply a graphical version of the algorithm presented in Fig. 3. Further, the DPN is constructed such that every transition in the net must fire, before the final transition can fire.

The DPN can by analyzed using the reachability tree method [15] to generate feasible DPPs. The algorithm for generating the DPN guarantees that the reachability tree contains only feasible DPPs. Once the DPPs are identified, cost functions can be applied to determine the best DPP.

SUMMARY

In this paper, we presented a methodology based on Petri nets (PNs) which can be used to generate disassembly process plans (DPPs). We presented an algorithm to automatically generate geometric-based disassembly precedence matrix (DPM). The product may contain AND, OR, complex AND/OR, and XOR precedence relationships. To our knowledge, this is the first paper which addresses the XOR case.

The DPN is constructed in such a way as to guarantee that it will generate only feasible DPPs, using the reachability tree method. Once the set of all feasible DPPs is generated, cost functions can be applied to determine the optimal DPP. Since the reachability tree method is *NP*-

complete, for complex products, a heuristic approach can be applied to generate near-optimal DPPs.

REFERENCES

[1] A. K. Subramani and P. Dewhurst, "Automatic generation of product disassembly sequences," *Annals of the CIRP*, vol. 40, pp. 115-118, 1991.

[2] Y.-Q. Lee and S. R. T. Kumara, "Individual and group disassembly sequence generation through freedom and interference spaces," *Journal of Design and Manufacturing*, vol. 2, pp. 143-154, 1992.

[3] Y. Pnueli and E. Zussman, "Evaluating the end-of-life value of a product and improving it by redesign," *International Journal of Production Research*, vol. 35, pp. 921-942, 1997.

[4] E. Zussman, A. Kriwet, and G. Seliger, "Disassembly-oriented assessment methodology to support design for recycling," *Annals of the CIRP*, vol. 43, pp. 9-14, 1994.

[5] A. J. D. Lambert, "Optimal disassembly of complex products," *International Journal of Production Research*, vol. 35, pp. 2509-2523, 1997.

[6] D. Dutta and T. C. Woo, "Algorithm for multiple disassembly and parallel assemblies," *Journal of Engineering for Industry*, vol. 117, pp. 102-109, 1995.

[7] K. E. Moore and S. M. Gupta, "Petri net models of flexible and automated manufacturing systems: a survey," *International Journal of Production Research*, vol. 34, pp. 3001-3035, 1996.

[8] P. Xirouchakis and D. Kiritsis, "A generic Petri net model for disassembly planning processes," *Proceedings TMCE'96*, Budapest, 1996.

[9] D. Kiritsis and M. Porchet, "A generic Petri net model for dynamic process planning and sequence optimization," *Advances in Engineering Software*, vol. 25, pp. 61-71, 1996.

[10] K. E. Moore, A. Gungor, and S. M. Gupta, "A Petri net approach to disassembly process planning," *Computers and Industrial Engineering*, vol. 35, pp. in press, 1998.

[11] K. E. Moore, A. Gungor, and S. M. Gupta, "Disassembly process planning using Petri nets," *Proceedings IEEE International Symposium on Electronics and the Environment*, pp. 88-93, Oak Brook, IL, 1998.

[12] A. Gungor and S. M. Gupta, "Disassembly sequence planning for complete disassembly in product recovery," *Proceedings Northeast Decision Sciences Institute Conference*, pp. 250-252, Boston, MA, 1998.

[13] I. Koh and F. DiCesare, "Modular transformation methods for generalized Petri nets and their application to automated manufacturing systems," *IEEE Transactions on Systems, Man and Cybernetics*, vol. 21, pp. 1512-1522, 1991.

[14] B. H. Krogh and C. L. Beck, "Synthesis of place/transition nets for simulation and control of manufacturing systems," *Proceedings Fourth IFACS/IFORS Symposium on Large Scale Systems: Theory and Applications*, pp. 583-588, Zurich, Switzerland, 1986.

[15] T. Murata, "Petri nets: properties, analysis and applications," *Proceedings of the IEEE*, vol. 77, pp. 541-580, 1989.