Second Asian Symposium on Industrial Automation and Robotics
BITEC, Bangkok, Thailand
May 17-18, 2001

DEVELOPMENT OF DISASSEMBLY SUPPORT SYSTEM FOR MECHANICAL PARTS

Eiji ARAI, Hidefumi WAKAMATSU, Akira TSUMAYA, Keiichi SHIRASE
Dept. of Manufacturing Science, School of Eng., Osaka University
2-1 Yamadaoka, Suita, Osaka 565-0871
JAPAN

Hisashi SASAJIMA
Advanced Architecture and Technologies Research and Development Headquarters
Yamatake Corporation
1-12-1, Kawana, Fujisawa, Kanagawa 251-8522
JAPAN

***Abstract**: Assembly/disassembly planning of mechanical products is one of the important manufacturing activities that must be supported by computers. Furthermore, recently, the design for disassemblability/recyclability becomes more important from the viewpoint of Life Cycle Assessment or Emission-Minimum. Therefore, we develop a system which can generate disassembly sequences and which can be applied to the design considering the disassemblability. First, methods to calculate possible motions of parts and to detect configurations where contact state transition occurs are introduced. Based on them, an algorithm for verifying disassemblability of parts is explained. Next, we propose a method for estimate the total number of feasible disassembly sequences without actually generating them. In order to reduce the number of disassembly sequences, precedence constraints such as the assembly feature are introduced. Finally, we implement a disassembly sequence generation system based on our proposed methods and demonstrate its efficiency.*
***Keywords:** assembly, disassembly, disassemblability, reuse, recycle.*

1. INTRODUCTION

Assembly/disassembly planning of mechanical products is one of the important manufacturing activities that must be supported by computers. Software systems that practically generate mechanical assembly/disassembly sequences must satisfy the following requirements.

- They must guarantee the feasibility of the obtained assembly/disassembly sequences (Arai et al., 1995). In order to satisfy this requirement, they must deal with various evaluation viewpoints such as geometrical interference, the stability of subassemblies, and the functionality of available assembly/disassembly machines (Uchiyama et al., 1995).

- They must reduce the number of obtained sequences. Since possible assembly/disassembly sequences for practical products often become surprisingly large, a great amount of computation may be required to generate the assembly/disassembly sequences (Uchiyama et al., 1994).

Furthermore, recently, not only the design for manufacturability/assemblability (Boothroyd et al., 1994) but also the design for disassemblability/recyclability becomes more important from the viewpoint of the ecology (Jovane et al., 1993)(Harjula et al., 1996)(Krause & Seliger, 1997)(Nishi et al., 1999)(Kasa & Suga, 1999)(Chiodo et al., 1999).
Therefore, we develop a system which can generate disassembly sequences with the feasibility and the efficiency and can be applied to the design considering the ease of disassembly for recycling/reuse.
First, the method for verifying disassemblability of parts is explained based on calculation of possible motions and detection of configurations where contact state transition occures. Next, in order to reduce the number of disassembly sequences, we propose a method for estimate it without actually generating disassembly sequences and introduce some precedence constraints. Finally, we implement a disassembly support system and some examples are shown to demonstrate its efficiency.

2. DISASSEMBLY SEQUENCE GENERATION

2.1 Geometric model and contact state

In this section, we propose a method of disassembly sequence generation. First, we assume that an assembled product is given by the CAD system and parts have rigid bodies. Let A and a be a selected part for disasemblability verification and one of its shape elements, respectively. B and b denote all other parts excepting A and one of their shape elements.
When a product is represented by a polyhedron, the shape element is either a vertex, an edge, or a planar surface (face). We deal with cylindrical surfaces whose two ends are both circles because axial parts with rotational functionality are typical in mechanical products. Their radius, normalized vector of the central axis, and position vector of a point on the axis, are given by the CAD system.
Contact states can be represented by the combination of the shape elements mentioned above, for example, vertex *a* contacts with face *b*. In our study, the contact states including a cylindrical surface are dealt with specifically. We regard only the case satisfying the following conditions as a cylindrical contact as shown in Fig.1:

- Contacting elements are both cylindrical surfaces.
- Their axes are colinear.
- Their radii are the same.

Other contact states including a cylindrical surface are represented as those between polyhedra by polyhedral approximation of the cylindrical surface.

2.2 Generation of possible motions from contact states

The motion of a part is constrained by contact with the other

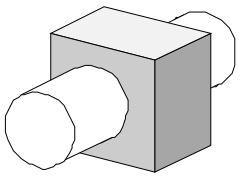


Fig.1. Cylindrical contact

parts. In our study, spatial motion is separated into translational motion and rotational motion. This separation allows analytical calculation of the configuration where contact states change as described in Section 2.4.

Figure 2 shows an example of contact of two parts. In this figure, *a* and *b*, a vertex of a part *A* and a face of *B* respectively, contact with each other. Possible translational directions of *A* are represented by $x \in R^3$ which satisfies the following condition:

$$\boldsymbol{f}^T \boldsymbol{x} \geq 0 \tag{1}$$

where $\boldsymbol{f}$ denotes the outward normal vector of face *b*. Moreover, possible rotational axis directions of *A* are represented by $\boldsymbol{x}$ which satisfies the following inequality:

$$\left[\left(\boldsymbol{v} - \boldsymbol{v}_c\right) \times \boldsymbol{f}\right]^T \boldsymbol{x} \geq 0 \tag{2}$$

where $\times$, $\boldsymbol{v}$, and $\boldsymbol{v}_c$ denote the outer product of vectors, the position vector of *a*, and the foot of the perpendicular from $\boldsymbol{v}$ to the rotational axis, respectively. Please note that $\left(\boldsymbol{v} - \boldsymbol{v}_c\right)$ is normalized.

We can also generate possible motions from other contact states which includes vertex-vertex, edge-face, face-face contact and so on.

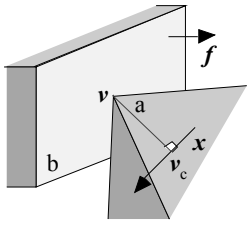


Fig.2. Contact state and possible motion

2.3 Calculation of configuration to transit contact states

Contact states of parts can be changed when one part moves. Such transition of contact states can be classified as follows:

1. The sum of dimensionality of contacting elements decreases, where dimensionalities of a vertex, an edge, and a surface are 0, 1, and 2, respectively. If any element and its bounds are in contact, only the maximum value of the sum of dimensionality is considered. For example, in Fig.3, an edge of *A* and a face of *B*, a vertex of *A* and a face of *B*, an edge of *A* and an edge of *B* are in contact at once. In this case, the edge-face contact has the maximum value of the sum of dimensionality, 1+2=3, and is used. If *A* is translated along the direction $\boldsymbol{x}$ as shown in Fig.3, the edge-face contact turns into the vertex-edge contact whose sum of dimensionality is 1. Thus the sum of dimensionality decreases. We call this configuration where contact state transition occurs contact decreasing configuration (CDC).

2. Contact states are changed by the collision between parts *A* and *B*. We call such a configuration contact increasing configuration (CIC).

When one element of part *A* is in cylindrical contact and the part rotates around its axis, contact state does not change. This rotational motion is not required for removing a part, and can be excluded.

Both CDC and CIC can be calculated if the geometrical shape

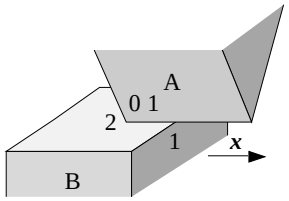


Fig.3. Contact states and dimensionalities

of parts and their removing motions are given.

2.4 Algorithm for verifying disassemblability

Figure 4 shows the algorithm for verifying the disassemblability. First we calculate $\boldsymbol{X}_s$ which denotes a set of possible motions $\boldsymbol{x}_{sk}$ ($k=1, \ldots, K$) calculated from $\boldsymbol{w}_s$ which denotes any configuration. Then, for each *k*, we find $\boldsymbol{x}_{sk}^d$ and $\boldsymbol{x}_{sk}^i$ meaning the CDC and CIC calculated with $\boldsymbol{x}_{sk}$ and $\boldsymbol{w}_s$, and set the nearer configuration of them to $\boldsymbol{w}_{sk}$. Replacing $\boldsymbol{w}_s$ by $\boldsymbol{w}_{sk}$ and repeating this procedure, a tree structure is constructed whose nodes and arcs denote configurations and possible motions of a verified part, respectively.

When $\boldsymbol{w}_{sk}^i$ does not exist for an arbitrary translational motion, we decide that the part is disassemblable. An upper boundary $N_{\max}$ is introduced in order to limit the number of times which a part can change the removing direction. If a state in which a part is disassemblable can not be found as the result of verifying all nodes satisfying the upper boundary, the part is not disassemblable. If a configuration which is exactly the same as that which has already been verified is found, it is excluded, since possible motions calculated from them are the same. In this procedure, the case that a verified part moves along the same path repeatedly does not occur. The current implementation is based on the breadth-first search.

In Fig.4, $\boldsymbol{w}_{\text{init}}$, N_s, $\boldsymbol{W}$, and $\boldsymbol{W'}$ denote an initial configuration of a verified part, the number of times that motion direction changes before reaching the configuration $\boldsymbol{w}_s$, a set of unchecked nodes, and a set of checked nodes, respectively.

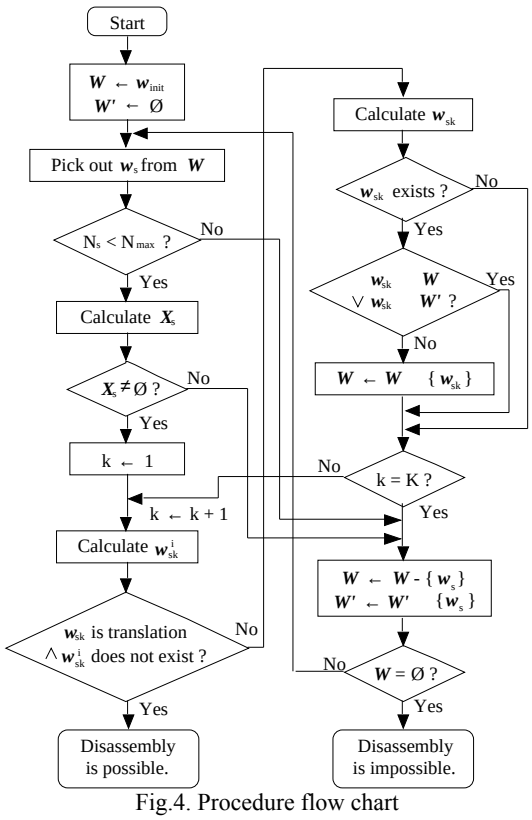


Fig.4. Procedure flow chart

3. ESTIMATION OF DISASSEMBLY SEQUENCE NUMBER

As mentioned in the previous section, We can verify on a computer whether one part of a product can be disassembled or not. By repeating this verification for all parts, we can generate possible assembly sequences. But if the number of parts becomes larger, a great amount of computation may be required to generate assembly sequences because the number of them also becomes larger. Therefore, we must eliminate inappropriate disassembly sequences by giving some precedence constraints in advance. In order to utilize for this elimination, we propose a method for estimating the number of disassembly sequences. It allows designers to decide whether other constraints should be given.

Precedence constraints for disassembly can be represented by a binomial relation between two parts. For example,

$$(p_i, p_j) \tag{3}$$

means that part p_i must be removed before part p_j.

The assumption that directions for removing each part can be given, allows us to represent the precedence constraints with only logical-and connections, without using logical-or. They can be also expressed by either an adjacency matrix or a directed graph. For convenience, we classify such graphs into as follows:

(a) out-tree graph
(b) in-tree graph
(c) disconnected graph consisting of out-tree graphs and in-tree graphs
(d) others

First, we consider only precedence constraints that can be represented by a type(a) graph after they are transformed into a non-transitive graph as shown in Fig.5(b). We transform the non-transitive type(a) graph into a transitive graph as shown in Fig.5(a). The graph shown in Fig.6(a) is a subgraph of the transitive graph as shown in Fig.5(b) where node p_1 is a initial node of arcs whose terminal nodes correspond to all other parts. It means that precedence constraints for disassembly between p_1 and all other parts, the number of them is r_1, are determined, whereas precedence constraints between other parts except part p_1 are not determined. Hence, the number of disassembly sequences from the graph as shown in Fig.6(a) is

$$\frac{n!\, r_1!}{(r_1+1)!} \tag{4}$$

where n denotes the total number of parts. Similarly, as shown in Fig.6(b), p_2 is the initial node of arcs whose terminal nodes are p_5 and p_6. This implies that sequences between p_2 and the two parts are determined, whereas sequence between the two parts is not determined. The number of disassembly sequences becomes

$$\frac{n!\, r_1!\, r_2!}{(r_1+1)!\,(r_2+1)!}. \tag{5}$$

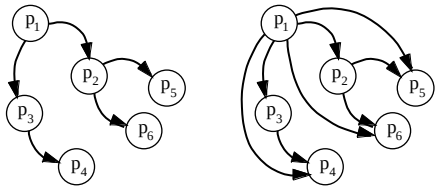
Repeating this procedure for all parts, we

$$\frac{n!\, r_1!\, r_2! \cdots r_n!}{(r_1+1)!(r_2+1)!\cdots(r_n+1)!} = \frac{n!}{(r_1+1)\,(r_2+1)\cdots(r_n+1)}. \tag{6}$$

r_j is the out degree of each node and it is equal to the sum of components of the jth column of the matrix corresponding to the directed graph. The transitive graph as shown in Fig.5(b) can be represented by a matrix as shown in Fig.7(a). The number of feasible sequences calculated from the matrix is

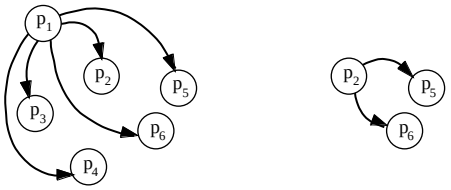
$$\frac{6!}{6\ 3\ 2\ 1\ 1\ 1} = 20. \tag{7}$$

Figure 7(b) shows the feasible sequences satisfying the matrix. These properties can be readily extended to precedence constraints represented by either type(b) or type(c) graphs.



(a) (b)

Fig.5. Non-transitive and transitive graphs



(a) (b)

Fig.6. Calculating of disassembly sequence number for precedence constraints represented by out-tree

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$\begin{array}{cccccc} 5 & 2 & 1 & 0 & 0 & 0 \\ + & + & + & + & + & + \\ 1 & 1 & 1 & 1 & 1 & 1 \end{array}$$

(a)

No.	sequence
1	$p_4\ p_3\ p_5\ p_6\ p_2\ p_1$
2	$p_4\ p_3\ p_6\ p_5\ p_2\ p_1$
3	$p_4\ p_5\ p_3\ p_6\ p_2\ p_1$
4	$p_4\ p_5\ p_6\ p_2\ p_3\ p_1$
5	$p_4\ p_5\ p_6\ p_3\ p_2\ p_1$
6	$p_4\ p_6\ p_3\ p_5\ p_2\ p_1$
7	$p_4\ p_6\ p_5\ p_2\ p_3\ p_1$
8	$p_4\ p_6\ p_5\ p_3\ p_2\ p_1$
9	$p_5\ p_4\ p_3\ p_6\ p_2\ p_1$
10	$p_5\ p_4\ p_6\ p_2\ p_3\ p_1$
11	$p_5\ p_4\ p_6\ p_3\ p_2\ p_1$
12	$p_5\ p_6\ p_2\ p_4\ p_3\ p_1$
13	$p_5\ p_6\ p_4\ p_2\ p_3\ p_1$
14	$p_5\ p_6\ p_4\ p_3\ p_2\ p_1$
15	$p_6\ p_4\ p_3\ p_5\ p_2\ p_1$
16	$p_6\ p_4\ p_5\ p_2\ p_3\ p_1$
17	$p_6\ p_4\ p_5\ p_3\ p_2\ p_1$
18	$p_6\ p_5\ p_2\ p_4\ p_3\ p_1$
19	$p_6\ p_5\ p_4\ p_2\ p_3\ p_1$
20	$p_6\ p_5\ p_4\ p_3\ p_2\ p_1$

(b)

Fig.7. Calculating of disassembly sequence number for graph as shown in Fig.5(a)

At present, we do not know how to calculate the number of the sequences from type(d) graph. We solve this problem by calculating the upper and the lower boundaries of the sequences.

The number given by eq.(6) is smaller than the actual number for type(d) graphs. We call this smaller number the lower boundary.

We calculate the upper boundary by removing several arcs from the type(d) graph, which transforms the graph into another type.

Thus, we can estimate the upper and the lower boundaries of the number of disassembly sequences without calculation of possible motions or geometrical configurations of parts.

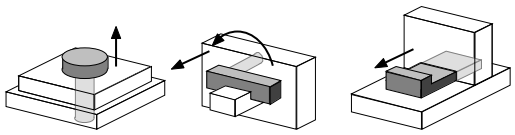
4. REDUCTION OF DISASSEMBLY SEQUENCE NUMBER

In order to reduce the number of disassembly sequences, we define three operations as follows:

(1) introduction of the assembly feature

(2) consideration of removing directions for disassembly
(3) grouping parts

First, we introduce the assembly feature. For example, bolts have the function to fix one part to another part. So it is obvious that they must be removed before these two parts for disassembly. Furthermore, the direction of motion for their removing is predetermined. We call such a set of parts whose disassembly sequence can be predetermined by considering their function the assembly feature. If a designer adds some information with respect to the assembly feature to a product, the number of disassembly sequences can be reduced. In our study, only three types of the assembly feature as shown in Fig.8 are prepared. They can be applied to a bolt, a kind of plug, and a key, respectively.



(a) bolt-type (b) plug-type (c) key-type
Fig.8. Examples of assembly feature

Next, precedence constraints between removing directions for disassembly are considered. In general, the assembly/disassembly of parts whose directions for assembly/disassembly are the same precedes that of parts whose directions are different. Therefore, precedence constraints with respect to removing directions can be given.
Finally, we can reduce the number of disassembly sequences by grouping parts. A set of parts g can be grouped if directions for disassembly of all grouped parts are the same and no parts satisfy both of the conditions below:

(a) p_i is not an element of g
(b) p_i corresponds to an intermediate node on the path between any two parts which are both elements of g.

By applying above three operations, we can add some precedence constraints for disassembly to a product. Increase of the number of precedence constraints leads to decrease of the number of disassembly sequences. For example, a product which is added 8 precedence constraints as shown in Fig.5(b) has 20 disassembly sequences, while that which is added 5 constraints as shown in Fig.6(a) has 120 sequences.
Thus, we can reduce the number of disassembly sequences by adding precedence constraints to a product. When it becomes sufficiently small, we can calculate actual disassembly sequences using the method explained in Section 2.

5. EXAMPLE OF DISASSEMBLY

We implemented a disassembly support system on a UNIX workstation in order to demonstrate the efficiency of our proposed methods. In this section, some case studies of

5.1 Disassembly sequence generation for all parts

We apply our developed system to a practical mechanical product consisting of 12 parts as shown in Fig.9. First, we generate precedence constraints from the viewpoint of geometric interference. Then the system calculates the number of disassembly sequences from generated precedence constraints. The number N becomes

$$1.901 \times 10^6 < N < 4.435 \times 10^6. \quad (8)$$

Next, we consider the assembly feature in order to reduce the number N. Part 9, 10, 11, and 12 have bolt-type assembly feature and part 5 and 6 have plug-type assembly feature. By adding precedence constraints with respect to these features, The number becomes

$$3.802 \times 10^5 < N < 2.218 \times 10^6. \quad (9)$$

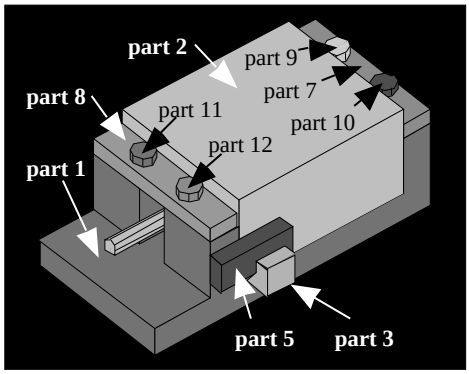
As the number is still large, we consider removing directions. In this case, we assume that disassembly of parts whose translational direction for removing is parallel to z-axis precedes that of any other parts. Then, the number becomes

$$8{,}870 < N < 110{,}880. \quad (10)$$

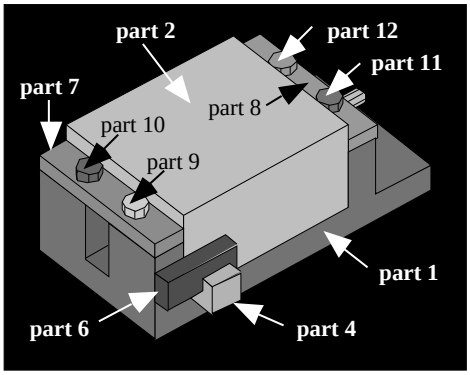
Finally, we group some parts. In this case, part 7, 8, 9, 10, 11, and 12 can be grouped and they are represented by g_1. Then, the number becomes

$$13 < N < 120. \quad (11)$$

This is sufficiently small for generating disassembly sequences. Consequently, 28 sequences are generated as shown in Table 1. After that, the system can calculate disassembly sequences of grouped parts.



(a) Front view



(b) Rear view
Fig.9. Example of product to disassemble

5.2 Disassembly sequence generation for particular part

Our system can also generate disassembly sequences in order to disassemble a particular part. By considering disassembly of one part, we can classify parts into three types as follows:

- type-I : a set of parts which must be removed before the part is disassembled
- type-II : a set of parts which can not be disassembled until the part is removed
- type-III : a set of parts which are neither included in type-I nor type-II.

When one part to disassemble is designated, the system checks whether it can be disassembled or not. If it can not be disassembled, type-I parts with respect to the designated part are searched and their disassemblabilities are verified one by one. If one type-I part can be disassembled, the system removes it and verifies disassemblability of the designated part again. These operations are repeated until the designated part can be disassembled. If type-I parts can not be disassembled, the system searches parts which must be removed before these parts are disassembled, that is, new type-I parts with respect to

Table 1. Result of disassembly sequence generation

No.	sequence
1	p_6 g_1 p_2 p_3 p_4 p_5 p_1
2	p_6 g_1 p_2 p_3 p_5 p_4 p_1
3	p_6 g_1 p_2 p_4 p_3 p_5 p_1
4	p_6 g_1 p_2 p_4 p_5 p_3 p_1
5	p_6 g_1 p_2 p_5 p_3 p_4 p_1
6	p_6 g_1 p_2 p_5 p_4 p_3 p_1
7	p_6 g_1 p_5 p_2 p_3 p_4 p_1
8	p_6 g_1 p_5 p_2 p_4 p_3 p_1
9	g_1 p_2 p_3 p_4 p_6 p_5 p_1
10	g_1 p_2 p_3 p_6 p_4 p_5 p_1
11	g_1 p_2 p_3 p_6 p_5 p_4 p_1
12	g_1 p_2 p_4 p_3 p_6 p_5 p_1
13	g_1 p_2 p_4 p_6 p_3 p_5 p_1
14	g_1 p_2 p_4 p_6 p_5 p_3 p_1
15	g_1 p_2 p_6 p_3 p_4 p_5 p_1
16	g_1 p_2 p_6 p_3 p_5 p_4 p_1
17	g_1 p_2 p_6 p_4 p_3 p_5 p_1
18	g_1 p_2 p_6 p_4 p_5 p_3 p_1
19	g_1 p_2 p_6 p_5 p_3 p_4 p_1
20	g_1 p_2 p_6 p_5 p_4 p_3 p_1
21	g_1 p_6 p_2 p_3 p_4 p_5 p_1
22	g_1 p_6 p_2 p_3 p_5 p_4 p_1
23	g_1 p_6 p_2 p_4 p_3 p_5 p_1
24	g_1 p_6 p_2 p_4 p_5 p_3 p_1
25	g_1 p_6 p_2 p_5 p_3 p_4 p_1
26	g_1 p_6 p_2 p_5 p_4 p_3 p_1
27	g_1 p_6 p_5 p_2 p_3 p_4 p_1
28	g_1 p_6 p_5 p_2 p_4 p_3 p_1

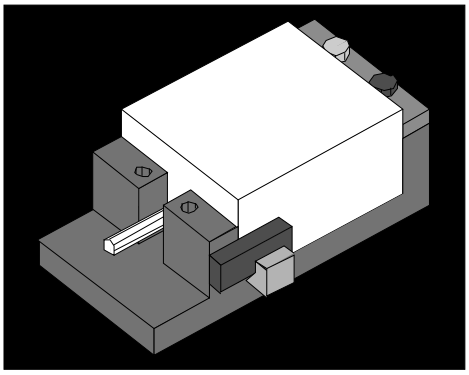
type-I parts, and continues disassemblability verification. When any type-I parts do not exist, the system requires the operator to select one type-III part in order to verify its disassemblability. If any type-III parts also do not exist, it is found that the designated part can be disassembled.

Figure 10(a) shows disassembly sequences of the product as shown in Fig.9 in order to disassemble part 2 which is emphasized with white color in Fig.10(b). Thus, disassembly sequences for not only one part but also a set of parts can be generated.

By using the developed system, designers/process planners can verify disassemblability of the whole parts or particular parts of a product. Furthermore, designers can check whether it is easy to disassemble the product or not when its maintenance, repair, recycling, and disposal are considered and they can improve their design if necessary.

No.	sequence
1	p_{11} p_{12} p_8
2	p_{12} p_{11} p_8

(a)



(b)

Fig.10. Example of designated part disassembly

6. CONCLUSION

A system which can generate disassembly sequences of mechanical parts was developed.

First, methods to calculate possible motions of parts and to detect configurations where contact state transition occurs were presented. Based on them, an algorithm for verifying disassemblability of parts was explained. Next, a method to estimate the total number of feasible disassembly sequences without actually generating them was proposed. In order to reduce the number of disassembly sequences, precedence constraints such as the assembly feature were introduced. Finally, a disassembly sequence generation system was implemented and its efficiency was demonstrated.

We expect that the developed system will be useful for not only the design verification or the assembly planning but also the design evaluation considering recycling/reuse.

7. ACKNOWLEDGEMENT

This study has been supported in part by a Grant-in-Aid for Scientific Research of Japan Society for the Promotion of Science No.11450057 in 1999.

8. REFERENCES

Arai E., Uchiyama N., Igoshi M. (1995). "Disassembly Path Generation to Verify the Assemblability of Mechanical Products", JSME Int. Journal, Series C, Vol.38, No.4, pp805-810.

Boothroyd G., Dewhurst P., Knight W. (1994). "Product Design for Manufacture and Assembly", Marcel Dekker.

Chiodo J.D., Billett E.H., D.J.Harrison D.J. (1999). "Preliminary Investigations of Active Disassembly Using Shape Memory Polymers", Proc. of 1st International Symposium on Environmentally Conscious Design and Inverse Manufacturing, pp590-596.

Harjula T., Knight W.A., Boothroyd G. (1996). "Design for Disassembly and the Environment", Annals of the CIRP, Vol.45-1, pp109-114.

Jovane F., Armillotta A., Feldmann K. (1993). "A Key Issue in Product Life Cycle: Disassembly", Annals of the CIRP, Vol.42-2, pp651-658.

Kasa D., Suga T. (1999). "Active Disassembly of Bonded Wafers", Proc. of 1st International Symposium on Environmentally Conscious Design and Inverse Manufacturing, pp588-589.

Krause F.L., Seliger G. (1997). "Life Cycle Networks", Chapman & Hall.

Nishi T. *et al.* (1999). "Study on TV Recyclability", Proc. of 1st International Symposium on Environmentally Conscious Design and Inverse Manufacturing, pp278-280.

Uchiyama N., Arai E., Igoshi M. (1995). "Testing the stability of Subassemblies Using Geometric Model for Mechanical Assembly Planning (Japanese) ", Trans. of JSME, Series C, Vol.61, No.581, pp324-329.

Uchiyama N., Arai E., Igoshi M. (1994). "Generation of Mechanical Assembly Sequences Considering Different Evaluation Viewpoints", Advancement of Intelligent Production, pp701-706.