Development of Disassembly Support System for Mechanical Parts and Its Application to Design Considering Reuse/Recycle

Hidefumi Wakamatsu, Akira Tsumaya, Keiichi Shirase, and Eiji Arai
Dept. of Manufacturing Science
Graduate School of Eng., Osaka University
2-1 Yamadaoka, Suita, Osaka 565-0871, Japan
{wakamatu, tsumaya, shirase, arai}@mapse.eng.osaka-u.ac.jp

Abstract

Recently, the design for disassemblability/ recyclability becomes more important from the viewpoint of Life Cycle Assessment or Emission-Minimum. Therefore, a disassembly support system which can be applied to the design considering reuse/recycle is developed. First, an algorithm for verifying disassemblability of parts is briefly explained. Next, a method for estimate the total number of feasible disassembly sequences without actually generating them is proposed. Furthermore, in order to reduce the number of disassembly sequences, precedence constraints such as the assembly feature are introduced. Finally, the efficiency of our developed disassembly sequence generation system is demonstrated and it is shown that the design can be evaluated from the viewpoint of disassemblability/recyclability by using our system.

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. 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.
- They must reduce the number of obtained sequences because a great amount of computation may be required to generate assembly/disassembly sequences.

Furthermore, recently, not only the design for manufacturability/assemblability but also the design for disassemblability/recyclability becomes more important from the viewpoint of the ecology[1][2].

Therefore, we develop a system which can generate disassembly sequences with the feasibility and the efficiency and which can be applied to the design considering the ease of disassembly for reuse/recycle.

Firstly, the method for verifying disassemblability of parts is explained based on calculation of possible motions and detection of configurations where contact state transition occurs. Secondly, 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. Thirdly, we implement a disassembly sequence generation system based on our proposed methods and demonstrate its efficiency. Finally, we show that our developed system can be applied to not only the design verification or the assembly planning but also the design considering reuse/recycle.

2. Disassemblability verification

In this section, we briefly explain a method of disassemblability verification. In detail, see [3]. First, we assume that an assembled product is given by the CAD system and parts have rigid bodies. When a product is represented by a polyhedron, the shape element is either a vertex, an edge, or a planar surface. Because axial parts with rotational functionality are typical in mechanical products, we deal with cylindrical surfaces whose two ends are both circles. Then, contact states can be represented as the combination of shape elements such as vertices, edges, and/or faces. In the case of contact between an axial part and a holed part, it is represented as the combination of a convex cylindrical surface and a concave cylindrical surface.

The motion of a part is constrained by contact with the other parts. We can generate possible motions of a part to disassemble from its contact state. 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.

When one part moves, contact states of parts can be changed. Such transition of contact states can be classified into two types: contact decreasing configuration (CDC) and contact increasing configuration (CIC). The former means the configuration in which one part is detached from another part, and the latter means the configuration in which it collides with another. Both CDC and CIC can be calculated if the geometrical shape of parts and their removing motions are given. Then, the algorithm for verifying disassemblability of one part is as follows: First, we calculate a set of possible motions from the current configuration. Then, for each motion, we find CDC and CIC, and set the nearer configuration of them to the next configuration. After that, we generate next possible motions from the next configuration. By repeating this procedure, a tree structure is constructed whose nodes and arcs denote configurations and possible motions of a verified part, respectively. When the next configuration which is CDC does not exist for an arbitrary translational motion, we decide that the part can be disassembled.

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 this section, we propose a method for estimating the number of disassembly sequences in order to utilize it for the elimination. It allows designers to decide whether other constraints should be given or not. Precedence constraints for disassembly can be represented by a directed graph. For convenience, we classify such graphs into as follows:

type(a): out-tree graph

type(b): in-tree graph

type(c): disconnected graph consisting of out-tree graphs and in-tree graphs

type(d): others.

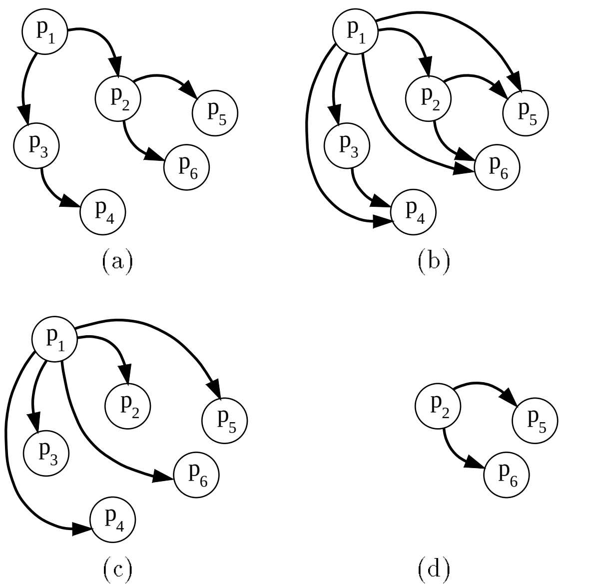


Figure 1. Procedure to calculate disassembly sequence number

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 Figure 1(a). We can transform the non-transitive type(a) graph into a transitive graph as shown in Figure 1(b). The graph shown in Figure 1(c) is a subgraph of the transitive graph as shown in Figure 1(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 shown in Figure 1(c) is

$$\frac{n! \cdot r_1!}{(r_1+1)!} \qquad (1)$$

where n denotes the total number of parts. Similarly, as shown in Figure 1(d), 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

$$\frac{n! \cdot r_1! \cdot r_2!}{(r_1+1)! \cdot (r_2+1)!}. \quad (2)$$

Repeating this procedure for all parts, we have

$$\frac{n! \cdot r_1! \cdots r_n!}{(r_1+1)! \cdots (r_n+1)!} = \frac{n!}{(r_1+1) \cdots (r_n+1)}. \quad (3)$$

In the case as shown in Figure 1(a), the number of feasible sequences is

$$\frac{6!}{6 \cdot 3 \cdot 2 \cdot 1 \cdot 1 \cdot 1} = 20. \quad (4)$$

These properties can be readily extended to precedence constraints represented by either type(b) or type(c) graphs. At present, we do not know how to calculate the number of the sequences from type(d) graph. however, we can solve this problem by calculating the upper and the lower boundaries of the sequences. The number given by eq.(3) 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

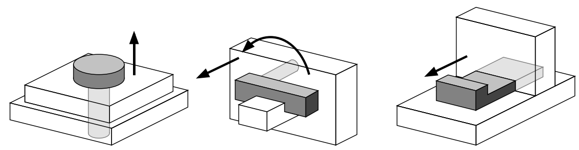
We can estimate the number of disassembly sequences as mentioned in the previous section. If it is too large to calculate disassembly sequences actually, we should add some precedence constraints to a product. Increase of the number of precedence constraints leads to decrease of the number of disassembly sequences. For example, a product which is added 8 precedence constraints as shown in Figure 1(b) has 20 disassembly sequences, while a product which is added only 5 constraints as shown in Figure 1(c) has 120 sequences. Therefore, in order to reduce the number of disassembly sequences, we define three operations which result in addition of some precedence constraints as follows:

(1) introduction of 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. Therefore, 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 assembly feature. If a designer adds some information with respect to assembly feature to a product, the number of disassembly sequences can be reduced. In our study, only three types of assembly feature as shown in Figure 2 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

Figure 2. Examples of assembly feature

Next, precedence constraints between removing directions for disassembly are considered. In general, assembly/disassembly of parts whose directions for assembly/disassembly are 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 same and no parts satisfy the following condition : p_i, which is not an element of g, 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 and it makes the number of disassembly sequences decrease. 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 based on our proposed methods. In this section, some case studies of disassembly sequence generation by our developed system are shown in order to demonstrate the efficiency of our proposed methods.

We apply our system to a practical mechanical product consisting of 12 parts as shown in Figure 3. First, we add precedence constraints from the viewpoint of geometric interference. Then, the system calculates the number of disassembly sequences from such precedence constraints. The number N becomes

$$N = 9.979 \times 10^6 \tag{5}$$

because the directed graph is represented as a out-tree graph. 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

$$N = 1.108 \times 10^6. \tag{6}$$

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

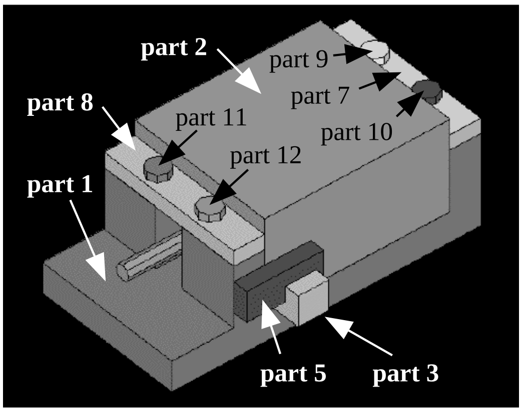
$$44,352 < N < 221,760. \tag{7}$$

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

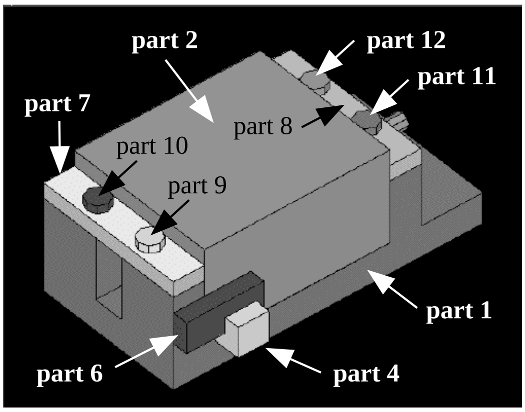
$$40 < N < 120. \tag{8}$$

This is sufficiently small for generating disassembly sequences. Consequently, 60 sequences are generated as shown in Table 1. After that, the system can calculate disassembly sequences of grouped parts.

Our system can also generate disassembly sequences in order to disassemble a particular part. Table 2 shows disassembly sequences of the product shown in Figure 3 in order to disassemble part 2 which is emphasized with white color in Figure 4. 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 the disassemblability of the whole parts or particular parts of a product.



(a) front view



(b) rear view

Figure 3. Example of product for disassembly

6. Disassembly considering reuse/recycle

In this section, we apply our developed system to the design considering reuse/recycle.

Evaluating disassembly sequences of products is important to consider their reuse/recycle. Evaluation viewpoints for reuse/recycle influence disassembly sequences. For example, if the value of parts is considered, expensive parts should be disassembled individually and reused/recycled but cheap parts may be disposed in a group. If the life span of parts is considered, long-lived parts may be reused as parts but short-lived parts should be melted and be recycled as materials. If the kind of material is considered, some parts can be melted together and others should not. Thus, whether some parts can be kept in a group or must be disassembled individually depends on viewpoints for reuse/recycle. On the other hand, disassembly sequences of products depend on their design. Therefore, disassembly sequences for reuse/recycle should be evaluated in design process. By applying our developed system, designers can evaluate them and improve their design if necessary.

In our study, we assume that the disassembly sequence for reuse/recycle is decided by considering the following evaluation viewpoints:

(a) the kind of each part's material

(b) the life span of each part

(c) whether each part can/should be melted or not

Table 1. Result of disassembly sequence generation for all parts

No.	sequence	No.	sequence
1	$p_5\ p_6\ g_1\ p_2\ p_3\ p_4\ p_1$	31	$g_1\ p_2\ p_4\ p_6\ p_3\ p_5\ p_1$
2	$p_5\ p_6\ g_1\ p_2\ p_4\ p_3\ p_1$	32	$g_1\ p_2\ p_4\ p_6\ p_5\ p_3\ p_1$
3	$p_5\ g_1\ p_2\ p_3\ p_4\ p_6\ p_1$	33	$g_1\ p_2\ p_5\ p_3\ p_4\ p_6\ p_1$
4	$p_5\ g_1\ p_2\ p_3\ p_6\ p_4\ p_1$	34	$g_1\ p_2\ p_5\ p_3\ p_6\ p_4\ p_1$
5	$p_5\ g_1\ p_2\ p_4\ p_3\ p_6\ p_1$	35	$g_1\ p_2\ p_5\ p_4\ p_3\ p_6\ p_1$
6	$p_5\ g_1\ p_2\ p_4\ p_6\ p_3\ p_1$	36	$g_1\ p_2\ p_5\ p_4\ p_6\ p_3\ p_1$
7	$p_5\ g_1\ p_2\ p_6\ p_3\ p_4\ p_1$	37	$g_1\ p_2\ p_5\ p_6\ p_3\ p_4\ p_1$
8	$p_5\ g_1\ p_2\ p_6\ p_4\ p_3\ p_1$	38	$g_1\ p_2\ p_5\ p_6\ p_4\ p_3\ p_1$
9	$p_5\ g_1\ p_6\ p_2\ p_3\ p_4\ p_1$	39	$g_1\ p_2\ p_6\ p_3\ p_4\ p_5\ p_1$
10	$p_5\ g_1\ p_6\ p_2\ p_4\ p_3\ p_1$	40	$g_1\ p_2\ p_6\ p_3\ p_5\ p_4\ p_1$
11	$p_6\ p_5\ g_1\ p_2\ p_3\ p_4\ p_1$	41	$g_1\ p_2\ p_6\ p_4\ p_3\ p_5\ p_1$
12	$p_6\ p_5\ g_1\ p_2\ p_4\ p_3\ p_1$	42	$g_1\ p_2\ p_6\ p_4\ p_5\ p_3\ p_1$
13	$p_6\ g_1\ p_2\ p_3\ p_4\ p_5\ p_1$	43	$g_1\ p_2\ p_6\ p_5\ p_3\ p_4\ p_1$
14	$p_6\ g_1\ p_2\ p_3\ p_5\ p_4\ p_1$	44	$g_1\ p_2\ p_6\ p_5\ p_4\ p_3\ p_1$
15	$p_6\ g_1\ p_2\ p_4\ p_3\ p_5\ p_1$	45	$g_1\ p_5\ p_2\ p_3\ p_4\ p_6\ p_1$
16	$p_6\ g_1\ p_2\ p_4\ p_5\ p_3\ p_1$	46	$g_1\ p_5\ p_2\ p_3\ p_6\ p_4\ p_1$
17	$p_6\ g_1\ p_2\ p_5\ p_3\ p_4\ p_1$	47	$g_1\ p_5\ p_2\ p_4\ p_3\ p_6\ p_1$
18	$p_6\ g_1\ p_2\ p_5\ p_4\ p_3\ p_1$	48	$g_1\ p_5\ p_2\ p_4\ p_6\ p_3\ p_1$
19	$p_6\ g_1\ p_5\ p_2\ p_3\ p_4\ p_1$	49	$g_1\ p_5\ p_2\ p_6\ p_3\ p_4\ p_1$
20	$p_6\ g_1\ p_5\ p_2\ p_4\ p_3\ p_1$	50	$g_1\ p_5\ p_2\ p_6\ p_4\ p_3\ p_1$
21	$g_1\ p_2\ p_3\ p_4\ p_5\ p_6\ p_1$	51	$g_1\ p_5\ p_6\ p_2\ p_3\ p_4\ p_1$
22	$g_1\ p_2\ p_3\ p_4\ p_6\ p_5\ p_1$	52	$g_1\ p_5\ p_6\ p_2\ p_4\ p_3\ p_1$
23	$g_1\ p_2\ p_3\ p_5\ p_4\ p_6\ p_1$	53	$g_1\ p_6\ p_2\ p_3\ p_4\ p_5\ p_1$
24	$g_1\ p_2\ p_3\ p_5\ p_6\ p_4\ p_1$	54	$g_1\ p_6\ p_2\ p_3\ p_5\ p_4\ p_1$
25	$g_1\ p_2\ p_3\ p_6\ p_4\ p_5\ p_1$	55	$g_1\ p_6\ p_2\ p_4\ p_3\ p_5\ p_1$
26	$g_1\ p_2\ p_3\ p_6\ p_5\ p_4\ p_1$	56	$g_1\ p_6\ p_2\ p_4\ p_5\ p_3\ p_1$
27	$g_1\ p_2\ p_4\ p_3\ p_5\ p_6\ p_1$	57	$g_1\ p_6\ p_2\ p_5\ p_3\ p_4\ p_1$
28	$g_1\ p_2\ p_4\ p_3\ p_6\ p_5\ p_1$	58	$g_1\ p_6\ p_2\ p_5\ p_4\ p_3\ p_1$
29	$g_1\ p_2\ p_4\ p_5\ p_3\ p_6\ p_1$	59	$g_1\ p_6\ p_5\ p_2\ p_3\ p_4\ p_1$
30	$g_1\ p_2\ p_4\ p_5\ p_6\ p_3\ p_1$	60	$g_1\ p_6\ p_5\ p_2\ p_4\ p_3\ p_1$

(d) whether each part can/should be clashed or not

(e) whether each part is expensive or not.

Then, we can make a matrix as shown in Table 3. This example is very simple but we can also make a database with respect to the kind of possible treatments for reuse/recycle of parts.

Let us assume that a designer thinks only expensive parts should be disassembled individually for reuse/recycle. Then, part 3 and 4 can be removed without disassembly of any other parts, however, it is found that three parts must be disassembled in order to remove part 2 as shown in Figure 4 by applying our developed system. If disassembly of these three parts is too difficult or costs too much, the designer can improve their design so that part 2 can be disassembled more easily.

Table 2. Result of disassembly sequence generation for removal of part 2

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

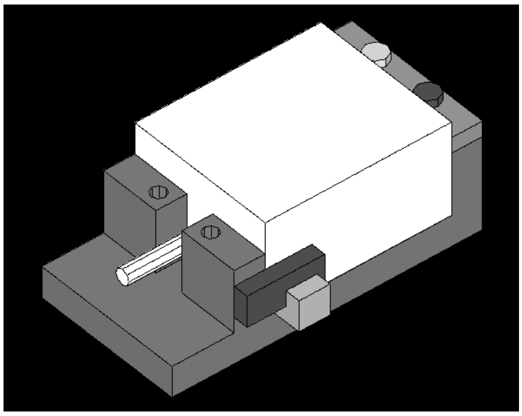


Figure 4. Result of partial disassembly for removal of part 2

If a designer requires the system to keep parts which can be melted in a group for recycle, the system emphasizes them with white color as shown in Figure 5(a) by referring the matrix described in Table 3, and generates disassembly sequences of all parts except them. Figure 5(b) shows the result of removing parts to disassemble. Next, the designer considers the kind of material of remaining parts. If he/she selects parts whose material is SS400 to fuse them together, they are emphasized as shown in Figure 5(b) and all parts whose material is not SS400 are disassembled as shown in Figure 5(c). It is found that part 5 and 6 do not need to be disassembled by considering their recycle. If the designer changes the material of part 2, 3, and 4 into SS400, they also do not need to be disassembled and it seems that the cost for disassembly of this product decreases. He/she can decide the material of parts by considering not only their function but also their disassembly cost for reuse/recycle.

Thus, by using the developed system, designers can evaluate their design of a product when its maintenance, repair, recycle, and/or disposal are considered and they can improve its shape, configuration, structure, and/or material, if necessary.

Table 3. Example of database of product shown in Figure 3 for its reuse/recycle

No.	material	life span	melted?	clashed?	expensive?
1	SS400	short	yes	no	no
2	SUS304	long	yes	no	yes
3	A7075	long	yes	no	yes
4	A7075	long	yes	no	yes
5	SS400	short	yes	no	no
6	SS400	short	yes	no	no
7	ABS	long	no	yes	no
8	ABS	long	no	yes	no
9	SS400	short	yes	no	no
10	SS400	short	yes	no	no
11	SS400	short	yes	no	no
12	SS400	short	yes	no	no

7. Conclusion

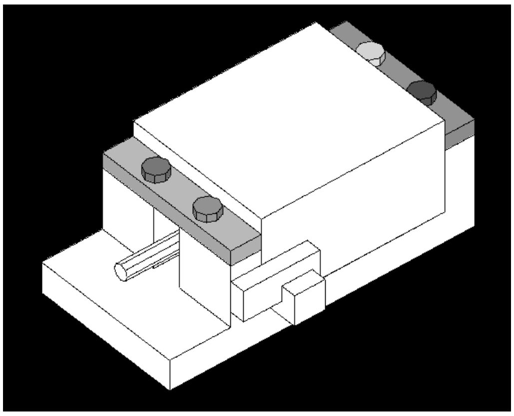
A disassembly support system which can be applied to the design considering reuse/recycle was developed. Firstly, methods to calculate possible motions of parts and to detect configurations where contact state transition occurs were presented. Based on them, an algorithm for verifying disassemblability of parts was explained. Secondly, 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. Thirdly, a disassembly sequence generation system was implemented and its efficiency was evaluated. Finally, we showed that the developed system can be applied to the design considering reuse/recycle of mechanical parts. 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 reuse/recycle.

References

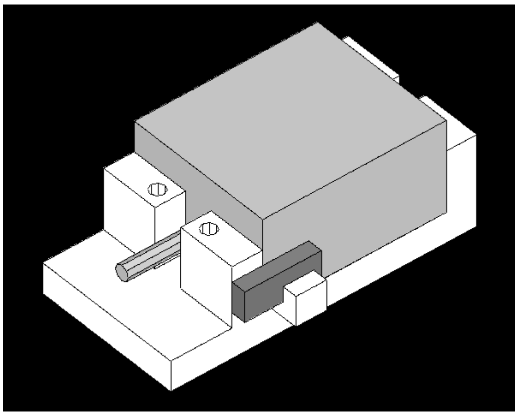
[1] F.Jovane, A.Armillotta, and K.Feldmann. A key issue in product life cycle: Disassembly. *Annals of the CIRP*, 42-2:651–658, 1993.

[2] T.Harjula, W.A.Knight, and G.Boothroyd. Design for disassembly and the environment. *Annals of the CIRP*, 45-1:109–114, 1996.

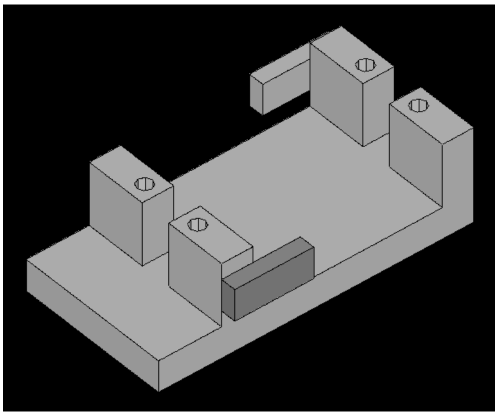
[3] E.Arai, N.Uchiyama, and M.Igoshi. Disassembly path generation to verify the assemblability of mechanical products. *JSME Int. Journal, Series C*, 38(4):805–810, 1995.



(a) Initial state



(b) Result of removing parts which can not be melted



(c) Result of removing parts whose material is not SS400

Figure 5. Example of disassembly of product shown in Figure 3 considering its reuse/recycle