Chapter 15
Disassembly Support for Reuse of Mechanical Products Based on a Part Agent System

Atsushi Nagasawa, Yuki Fukumashi, Yoshinori Fukunaga, and Hiroyuki Hiraoka

15.1 Introduction

The effective reuse of mechanical parts is important for the development of a sustainable society [1]. To realize effective part reuse, it is essential to manage the individual parts of a product over their entire life cycle because they have different reuse histories. However, it is difficult for manufacturers to predict such information owing to the uncontrollable and unpredictable diversity of user behavior. Based on these considerations, we propose a scheme whereby a part "manages" itself and supports user maintenance activities.

For this purpose, we are developing a network agent called a "part agent" that manages its corresponding part [2]. It is programmed to follow its real-life counterpart throughout its life cycle. A part agent collects the information related to its counterpart through the network and provides the users with appropriate instructions for reusing it to promote the circulation of reused parts.

A part agent proposes the reuse of its associated part by evaluating the collected information such as the operational history and deterioration of the part [2]. Because a product must be disassembled to extract the parts to be reused, a part agent shows a user how to disassemble the product and extract the part when the user agrees to the reuse. This paper describes a function that generates the disassembly procedure of a product that defines the sequence to extract its parts and another function that displays the instructions by overlaying them on an image captured by a camera. The former is based on predefined product model data and disassembly order data that preset a disassembling order for extracting each part. In the next section, we present the part agent system that is under development. The scheme of the disassembly support system including the representation of the assembly data of the product and generation of the disassembly instructions is described in Sect. 3. A prototype

A. Nagasawa (✉) · Y. Fukumashi · Y. Fukunaga · H. Hiraoka
Department of Precision Mechanics, Chuo University, Tokyo, Japan
e-mail: nagasawa@lcps.mech.chuo-u.ac.jp

A. H. Hu et al. (eds.), *Technologies and Eco-innovation towards Sustainability II*,
https://doi.org/10.1007/978-981-13-1196-3_15

system that we developed and its experimental results are explained in Sect. 4. It is to be noted that destructive disassembly must be considered for reusing a part. Therefore, we designed a concept of supporting destructive disassembly by a user. This is described in Sect. 5. Section 6 concludes the paper.

15.2 Part Agent System

A part agent manages all the information regarding its corresponding part throughout its life cycle. The proposed scheme assumes a spread of networks and high-precision radio-frequency identifier (RFID) technology [3].

A part agent is generated when a part is fabricated and an RFID tag is attached to this part. The part agent uses the RFID tag for tracking the life cycle of the part through the network. We chose an RFID tag for identification because RFIDs have a higher resistance to a smudge or discoloration than printed bar codes and will operate throughout the life of a part.

Figure 15.1 shows the conceptual scheme of a part agent. The part agent communicates with various functions within the network and collects the information required to manage its corresponding part such as product design information, predicted deterioration of parts, logistic information, and market information. It also

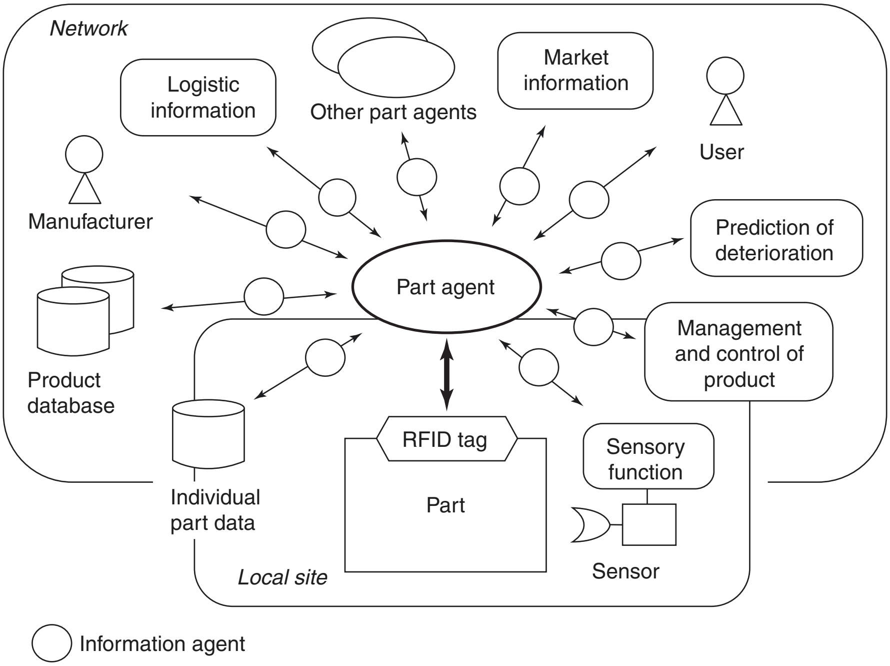


Fig. 15.1 Conceptual scheme of a part agent

communicates with on-site local functions such as sensory functions that detect the state of the part, storage functions for the individual part data, as well as management and control functions of the product. The communication is established through information agents that are subordinate network agents generated by the part agents [4].

15.3 System for Generating Product Disassembly Instructions

Figure 15.2 displays our proposed scheme to support the disassembly of a product. Herein, a part agent proposes the reuse of the corresponding part by predicting its future state based on the current level of deterioration, its operational history, market information of new/used products, and preferences of the user. When a user accepts this suggestion, the agent launches the support system for disassembly. This support system receives the product model data and disassembly order data from the agent and generates the disassembly procedure data to support the disassembly. The disassembly order is the order in which a product is disassembled. It is represented by a sequence of identifiers of the parts to be extracted. We assume that this order is generated when the assembly is designed and is provided to the part agent. The disassembly procedure is the sequence of disassembly operations that must be performed to disassemble a product. For every part in a disassembly order, necessary disassembly operations are generated using the product model data.

Based on the disassembly procedure data, the system displays the instructions for the disassembly by overlaying them on an image captured by a camera.

Product model data is necessary for generating the disassembly operations and displaying the disassembly instructions on the image of the actual product. We developed a representation of the product model using two data structures focusing

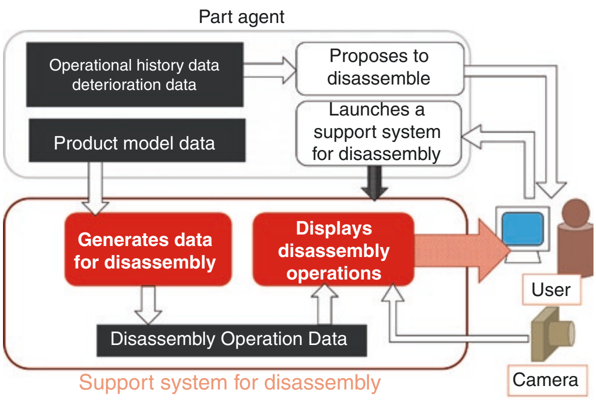


Fig. 15.2 Scheme to support product disassembly

on the assembly. They represent the structure of the assembly and connections between the parts.

L.S. Homem de Mello [5] proposed a relational model graph to represent the contacts and attachments between the parts in an assembly. We designed tree and graph structures for the part agent system. They include the information of the location, shape, and connections of the parts.

Figure 15.3 presents a representation of the assembly structure. This tree structure provides the configuration of the assembly with the positions and postures of all the assembled parts. In this figure, a node represents an assembly, a subassembly, or a part; and an arrow denotes the relation between the assembly and a component. Each part includes the shape data in its local coordinate system. A node contains the coordinate transformation from its local coordinate system to that of its parent node. The parts are connected in the assembly. A portion of the shape of a part that is connected to another part, such as a pin, hole, and screw, is called as an assembly feature. An assembly feature also includes the coordinate transformation to the part it belongs to.

Figure 15.4 displays a representation of the connections between the parts. This graph structure shows the connections between the assembly features of the parts and its states. Each connection includes a pair of assembly features and a type of assembly relation called "connection property" that connects the features.

The assembly properties are referenced for providing a user with information on the disassembly operations such as pulling a part, turning a part, and removing bolts. When connections of all the assembly features of a part are disconnected from those of other parts, the part is removed from the assembly.

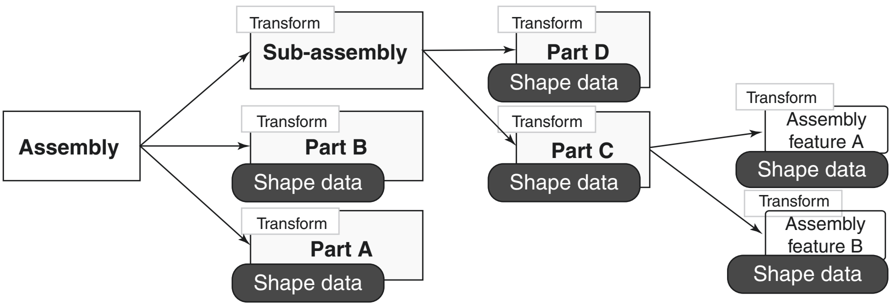
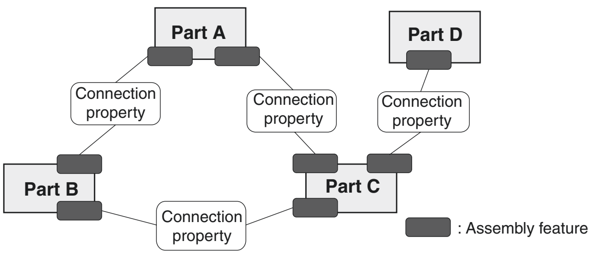


Fig. 15.3 Representation of the assembly structure

Fig. 15.4 Representation of the assembly connections



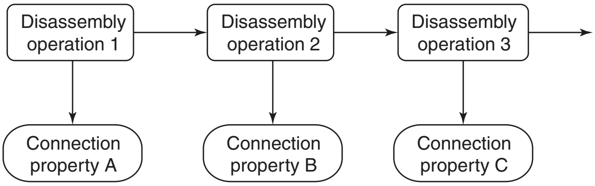


Fig. 15.5 Representation of the disassembly procedure

Figure 15.5 shows a representation of the disassembly procedure that is generated from the assembly connection data and disassembly order data. It consists of the disassembly operations in a tree structure, with each operation having a connection property that must be disconnected for removing the parts according to the disassembly order data. The system displays the instructions for the disassembly operations based on this data.

Each part agent includes the assembly structure and assembly connections of its real-life counterpart. When a part agent launches a disassembly support system, it sends this product model data to the system. Subsequently, the system generates the disassembly procedure based on the assembly connections and calculates the global positions of every part from the assembly structure. Using these results and a marker recognized in a captured camera image, the system displays the disassembly instructions on the image.

Figure 15.6 is a flowchart of an example scheme of the support disassembly of a pinhole connection in the prototype system described in the next section. The arrows identify the pull direction using the transforms of the feature, part, and upper one in the assembly structure tree.

This example scheme has the following steps for displaying the information:

1. Generate the disassembly procedure.
   The disassembly procedure is generated based on a predefined disassembly order. To extract a part, all the connections of the part should be removed. To remove each connection of a part, a corresponding operation is generated based on the connection property and is added to the procedure.
2. Obtain the next connection property.
   Information to support the disassembly is presented to a user based on the generated disassembly procedure. Each operation in the disassembly procedure has a target connection property to be removed. For each connection property, such as a pinhole connection, the corresponding information is generated and presented as follows.
3. Determine the parts related to this connection.
   The system contains the assembly model of the product to be disassembled that includes its assembly structure and connections. A part that contains the connection property to be removed is identified based on this model information.
4. Determine the parts related to this connection.
   The system contains the assembly model of the product to be disassembled that includes its assembly structure and connections. A part that contains the connection property to be removed is identified based on this model information.

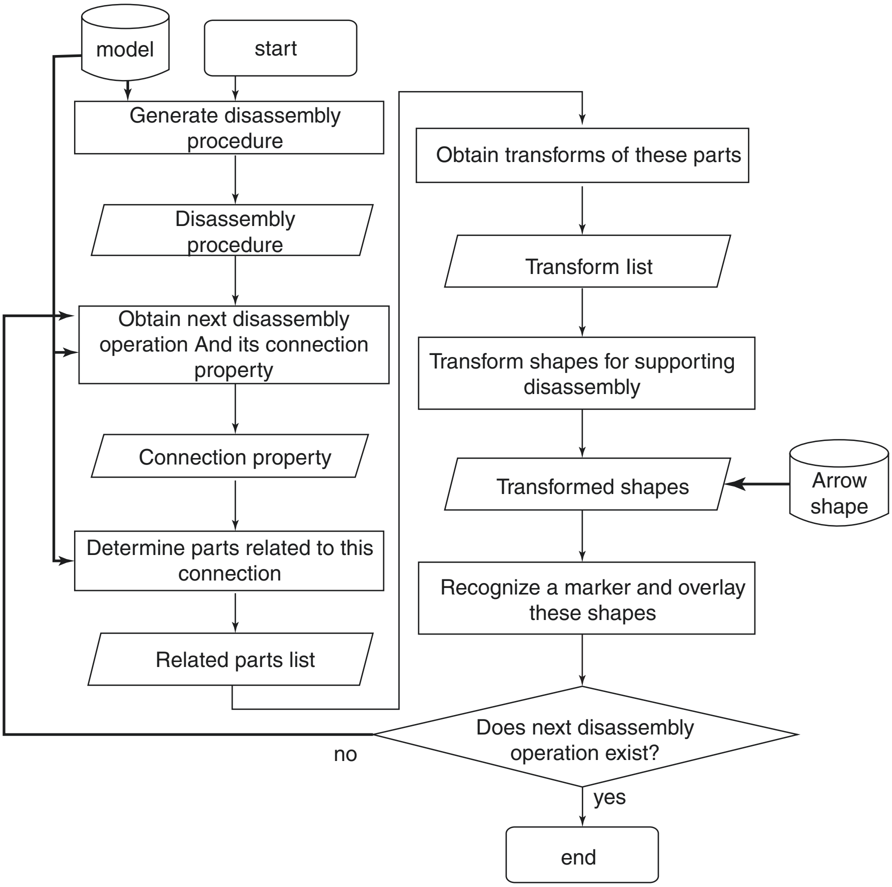


Fig. 15.6 Example scheme to support disassembly of a pinhole connection

5. Obtain the transforms of the parts.
   As described previously, each part includes a coordinate transformation that represents the current position and posture of the part.
6. Transform the shapes to support the disassembly.
   Shapes such as an arrow should be appropriately displayed in the local coordinate system of the part to show a user the necessary motion of the part to be extracted. These shapes are transformed based on the transformation of the part.
7. Recognize a marker and overlay the shapes.
   These shapes should be displayed relative to the real coordinate system. The real coordinate system is calculated by recognizing a marker included in an image captured by the camera. Using this information, the system displays correctly the shapes for the instructions for the disassembly operations by overlaying them on the acquired image.

15.4 Experiment for Disassembly

We developed a prototype system to support disassembly and the required data structures in the Java language. The system uses OpenCV, an open-source computer vision software library, for marker recognition in a camera image. To test this prototype system, we constructed a simple example assembly and generated its product model data.

The example assembly consists of two parts, Part A and Part B, as shown in Fig. 15.7. A hole of Part A and pin of part B are connected. The corresponding model data is presented in Figs. 15.8 and 15.9. The shapes of the parts and assembly features are described in their own local coordinate systems, as shown in Fig. 15.7. For conducting the experiment, we used the model data as inputs and obtained the system generating disassembly operation for extracting Part B.

The system loads the assembly structure and connection data and generates the disassembly procedure. For displaying the information, the system obtains the

Fig. 15.7 Example assembly

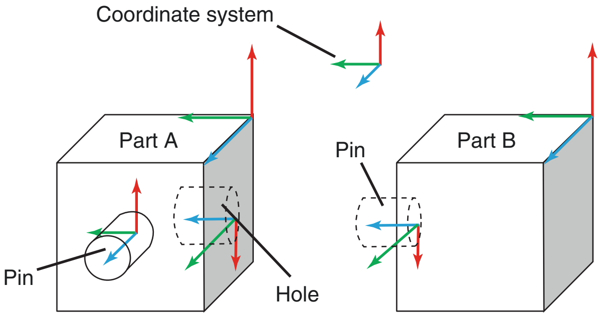


Fig. 15.8 Structural data for the example assembly

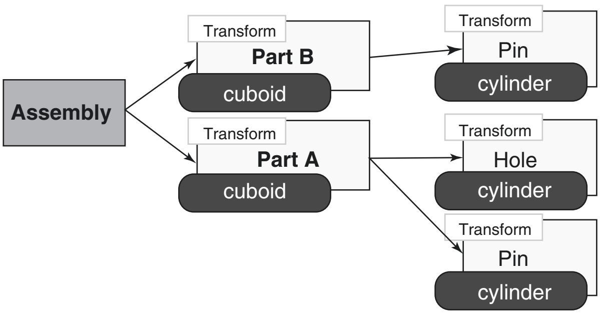
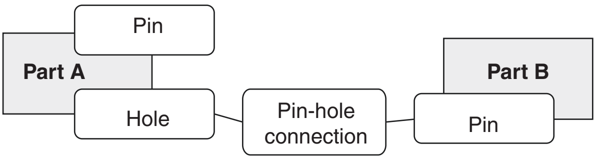


Fig. 15.9 Connection data for the example assembly



shape data of Part B and its transforms that transform the coordinates of the related parts and feature to those of the assembly. The system recognizes a marker in an image captured by the camera and calculates the position and posture of the recognized marker. It shows the shapes to support the disassembly based on the transforms, position, and posture.

The experimental results are displayed in Figs. 15.10 and 15.11. In Fig. 15.10, the shape of Part B and instruction for its disassembly are shown on the captured camera image. The arrow in the figure denotes the direction to pull Part B. Note that a rectangular marker is attached to Part B for the recognition of the position of the assembly. This result shows that the developed function is successful in displaying the disassembly operations by overlaying them on a captured camera image and through marker recognition.

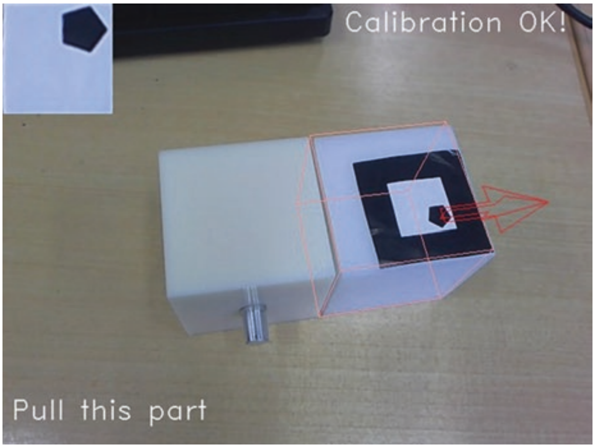


Fig. 15.10 Instruction displayed for disassembly

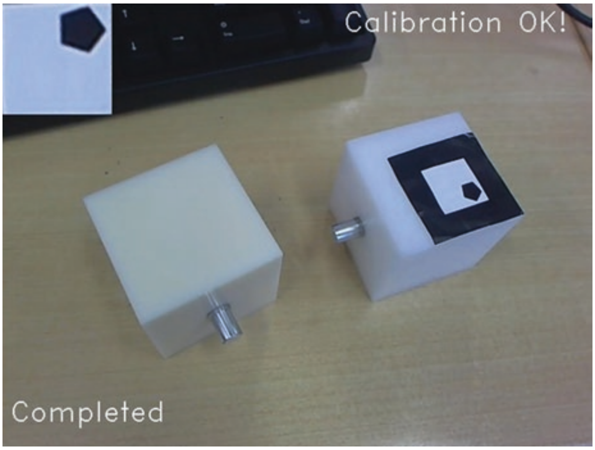


Fig. 15.11 Disassembly completed based on the instruction

15.5 Supporting Destructive Disassembly

Although the function formulated previously works for new products, we must consider the deterioration of parts for generating the procedure for the disassembly of used products. The deterioration of a part may hamper removing the connections between the parts, which occurs such as in a stuck fastener, distorted connection, and eroded bolt head.

In this paper, to deal with this problem, we propose the following two methods for the improvement of the function to disassemble a product. The first method is to include branches in the disassembly order of the disassembly procedure to provide an alternative sequence of disassembly operations. If an operation is found to be infeasible during a disassembly process, the system generates another disassembly operation based on an alternative branch sequence in the disassembly procedure. Figure 15.12 shows an example of the branched disassembly procedure. In this procedure there is no precedence between disassembly operation 2 that removes connection B and disassembly operation 3 that removes connection C. When disassembly operation 2 is found to be infeasible in the upper branch of this procedure, the system attempts disassembly operation 3 in the lower branch that may make disassembly operation 2 feasible or may make it possible to extract a target part.

The other method is to propose a disjunction by breaking the connection or the part that blocks the disassembly when no alternative disassembly operation is available. This destructive disassembly is often used in practical situations when the aim of disassembly is to extract only precious or poisonous materials.

Shiraishi et al. [6] developed a method to support destructive assembly in the product design phase by adding split-lines to the shells of a product. A split-line is provided on a shell of a part along which the shell is intentionally made prone to breakage. This leads to easier extraction of a reusable part inside the shell by the destruction of the shell. This method requires modification of the product design and provides the information about the position of the split-lines to a user.

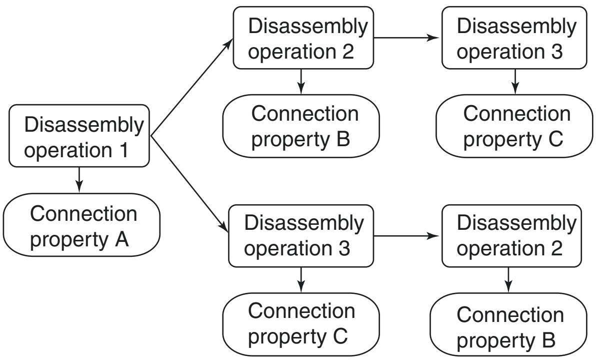


Fig. 15.12 Representation of the disassembly procedure including branches

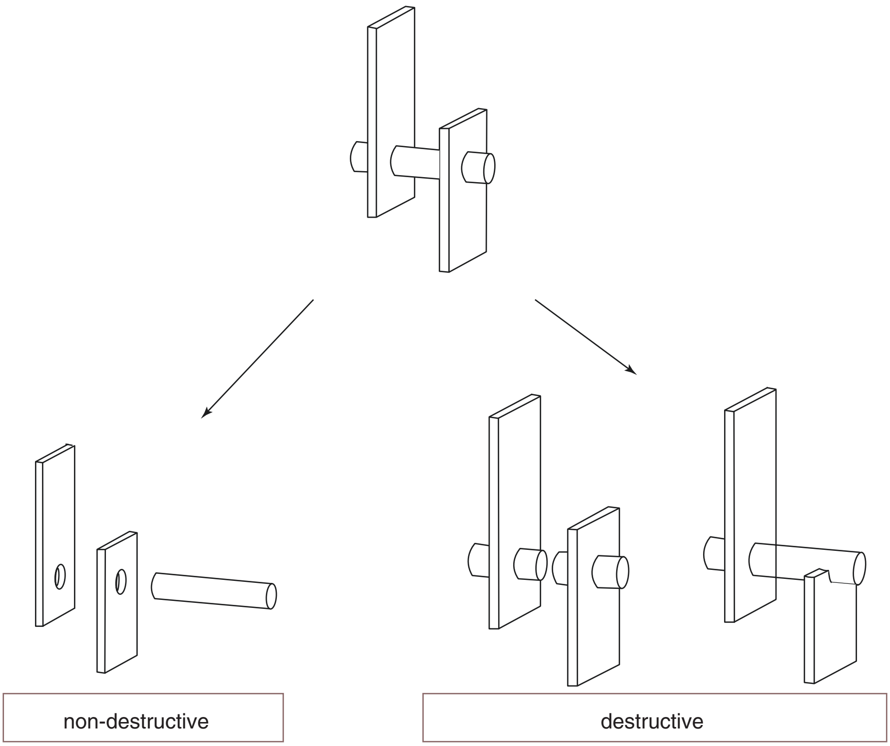


Fig. 15.13 Example of destructive disassembly

We propose a concept of destructive disassembly based on a type of connection property. If a nondestructive disassembly operation on a connection property is not feasible, the system displays the information for the destructive disassembly.

Figure 15.13 illustrates an example of destructive disassembly. In this case, a reusable part can be extracted if a pin shown in the figure is removed. If this pin cannot be removed by nondestructive disassembly, destructive disassembly must be performed. When a user informs the system that nondestructive assembly is infeasible, the system shows the information for the destructive disassembly based on connection properties such as cutting the pin and breaking the hole, as shown in this figure. For this purpose, predefined data for the destructive disassembly is provided to each connection property including the destructive part and its region, with additional information such as split-lines.

An appropriate representation and the contents of the information that are provided to the connection property for destructive assembly are still to be developed. We are also developing a prototype system to evaluate the proposed method.

Each connection property includes the predefined data for destructive disassembly including the destructive part and its region. The system shows a split-line or other information for the destructive disassembly.

15.6 Conclusion

A prototype system for a part agent is developed to support the disassembly of a product using a product model that consists of two data structures focused on the assembly. This system supports users for disassembling a product by displaying instructive information that is overlaid on an image captured by a camera.

The conceptual scheme of supporting destructive disassembly that is designed displays the necessary information based on the connection properties. In the future, it will be implemented in a prototype system.

The remaining issues and future prospects are related to the application of this system in more practical examples that require further investigation of the product models, complex disassembly operations, generation of the disassembly procedure, and display of the disassembly instructions.

Acknowledgments This work was supported by JSPS KAKENHI Grant Number 24560165.

References

1. Hauschild M, Jeswiet J, Alting L. From life cycle assessment to sustainable production: status and perspectives. CIRP Ann. 2005;54:535–55.
2. Hiraoka H, Ueno T, Kato K, Ookawa H, Arita M, Nanjo K, Kawaharada H. Part agent advice for promoting reuse of the part based on life cycle information. In: The 20th CIRP international conference on life cycle engineering, Singapore. Apr 17–19 2013.
3. Borriello G. RFID, tagging the world. Commun ACM. 2005;44:34–7.
4. Nanjo K, Yamamori Y, Yokoki Y, Sakamoto Y, Hiraoka H. Maintenance decisions of part agent based on failure probability of a part using Bayesian estimation. In: The 22nd CIRP conference on life cycle engineering, Sydney. Apr 7–9 2015.
5. Homem De Mello LS, Sanderson AC. A correct and complete algorithm for the generation of mechanical assembly sequences. IEEE T Robotic Autom. 1991;7:228–40.
6. Shiraishi Y, Miyaji N, Fukushge S, Umeda Y. Proposal of a design method for dismantling products with split-lines. Procecia CIRP. 2015;29:114–9.