Eco-Modular Product Architecture Identification and Assessment for Product Recovery

AUTHORS:
Samyeon Kim
Graduate Student
School of Mechanical & Aerospace Engineering
Nanyang Technological University
Singapore 639798
Email: samyeon001@e.ntu.edu.sg

Seung Ki Moon*
Assistant Professor
School of Mechanical & Aerospace Engineering
Nanyang Technological University
50 Nanyang Avenue, N3.2, 01-29
Singapore 639798
Phone: +65-6790-5599
Fax: +65-6792-4062
Email: skmoon@ntu.edu.sg

*Corresponding author

Abstract

In order to improve the efficiency of disassembly and product recovery of an abandoned product at the end-of-life stage, it is essential to develop modular product architecture by considering manufacturing and recovering processes in early product design stage. In this paper, a novel concept of a design methodology is introduced to develop eco-modular product architecture and assess the modularity of the architecture from the viewpoint of product recovery. Eco-modular product architecture contributes to enhancing product recovery processes by recycling and reusing modules without full disassembly at component or material levels. It leads to less consumption of natural resources and less landfill damage to the environment. Three sustainable modular drivers, namely, interface complexity, material similarity, and lifespan similarity, are introduced to reconstruct the modular architecture of commercial products into the eco-modular architecture. Alternatives of modular architectures are identified by Markov Cluster Algorithm based on these sustainable modular drivers and physical interconnections of the components of product architecture. To select the eco-modular architecture from these alternatives, we propose modularity assessment metrics to identify independent interactions between modules and the degrees of similarity within each module. To demonstrate the effectiveness of the proposed methodology, a case study is performed with a coffee maker.

Keywords: Eco-module, Markov Cluster Algorithm, Modularity assessment, Product architecture, Product recovery

1 Introduction

As governments become more stringent concerning environment issues and take more interest in sustainability, manufacturers are also required to meet environmental legislations and produce eco-friendly products (Ramani et al. 2010; Ilgin and Gupta 2010). However, with the rapid development of technology, high-end products, especially electronic products, incite customers to purchase goods without thinking of the impact it will have on the environment. The technology push strategy of a company results in mass production of new products and frequent product replacements by the customers. The mass production for new products increases the impact on environmental pollution during the manufacturing process. Although a product is still good enough to use, when the product is discarded, the product lifecycle shortens. This has a negative effect on the environment as the landfill rate heightens. Therefore, product recovery must be considered in the early product design stage to minimize the landfill rate.

To recover products in the end-of-life (EOL) stage, product architecture development is necessary to facilitate the disassembly process. If product architecture is complex at the disassembly stage, individual parts of the product can be difficult to retrieve and can increase the chances for damage (Kim et al. 2015; Behdad et al. 2014). In this sense, modular product design should be one of the most important considerations to improve disassemblability for the product recovery. Modular architecture mainly contributes to diminishing the complexity of the product architecture by identifying physical chunks (Pimmler and Eppinger 1994; Ulrich and Eppinger 2008). As the modular architecture has a higher number of internal couplings within the module and less external couplings between modules, the modular product can be maintained or replaced effectively by substituting the modules (Ulrich 1995; Chiriac et al. 2011). However, most of the modules in previous research are generated based on the functional or physical relationships of the product architecture (Yu et al. 2007; Thebeau 2001; Du et al. 2014), and predefined as module variants by using optimization approaches (Du et al. 2014; Kwak et al. 2009; Fujita and Yoshida 2004; Yigit et al. ; Moon et al. 2014), market-based decision making approaches (Moon and McAdams 2012) or data mining approaches (Moon et al. 2010). Also, most of the previous modular product designs have focused more on clustering modules to maximize internal coupling and minimize external coupling without considering product recovery processes. Although the existing modules are well-defined and effective for manufacturing, these modules would not be easy to disassemble at the product recovery stage. If components with different materials and lifespans are mixed in a module, disassembly operators need to make more effort to fully disassemble the module into components or materials (Otto and Wood 2001).

To develop modular architecture from the viewpoint of the product recovery, we introduce a novel design methodology to extend existing research from modular product design to eco-modular product design. The objective of this research is to determine eco-modular architecture and assess its degree of modularity by considering the product recovery based on sustainable modular drivers. The sustainable modular drivers consist of interface complexity, material similarity, and lifespan similarity. With the sustainable modular drivers, Markov Cluster Algorithm is applied to identify eco-modules, which is a chunk of components with similar materials and/or lifespans. Also, a novel modularity assessment method is proposed to quantify the degree of modularity of the eco-modular architecture by using module independence and module similarity metrics. The module independence metric indicates the degree of independence by assessing interface complexity between modules, while the module similarity metric indicates the degree of similarity within modules. Based on the results from the metrics, the eco-modular architecture can be selected.

The subsequent sections of this paper are structured as follows. Section 2 reviews literature related to modular product architecture design and assessment, and product design for product recovery. Section 3 describes the proposed methodology to guide eco-modular architecture design. Section 4 provides a case study to demonstrate the effectiveness of the proposed methodology and section 5 presents the conclusion, limitations and future work of this paper.

2 Literature review

2.1 Modular product design and its assessment

In product recovery, modular product design is one of the solutions to improve disassemblability and manage the efficiency and difficulty of the disassembly. The modularity has been developed to improve manufacturing production (Jose and Tollenaere 2005), maintainability, serviceability (Otto and Wood 2001), and disassemblability for the end of product lifecycle (Newcomb et al. 1998). In order to identify the modules, product architecture was largely represented by matrix and graph based representations. Matrix-based representation makes up rows, columns and matrix elements. The rows and columns are developed by component lists, and the matrix elements represent functional or physical relationships between the components. The matrix elements have a binary value to show the connectivity between components or weighted value to show the strength of connectivity. As for the product information to be used for modularization, physical structure information and functional information were mainly considered. Firstly, physical relationships are fundamentally used to figure out modules. The design structure matrix (DSM) is a popular method to represent the physical relationships between components. Yu et al. (2007) considered the physical relationships using the DSM method and applied minimum description length (MDL)-based clustering method with a genetic algorithm (GA) to identify modules. The feature of the clustering method is to allow bus modules and overlapping modules. Yan et al. (2012) utilized the DSM to establish the correlation between the components based on the strength of interaction and then identified modules with the kernel-based fuzzy c-means algorithm. Secondly, with regard to functional information, Helmer et al. (2010) used information values about types of interactions, such as structural, spatial, signal, material, and energy, from the research of Pimmler and Eppinger (1994) and Sosa et al. (2003). Chun-Che and Kusiak (1998) interpreted three types of modularity, namely, component swapping, component sharing and bus modularity, and determined modules based on the decomposition approach and three axioms related to the three types of modularity. To identify the three modules, physical interactions and suitability values were considered in the DSM. The minimum description length (MDL)-based DSM clustering was utilized to cluster modules based on the types of functional interactions. Asikoglu and Simpson (2012) measured the weighted modular complexity score (wMCS) according to different types of interfaces. Then, Jung et al. (2014) used the wMCS and the GA to improve the modularity of products. Heuristic methods are also applied to modularize functional structures based on shared and unique functions (Dahmus et al. 2001; Krause et al. 2014; Hölttä and Otto 2005). By utilizing the heuristic method, a group that shares similar functions and flows should be clustered in a single module, and components with unique functions can be grouped in a module. Erixon (1998) introduced the modular function deployment (MFD) which heuristically determines modules by maximizing similarity technical solutions in terms of module drivers.

Aforementioned methods developed modules based on either the physical relationship or the functional relationship. To overcome this issue, Borjesson and Hölttä-Otto (2014) suggested a new modularization algorithm to consider both the DSM for the physical relationships and the modular function deployment (MFD) for functions. Meng et al. (2007) proposed a multi-objective combinatorial optimization problem for module identification by using the GA. To consider module configurability, these kinds of information were analysed and used as follows: the simultaneous use of module alterability and module manufacturing efficiency; and both physical and functional information, such as functional relationships and structural relationships.

After identifying various module candidates by considering various information, modularity assessment of the module candidates was necessary to evaluate whether the modules were well-clustered to meet designers' intent or not. Fixson (2005) introduced product architecture assessment with a function-component allocation scheme to distinguish modular or integral architectures; interface characteristics, such as type, reversibility, and standardization were also measured. Chiriac et al. (2011) analysed modular product architecture to determine the degree of modularity and the level of system granularity. Newcomb et al. (1998) developed similarity metrics to assess modularity for the EOL stage by comparing the correspondence of the inner module's connection between more than two product architectures for the product lifecycle. Moreover, the interface design related to compatibility and interchangeability was significantly considered when the module was defined. Well-designed interfaces can minimize complexity and uncertainty in disassembly operation. Pimmler and Eppinger (1994) measured the degree of generic interaction by using categories of energy, spatial, information, and materials. Fixson (2005) additionally analysed the reversibility of interfaces for product changes during product lifecycle, such as upgrades, adaptation, wear, or reuse, together with interface standardization. Dobberfuhl and Lange (2009) proposed a modular complexity score (MCS) to identify the degree of modules' complexity. Asikoglu and Simpson (2012) introduced wMCS based on the MCS. The wMCS includes six different types of interface and its weight value. Jung and Simpson (2014) developed two modularity indices to measure the proximity of components to the diagonal of a DSM and to measure modularity independency.

On the literature review on modular design, most of the modularization methods had identified modules to minimize product complexity for manufacturing based on physical or functional interrelationships between components. However, few studies have addressed how to identify the modules to enhance product recovery processes. In order to develop eco-modular architecture, we use current modular product design as the foundation of modularization and extended the design to consider the product recovery. To consider product recovery, modular drivers related to the product recovery are needed to identify the eco-modular architecture. To measure the modularity of the modular architecture, most of the modularity assessment methods have focused on measuring intra- and inter- connectivity of modular architectures. Modularity assessment in previous research has not considered the product recovery in their product design. The eco-modular product design focuses on the modular architecture by considering how the product will be recovered. Therefore, this new concept of a modularity assessment method is needed to assess the eco-modular architecture based on the modular drivers.

2.2 Product recovery

Product recovery at the EOL stage is a promising solution to the challenges of achieving sustainable product design and sustainable business. The product recovery is a process of restoring the inherent value of retired products when the products no longer satisfy customers' needs (Lindahl et al. 2006). Reusable components and modules, and recyclable materials are retrieved from the retired products for the product recovery which results in the retrieved modules, components and materials having a second life. The main contribution of the product recovery is to minimize the amount of waste disposal and the usage of natural resources by recovering obsolete parts to serviceable parts. Recovery options include recycling and reuse; additionally, reconditioning, refurbishment, and remanufacturing options are utilized to retrieve products or modules for reuse (Thierry et al. 1995).

In order to consider the product recovery in early product design stage, design for disassembly (DfD) was researched to easily and effectively take products apart for reuse and/or recycle in the EOL stage (Lambert 2002). DfD mainly focused on disassembly sequence planning by minimizing disassembly cost and time rather than identifying the modules (Behdad et al. 2014; Smith et al. 2012; GÜngÖr and Gupta 2001; Smith and Chen 2011). Several studies dealt with product architecture, EOL values and strategies, and the DfD. Kwak et al. (2009) introduced a new concept of eco-architecture to allocate modules with the EOL strategy from the viewpoint of product retirement. Linear programming was used to identify eco-architecture that consists of eco-modules and its EOL strategy based on disassembly cost and an economic value. Behdad et al. (2010) extended the work of Kwak et al. (2009) to consider sharing disassembly operations between products with similar product architecture. Modular architectures were assumed and represented by transition matrix based on a disassembly order. Ishii et al. (1994) suggested a method for evaluating design for product retirement based on modular architecture with quantitative retirement cost analysis and qualitative guidelines. The major factors of the evaluation method were material compatibility and system disassembly cost. Although the method was built based on the modular architecture, module identification was assumed. Tseng et al. (2008) clustered modules based on liaison intensity values that measured assembled connections between components. The grouping genetic algorithm was applied to maximize the liaison intensity within the modules and to minimize the liaison intensity between the modules, as carried out in the conventional modularization method.

As aforementioned, previous research for product recovery has focused on the disassembly planning for minimizing disassembly costs based on predefined modules rather than configuring modules and modular architecture for the product recovery strategy. On the other hand, previous research for modular product design has mainly focused on developing modular product architecture for manufacturing by decomposing a product into modules and components at the system-level design stage of the product development process (Ulrich and Eppinger 2008). Although a product concept has already been determined, alternatives of feasible product architecture can be developed to reflect a product recovery strategy into product architecture design towards a sustainable product at this stage. In this respect, specific design information related to the product recovery strategy from detail design stage, like material types and lifespan of components, is used in this research. Therefore, the proposed methodology can be used at the early product design stage, specifically, the system-level design and detail design stages to develop an eco-modular product architecture. In terms of facilitating the product recovery at the product architecture level, the proposed methodology contributes to design activities for enhancing the recyclability and reusability of products without further disassembly at material or component levels.

3 An eco-modular architecture design and assessment methodology

The proposed eco-modular architecture design methodology is shown in Figure 1. A design team developing a new product portfolio by considering product recovery would carry out the process outlined here. The process is similar in determining modular product architecture based on the basic physical interrelationships of each product in the product portfolio. However, the design team would include considerations of the product recovery process into its product architecture. We define eco-modular architecture as a scheme by which the components with similar design features are organized into modules to enhance end-of-life characteristics of the architecture. The first step is to analyze current product architecture and product information. Designers can obtain the information on components' interaction based on the DSM, the types and number of interfaces, its complexity, material types, component lifespan, environmental impact and so on from this initial step. In step 2, sustainable modular drivers are identified and measured to cluster components into modules for product recovery. It is important to produce a feasible architecture design, so that physical connections between components are considered to develop eco-modular product architecture. The DSM from step 1 and the values of the sustainable modular drivers from step 2 are converted to an adjacency matrix in step 3. Based on the adjacency matrix, the Markov Cluster Algorithm is applied to identify eco-modular architectures while considering both the physical relationships between components and the sustainable modular drivers. The final step is to assess the modular architectures based on the module independence and similarity metrics. The module independence metric can measure interface complexity between modules to verify how complex the modular product is when disassembling modules at the EOL stage. The module similarity metric assesses the similarity of characteristics of the components in modules of a product. Finally, decision makers can select an eco-module architecture based on the assessment result.

Figure 1 will be inserted here

3.1 Product architecture analysis

Product architecture can be defined as comprehensive representations of a set of characteristics, such as the number and type of components, the number and type of interfaces between components, and the fundamental structure of the product (Fixson 2005). In order to emphasize design for recovery, the product architecture is explained with disassembly process as shown in Figure 2. A product is disassembled into components through modules which are sub-assemblies. The disassembly process incurs different amounts of disassembly process costs according to the difficulty of disassembly and the condition of modules after disassembly (Otto and Wood 2001). In the disassembly process of the architecture level, designers can collect information on the negative environmental impact, disassembly cost, product recovery cost, and reassembly cost. The product recovery and reassembly operations are then conducted to retrieve obsolete components. After disassembling the modules into components, the components can then be further disassembled into materials if a module contains components with different materials. The component based DSM is developed to represent interactions between components in a product. The component based DSM is used as a fundamental tool to represent interface complexity in Section 3.2.1 and develop adjacency matrices for the similarities of material or lifespan in Section 3.3.

Figure 2 will be inserted here

3.2 Identification of sustainable modular drivers

Based on the product information, new modular drivers are required to be identified to configure modular architecture for the EOL stage. The modular drivers are used as different criteria to generate modules behind modularization (Erixon 1996). Erixon (1996) introduced 12 modular drivers to satisfy a company's strategic requirements: carryover, technology push, common unit, maintenance and recycling, to name a few.

In this research, sustainable modular drivers related to product architecture and sustainability are introduced to consider disassemblability for the product recovery as shown in Figure 3. An architecture factor is interface complexity; this represents the complexity of functional and geometric connections between components. The interface complexity helps determine modules with high interface complexity between components to minimize interface complexity between the modules. The sustainability factors are component lifespan and materials. The sustainable factors are identified to cluster components with similar material or lifespan to form modules without disassembly at either the component or material levels. These sustainability modular drivers help reconstruct the modular architecture of commercial products into eco-modular architecture.

Figure 3 will be inserted here

3.2.1 Interface complexity

As modular product design aims to maximize internal coupling within modules and minimize external coupling between modules (Ulrich 1995), managing the ‘coupling’, which refers to the interface in this research, is important. The interface connects functions through material, energy, and signal flows among components via geometric contacts (Hirtz et al. 2002). Thus, the interface plays a critical role in estimating the difficulty of disassembly as well as the functional importance between components and modules, quantitatively.

In this research, we adopted interface types and weights that were introduced by Asikoglu and Simpson (2012). The interface types related to the physical connection are the attachment (A) and spatial (S), while the interface type related to functional interfaces are transfer (T), control and communication (C), field (F), and power (P) as shown in Table 1. The complexity of direct connections between components/modules is calculated by using the weighted-modular complexity score (wMCS) in equation (1) (Asikoglu and Simpson 2012).

$$\begin{aligned} wMCS = {} & [w_i \times (\#\text{of } 1\text{'s} \times 1)] + [w_i \times (\#\text{of } 2\text{'s} \times 2)] + \\ & [w_i \times (\#\text{of } 3\text{'s} \times 3)] + [w_i \times (\#\text{of } 4\text{'s} \times 4)] + \\ & [w_i \times (\#\text{of } 5\text{'s} \times 5)] + [w_i \times (\#\text{of } 6\text{'s} \times 6)] \end{aligned} \tag{1}$$

Where, w_i is a weight of an interface type (i= A, S, T, C, F, or P) as shown in Table 2. The weights (w_i) of different types of interfaces in Table 2 were determined based on the frequency of occurrence of interface types by using a non-linear weight function (Asikoglu and Simpson 2012). There are six levels of complexity according to combinations of interface types. The terms “# of 1’s, 2’s, 3’s, and etc.” are the number of interface complexity at each level, while the numbers 1 to 6 following these terms refer to the complexity constants at each level (Dobberfuhl and Lange 2009). To calculate wMCS, the number of interfaces at each level is multiplied by the weights (w_i) and complexity constant.

Table 1 Definition of interface types (Asikoglu and Simpson 2012)

Interface type	Definition
Attachment (A)	The structural connections between two components that require a type of connector (e.g. bolts, screws, rivets)
Spatial (S)	The geometrical and locational constraints of a component with respect to other components (e.g. snap-fit, welding)
Transfer (T)	The flow of materials or power between components (e.g. water flow in a coffee maker or in an iron)
Control & Communication (C)	The relationship of signal or information flow is communicated or controlled by another component. (e.g. circuit boards)
Field (F)	The interaction between two components in which one component can generate heat, vibration, or magnetic field
Power (P)	The electrical connection between two components unlike communications and controls interfaces

Table 2 Weight value of interface types (Asikoglu and Simpson 2012)

Interface	Weight	Interface	Weight	Interface	Weight	Interface	Weight
A	2.974	SC	21.435	ATC	64.025	PCF	76.142
S	1.795	SF	22.843	ATF	66.137	TCP	75.598
P	6.832	SP	17.253	ATP	57.752	TFP	77.710
C	8.923	TC	36.735	ACF	64.568	ASTC	92.546
T	9.445	TF	38.143	ACP	56.184	ASTF	95.362
F	9.627	PF	32.916	AFP	58.296	ASTP	84.182
AS	9.538	CF	37.098	STC	60.488	ATCF	123.872
AT	24.838	PC	31.508	STF	62.600	ATCP	112.692
AC	23.793	AST	42.642	STP	54.215	ACFP	113.418
AF	25.201	ASC	41.074	SCF	61.032	ASTCF	163.815
AP	19.611	ASF	43.186	SCP	52.647	ASTCP	149.841
ST	22.480	ASP	34.801	SFP	54.759	ASTCFP	237.568

For example, in Figure 4 (a), the number ‘1’ in the cells of the DSM means that there are interconnections between two components. Based on the interconnections of the DSM in Figure 4 (a), the interface complexity is determined in the cells that have the interconnections in Figure 4 (b). For example, when the interface type between components *A* and *C* are two screws (Type A) and one snap-fit (Type S), the value of the interface complexity is 7.743 (=2.974 (A)×2 + 1.795 (S) ×1).

Figure 4 will be inserted here

3.2.2 Material

Environmental responsibility of manufacturers is extended beyond the manufacturing stage to customer usage stage and the EOL stage in the product lifecycle. Environmental regulations, such as the WEEE (Waste of Electrical and Electronic Equipment), EuP (Energy Using Product), and ELV (End-of-Life Vehicles), concerns fabrication and disposal of products (Kwak et al. 2009). Carefully selecting materials by considering the customer usage stage and the EOL stage is one of the most important considerations. From the consumer's point of view, certain materials can affect their health; from the manufacturer's perspective, the depletion of natural resources through overconsumption must be considered. Materials are selected in different ways to optimize production methods, function and structural demands, market or user demands, design, price, environmental impact, and lifetime (Ljungberg 2007). To minimize environmental impact, products at the EOL stage should be recycled and reused. Since modular architecture consists of modules that can be disassembled as a chunk, it can inherently improve disassemblability of the products. Accordingly, in terms of design for recycling, components with similar materials should be grouped in a module if components can be recycled by the same product recovery process. This will be advantageous in terms of reducing disassembly costs and recycling processes.

3.2.3 Lifespan

Lifespan is related to the residual value of an EOL product which is changing over time. When a product or a module is disposed of because of physical deterioration or technology obsolescence, the residual value is an indicator of whether or not the module can be performed for a second life. Therefore, the residual value of a product was estimated by its physical and functional lifetimes based on the empirical disposal data of the product in the market. The remaining lifetime (L_R) for the product recovery is represented with its mean lifetime (L_M) and actual lifetime (L_A) under the operation conditions as shown in equation (2) (Mazhar et al. 2007).

$$L_R = L_M - L_A \tag{2}$$

L_M is the component's total functional life and is estimated by the time-to-failure data for a family of components, while L_A depends on the actual used condition. The L_M of each component can be obtained by the empirical time-to-failure data and estimated by Weibull analysis (Kara et al. 2005). The cumulative distribution function for the Weibull distribution is:

$$F(t;\beta,\eta) = 1 - \exp\left[-\left(\frac{t}{\eta}\right)^{\beta}\right] \tag{3}$$

Where, t is time to failure. The distribution is affected by two parameters: the scale parameter, η, and the shape parameter, β. The value of β is closely related to failure mode. When β<1, there is infant mortality. β=1 means that the failure rate is constant over the time (random failure). β>1 indicates a high failure rate as time passes. The mean life L_M is measured by using the following equation (Mazhar et al. 2007):

$$L_M = \eta\Gamma\left[\frac{\beta+1}{\beta}\right] \tag{4}$$

In product recovery, the product lifespan can be an attribute for modular design. With a modular design method, components with similar or high lifespan can be grouped into a module. The components in the module can be treated by the same or similar reuse processes, so that a company can save recovery process costs.

3.3 Identifying eco-modules based on Markov Cluster Algorithm

After measuring and/or collecting the value of the sustainable modular drivers, the value is represented as an adjacency matrix to cluster components with similar material or lifespan into modules. Based on the adjacency matrix as an input, the eco-modular product architectures are configured by Markov Cluster Algorithm (MCL).

The MCL was developed by Van Dongen (2000) and is a commonly used algorithm for clustering networks in the field of bioinformatics (Lei et al. 2016). As this algorithm is simple, there is no high-level procedural instruction for assembling, joining, or splitting of groups (Van Dongen 2000). The MCL can generate modules with high similarity values, and dismiss the connections between components that have low similarity based on the sustainable modular drivers. In this respect, we used the MCL to generate eco-modular architectures.

The clustering process by the MCL is explained in Figure 5. Intuition of graph clustering is to group nodes connected with high strength in one cluster. The MCL finds clusters in graphs by a mathematical bootstrapping procedure. The procedure is to compute random works through the graph by using stochastic matrix (also called Markov matrix) that represents the mathematical concept of the random walks on the graph (http://micans.org/mcl/). The random walks in the MCL algorithm infrequently transit from node i in one cluster to another cluster to lead to node j in the same cluster (Bernardes et al. 2015). The MCL is a fast and scalable unsupervised clustering algorithm based on simulation of stochastic flow in a graph.

Figure 5 will be inserted here

In the MCL, an adjacent matrix is developed with values of the sustainable modular drivers as weight value on the edges. The edges are identified by the physical relationships in the component based on the DSM. Then, the adjacent matrix is converted to a weighted undirected graph. Let $G=(V,E)$ be the undirected graph that includes self-loops; V denotes the node set and E means the edge set. A node in V is v_i, and an edge between v_i and v_j in E is represented as (v_i, v_j). A weight value on the edge is denoted as $w(v_i, v_j)$, which represents the interface complexity, material similarity, or lifespan similarity. A is the adjacency matrix of a weighted and undirected graph as follows:

$$A(i, j) = \begin{cases} w(v_i, v_j) & \text{if } (v_i, v_j) \in \mathrm{E} \\ 0 & \text{else} \end{cases} \tag{5}$$

When developing the adjacency matrix for material similarity, the material similarity is decided according to the experts' knowledge based on the material information, such as material property, conditioning process, recovery process, and so on. In this research, to simplify calculating the material similarity, the weight value of the material similarity is represented between 0 and 1. Equation (6) is an example of the weight value.

$$w(v_i, v_j) = \begin{cases} 1 & \text{if same material is used in } i\text{th and } j\text{th components} \\ 0.5 & \text{if similar materials are used} \\ 0.3 & \text{else} \end{cases} \tag{6}$$

For measurement of lifespan similarity between two components, the lifespan of components has to be predicted by designers based on the accumulated company data and knowledge. Then, the lifespan similarity between components is decided to build an adjacency matrix. Calculation of the lifespan similarity is represented by a ratio of the minimum lifespan and maximum lifespan between v_i and v_j in equation (7). If the lifespan is the same, the lifespan similarity would be 1. On the other hand, the lifespan similarity value would be close to 0. This means that one of the components has a short lifespan or there is a big difference in lifespans.

$$w(v_i, v_j) = \frac{\min\ \text{lifespan}(v_i, v_j)}{\max\ \text{lifespan}(v_i, v_j)} \tag{7}$$

When considering these modular drivers together, the weighted value of edges ($\overline{W}$) is calculated by the weighted average method as illustrated in equation (8).

$$\overline{W}(i, j) = \frac{w_1 \times \text{Interface complexity}(i, j) + w_2 \times \text{Material similarity}(i, j) + w_3 \times \text{Lifespan similarity}(i, j)}{w_1 + w_2 + w_3} \tag{8}$$

$$\sum_{i=1}^{3} w_i = 1$$

Where, w_i is the weighted value of each sustainable modular driver.

After building the adjacent matrix, the adjacency matrix *A(i,j)* should be transformed to Markov matrix, *M*, that is diagonally similar to a symmetric matrix. In order to identify random walks from the adjacency matrix, the sum of all outgoing edges, which is the sum of each column, equals to one as shown in equation (9). So, *M(i,j)* represents the probability of transition of the random walks, which is termed as stochastic flow, from v_i to v_j (Van Dongen 2000).

$$\mathrm{M}(i, j) = \frac{A(i, j)}{\sum_{k=1}^{n} A(\mathrm{k}, \mathrm{j})} \tag{9}$$

The MCL process contains two operations: *expansion* and *inflation*. The expansion operation is normal matrix multiplication and represented as $M_{exp} = M \times M$. It allows the flow to connect different regions of the graph and becomes more homogenous. The expansion corresponds to random walks with many steps. It associates new probabilities with all pairs of nodes, where one node is the point of departure and the other is the destination. Higher length paths are more common within a cluster than between clusters.

$$\mathrm{M}_{\mathrm{inf}}(i, j) = \frac{M(i, j)^r}{\sum_{k=1}^{n} M(k, j)^r} \tag{10}$$

The inflation models the contraction of the flow in equation (10). It becomes thicker in regions of higher value on edges and thinner in regions of lower value on edges. The inflation operation has the effect to rescale each of the columns of *M* with the inflation parameter *r*. The parameter *r* is a non-negative real number. Each column *j* of a Markov matrix *M* is node *j* of the stochastic graph *G*. The inflation operation results in the stochastic matrix M_{inf}. The elements of the matrix M_{inf} correspond to the probability values on the edges. The inflation will boost the probabilities of intra-cluster and dismiss the probabilities of inter-cluster to result in cluster structures. Iterating expansion and inflation result in different clusters of the graph. Therefore, eco-modular architectures are generated based on adjacency matrix from each modular driver.

3.4 Assessment of modular architectures

After determining the eco-modular architectures, an assessment should be made of whether the architectures reflect the intent of the sustainable modular drivers or not. Since modular product architecture is composed of interfaces and modules, we propose two modularity metrics to assess interfaces and modules: (1) module independence (MI) to assess interface complexity between modules; and (2) module similarity (MS) to analyse similarity within modules as shown in Figure 6. The MI is a ratio of interface complexity between modules to total interface complexity of a product in equation (11). The module similarity metric is the degree of similarity within modules in product architecture as shown in equation (12).

The following are assumptions to assess modular architectures with EOL strategies. (1) It is assumed that the MI metric represents the difficulty of disassembly. (2) It is assumed that the MS metric represents possibility of disassembly. This is because a module requires further disassembly at component or material level during product recovery process if modules have components with less similarity of lifespan or material. (3) It is assumed that the lifespan of a component is estimated by the Weibull analysis based on accumulated company data, and the lifespan of the component which is longer than a product lifespan can be performed well for its remaining life.

Figure 6 will be inserted here

MI has been modified from Newcomb et al. (1998). The MI can be calculated by a ratio of intra-module interface complexity for all modules and total interface complexity in equation (11).

$$\text{Module independence (MI)} = \frac{\sum_{i=1}^{N} \text{Intra-module interface complexity}_i}{\text{Total interface complexity}} \tag{11}$$

The MI is calculated by using intra-module interface complexity instead of inter-module interface complexity because it yields that the MI is a maximum value when the interface complexity is low. Theoretically, the maximum value of MI is 1 if all connections are within modules but there are no connections between modules in a product. The minimum value of MI is 0 when the modules are a single component.

Since eco-modular product architectures were identified based on similar values, a designer has to assess the degree of similarity in each module. The MS is assessed by following equation (12).

$$\text{Module similiarity (MS)} = \frac{\sum_{i=1}^{N} Max(S_{i,j})}{N} \tag{12}$$

$$S_{i,j} = \left[\frac{s_{i,1}}{n}, \frac{s_{i,2}}{n}, \ldots, \frac{s_{i,j}}{n}\right], \quad \sum_{j=1}^{n} S_{i,j} = n \tag{13}$$

Where $S_{i,j}$ is the similarity ratio of jth different characteristics to the number of components in ith module (n), N is the number of modules in a product. For example, the first module consists of a component with a lifespan of 5 years ($s_{1,1}$), three components with a lifespan of 8 years ($s_{1,2}$), and two components with a lifespan of 10 years ($s_{1,3}$). In this case, the number of components in the first module is 6 (n=6), and $S_{1,j}$=[1/6, 3/6, 2/6]. As $S_{1,2}$ equals 0.5 which is the maximum value of the first module, it represents a lifespan similarity (S_1) of the first module based on equation (12) and (13). However, $S_{i,j}$=0 when all components have different lifespans. If all modules in a product consist of components with the same materials, MS = 1. Otherwise, it is 0 if each module has components with all different materials. Finally, to consider the MI and MS simultaneously, the modularity for the EOL stage (M_{EOL}) is developed by using the weighted average calculation as shown in equation (14). The modularity ranges from 0 to 1.

$$\textit{Modularity} \text{ for EOL stage } (\mathrm{M}_{EOL}) = w_1 MI + w_2 MS_{Material} + w_3 MS_{Lifespan} \tag{14}$$

$$\sum_{i=1}^{3} w_i = 1 \tag{15}$$

When M_{EOL} is close to 1, the modular architectures can be designed well with considerations of the product recovery. Accordingly, designers can choose an eco-modular architecture of which M_{EOL} is close to 1. As the M_{EOL} measures the modularity only for the EOL stage and is flexible, the M_{EOL} can be modified by adding other similarities that designers wish to measure. Figure 7 shows the proposed four regions with two thresholds (θ_1, θ_2) that indicate decision points for determining categories of a product architecture for EOL recovery options. The value of the thresholds can be decided by design strategies including customer demands for recovered products/modules, recovery cost and profit, and quality of recovered products/modules. The categorized architectures (g) can be represented as equation (16) based on the MI and MS, and provide a guideline for eco-modular architectures in the viewpoint of disassembly for product recovery.

$$g = \begin{cases} \text{A}, & \text{if } MS \geq \theta_1, MI \geq \theta_2 \\ \text{B}, & \text{if } MS \geq \theta_1, MI < \theta_2 \\ \text{C}, & \text{if } MS < \theta_1, MI \geq \theta_2 \\ \text{D}, & \text{if } MS < \theta_1, MI < \theta_2 \end{cases} \tag{16}$$

The four regions of product architecture are proposed based on the MI ($0 \leq \text{MI} \leq 1$) and the MS ($0 \leq \text{MS} \leq 1$): (A) eco-modular architecture, (B) eco-modules with high disassembly cost and time, (C) component reuse or recycle, and (D) disposal of a product as shown in Figure 7 (a). The eco-modular architecture in region A provides efficiency for product recovery process by minimizing the possibility of disassembly operations, and disassembly cost and time. In region B, the eco-modular architecture with high disassembly cost and time provides eco-modules but these connect with high interface complexity. These two eco-modular architectures can be processed for product recovery at the sub-assembly level in Figure 7 (b). However, when the module similarities are low because modules contain components with different lifespans and materials as shown in region C, the modules have to be disassembled into components or materials. Then, components can be recycled or reused if interface complexity between modules is low. In region D, as both the module similarity and independence are low, the product can be disposed of.

Figure 7 will be inserted here

To demonstrate the effectiveness of the proposed eco-modular design and assessment methodology, a case study is performed with a coffee maker in the next section.

4 Case study

In this section, the proposed design methodology is described with a case study of a coffee maker. The electric coffee maker is composed of a simple structure and mechanism but is still efficient. The specification of the coffee maker is described in Table 3. A functional diagram of the coffee maker is shown in Figure 8. The functional diagram represented energy, material, and signal flows between components (Hirtz et al. 2002).

Table 3 Specification of the coffee maker

#	Part name	Material	Weight(g)	Model (Philips HD7450)
1	Bottom cover	PP	80.5	
2	Silicon ring	Silicon	1.8	
3	Hot plate	Al	61.3	
4	Casing for heater	PP	55.2	
5	Heater	Al	118.2	
6	Power cord	Copper	120	
7	Water tube set	PP	14.7	
8	Silicon tube	Silicon	21.4	
9	Water reservoir	PP	243.2	
10	Steam sprout	PP	11.5	
11	Filter basket	PP	79.1	220V, 50Hz, 650W
12	Filter	PP	12.0	4~6 cups / 0.6 L
13	Lid of coffee maker	PP	72.6	Brewing time < 10
14	Decanter	Glass	214.5	minutes
15	Bottom casing	PP	155.0	

Figure 8 will be inserted here

4.1 Product architecture analysis

The physical relationships between components were identified after decomposing the coffee maker. The DSM was applied to represent interrelationships among components as shown in Figure 9. Accordingly, the DSM was used to develop an interface and adjacency matrix shown in Section 4.2. Modules were generated by the DSM clustering algorithm by Thebeau (2001). As shown in Figure 9 and the first column of Table 9, there are five modules: 1) bottom casing; 2) decanter and connecting parts; 3) water reservoir (upper casing) and water tube parts; 4) filter; and 5) heater.

Figure 9 will be inserted here

4.2 Identification of modular drivers for sustainable design

Interface design of the coffee maker was assessed based on equation (1) and the number of interface types and its weight values based on Table 2. As shown in Table 4, direct relationships between any two components were represented by the definition of the interface type. For example, Heater (No.5) and Silicon tube (No.8) are spatially connected from Water reservoir to Heater, and from the Heater to the Steam sprout. So, 'S(2)' means that there are two spatial contacts. As the Heater and the Silicon tube transfer water in the coffee maker, it is also a 'Transfer (T)' interface type. The complexity of interface was developed as the adjacency matrix in Table 5.

Table 4 Interface matrix with interface types

	Interface	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	Bottom cover				S											A(3),S
2	Silicon ring			S	S											
3	Hot plate		S		S	S,F									S,T	S
4	Casing for heater	S	S	S		S										S
5	Heater			S,F	S		P		S(2),T						F	S
6	Power cord					P										S
7	Water tube set								S(2),T(2)							
8	Silicon tube					S(2),T		S(2),T(2)		S,T	S,T					
9	Water reservoir								S,T		S(2)	S		S		A(3),S
10	Steam sprout								S,T	S(2)		T		S(2)		
11	Filter basket									S	T		S,T		S,T	
12	Filter											S,T				
13	Lid of coffee maker									S	S(2)					S(2)
14	Decanter			S,T		F						S,T				S
15	Bottom casing	A(3),S		S	S	S	S			A(3),S				S(2)	S	

Table 5 Adjacency matrix with interface values

	Interface	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	Bottom cover				1.795											10.717
2	Silicon ring			1.795	1.795											
3	Hot plate		1.795		1.795	11.422									11.240	1.795
4	Casing for heater	1.795	1.795	1.795		1.795										1.795
5	Heater			11.422	1.795		6.832		13.035						9.627	1.795
6	Power cord					6.832										1.795
7	Water tube set								22.480							
8	Silicon tube					13.035		22.480		11.240	11.240					
9	Water reservoir								11.240		3.590	1.795		1.795		10.717
10	Steam sprout								11.240	3.590		9.445		3.590		
11	Filter basket									1.795	9.445		11.240		11.240	
12	Filter											11.240				
13	Lid of coffee maker									1.795	3.590					3.590
14	Decanter			11.240		9.627						11.240				1.795
15	Bottom casing	10.717		1.795	1.795	1.795	1.795			10.717				3.590	1.795	

To build the adjacency matrix for the clustering of modules, the information of each component was analysed as shown in Table 6. The typical lifespan of the drip filter coffee machine was approximately 6 years and was approved by the European Committee of Domestic Equipment Manufacturers and other stakeholders (European Commission 2011). In the case of the product lifespan of components, as the coffee maker has a simple function and operating conditions, the lifespan of components would then only largely be affected by material types. Accordingly, we assumed an average lifespan of each component based on material types. The lifespan of components made of aluminum, copper, and glass was assumed as 10 years, and the lifespan of components made of PP or silicon was assumed as 8 or 5 years, respectively in Table 6.

Table 6 Material types and estimated lifespan of the coffee maker

#	Part name	Material	Typical product lifespan	Estimated component lifespan
1	Bottom cover	PP	6	8
2	Silicon ring	Silicon		5
3	Hot plate	Al		10
4	Casing for heater	PP		8
5	Heater	Al		10
6	Power cord	Copper		10
7	Water tube set	PP		8
8	Silicon tube	Silicon		5
9	Water reservoir	PP		8
10	Steam sprout	PP		8
11	Filter basket	PP		8
12	Filter	PP		8
13	Lid of coffee maker	PP		8
14	Decanter	Glass		10
15	Bottom casing	PP		8

Based on the product information, similarities of components were measured by equations (6) and (7). Similarity matrices which are the same as adjacency matrices were developed to represent material similarity and lifespan similarity as shown in Table 7.

Table 7 Adjacency matrix for similarities of (a) material and (b) lifespan

(a)

	Material similarity	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	Bottom cover				1.00											1.00
2	Silicon ring			0.30	0.50											
3	Hot plate		0.30		0.30	1.00									0.30	0.30
4	Casing for heater	1.00	0.50	0.30		0.30										1.00
5	Heater			1.00	0.30		0.50		0.30						0.30	0.30
6	Power cord					0.50										0.30
7	Water tube set								0.50							
8	Silicon tube					0.30		0.50		0.50	0.50					
9	Water reservoir								0.50		1.00	1.00		1.00		1.00
10	Steam sprout								0.50	1.00		1.00		1.00		
11	Filter basket									1.00	1.00		1.00		0.30	
12	Filter											1.00				
13	Lid of coffee maker									1.00	1.00					1.00
14	Decanter			0.30		0.30						0.30				0.30
15	Bottom casing	1.00		0.30	1.00	0.30	0.30			1.00				1.00	0.30	

(b)

	Lifespan similarity	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	Bottom cover				1.00											1.00
2	Silicon ring			0.50	0.63											
3	Hot plate		0.50		0.80	1.00									1.00	0.80
4	Casing for heater	1.00	0.63	0.80		0.80										1.00
5	Heater			1.00	0.80		1.00		0.50						1.00	0.80
6	Power cord					1.00										0.80
7	Water tube set								0.63							
8	Silicon tube					0.50		0.63		0.63	0.63					
9	Water reservoir								0.63		1.00	1.00		1.00		1.00
10	Steam sprout								0.63	1.00		1.00		1.00		
11	Filter basket									1.00	1.00		1.00		0.80	
12	Filter											1.00				
13	Lid of coffee maker									1.00	1.00					1.00
14	Decanter			1.00		1.00						0.80				0.80
15	Bottom casing	1.00		0.80	1.00	0.80	0.80			1.00				1.00	0.80	

4.3 Identifying modules based on the Markov Cluster Algorithm

Based on adjacency matrices, the Markov matrix (M) was determined by equation (9) as shown in Table 8. Eco-modules were determined by using Markov Cluster Algorithm (MCL). Since the proposed methodology arranged constituents in modules while keeping the physical relationships, it could result in feasible eco-module architectures. In the case of the optimal number of modules, we assumed that the number of modules for the coffee maker was decided as five modules to compare alternatives of eco-modular architectures. Accordingly, alternatives of eco-modular architectures were identified based on physical relationship, interface complexity, material similarity, lifespan similarity, and weighted average of these three modular drivers by using MCL as shown in Table 9. The module number in Table 9 did not mean that the same number of modules for each architecture had the same characteristics, but showed that each architecture had five modules.

Table 8 Markov matrices for interface complexity, material similarity, and lifespan similarity

$$M_{\text{Interface}} = \begin{bmatrix}
0.00 & 0.00 & 0.00 & 0.20 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.32 \\
0.00 & 0.00 & 0.06 & 0.20 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 \\
0.00 & 0.50 & 0.00 & 0.20 & 0.26 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.33 & 0.05 \\
0.14 & 0.50 & 0.06 & 0.00 & 0.04 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.05 \\
0.00 & 0.00 & 0.41 & 0.20 & 0.00 & 0.79 & 0.00 & 0.22 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.28 & 0.05 \\
0.00 & 0.00 & 0.00 & 0.00 & 0.15 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.05 \\
0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.39 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 \\
0.00 & 0.00 & 0.00 & 0.00 & 0.29 & 0.00 & 1.00 & 0.00 & 0.39 & 0.40 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 \\
0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.19 & 0.00 & 0.13 & 0.05 & 0.00 & 0.20 & 0.00 & 0.32 \\
0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.19 & 0.12 & 0.00 & 0.28 & 0.00 & 0.40 & 0.00 & 0.00 \\
0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.06 & 0.34 & 0.00 & 1.00 & 0.00 & 0.33 & 0.00 \\
0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.33 & 0.00 & 0.00 & 0.00 & 0.00 \\
0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.06 & 0.13 & 0.00 & 0.00 & 0.00 & 0.00 & 0.11 \\
0.00 & 0.00 & 0.40 & 0.00 & 0.22 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.33 & 0.00 & 0.00 & 0.00 & 0.05 \\
0.86 & 0.00 & 0.06 & 0.20 & 0.04 & 0.21 & 0.00 & 0.00 & 0.37 & 0.00 & 0.00 & 0.00 & 0.40 & 0.05 & 0.00
\end{bmatrix}$$

$$M_{\text{Material}} = \begin{bmatrix}
0.00 & 0.00 & 0.00 & 0.32 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.19 \\
0.00 & 0.00 & 0.14 & 0.16 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 \\
0.00 & 0.38 & 0.00 & 0.10 & 0.37 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.25 & 0.06 \\
0.50 & 0.63 & 0.14 & 0.00 & 0.11 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.19 \\
0.00 & 0.00 & 0.45 & 0.10 & 0.00 & 0.63 & 0.00 & 0.17 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.25 & 0.06 \\
0.00 & 0.00 & 0.00 & 0.00 & 0.19 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.06 \\
0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.28 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 \\
0.00 & 0.00 & 0.00 & 0.00 & 0.11 & 0.00 & 1.00 & 0.00 & 0.11 & 0.14 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 \\
0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.28 & 0.00 & 0.29 & 0.30 & 0.00 & 0.33 & 0.00 & 0.19 \\
0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.28 & 0.22 & 0.00 & 0.30 & 0.00 & 0.33 & 0.00 & 0.00 \\
0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.22 & 0.29 & 0.00 & 1.00 & 0.00 & 0.25 & 0.00 \\
0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.30 & 0.00 & 0.00 & 0.00 & 0.00 \\
0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.22 & 0.29 & 0.00 & 0.00 & 0.00 & 0.00 & 0.19 \\
0.00 & 0.00 & 0.14 & 0.00 & 0.11 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.09 & 0.00 & 0.00 & 0.00 & 0.06 \\
0.50 & 0.00 & 0.14 & 0.32 & 0.11 & 0.38 & 0.00 & 0.00 & 0.22 & 0.00 & 0.00 & 0.00 & 0.33 & 0.25 & 0.00
\end{bmatrix}$$

$$M_{\text{Lifespan}} = \begin{bmatrix}
0.00 & 0.00 & 0.00 & 0.24 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.14 \\
0.00 & 0.00 & 0.12 & 0.15 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 \\
0.00 & 0.44 & 0.00 & 0.19 & 0.20 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.28 & 0.11 \\
0.50 & 0.56 & 0.20 & 0.00 & 0.16 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.14 \\
0.00 & 0.00 & 0.24 & 0.19 & 0.00 & 0.56 & 0.00 & 0.21 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.28 & 0.11 \\
0.00 & 0.00 & 0.00 & 0.00 & 0.20 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.11 \\
0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.26 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 \\
0.00 & 0.00 & 0.00 & 0.00 & 0.10 & 0.00 & 1.00 & 0.00 & 0.14 & 0.17 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 \\
0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.26 & 0.00 & 0.28 & 0.26 & 0.00 & 0.33 & 0.00 & 0.14 \\
0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.26 & 0.22 & 0.00 & 0.26 & 0.00 & 0.33 & 0.00 & 0.00 \\
0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.22 & 0.28 & 0.00 & 1.00 & 0.00 & 0.22 & 0.00 \\
0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.26 & 0.00 & 0.00 & 0.00 & 0.00 \\
0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.22 & 0.28 & 0.00 & 0.00 & 0.00 & 0.00 & 0.14 \\
0.00 & 0.00 & 0.24 & 0.00 & 0.20 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.21 & 0.00 & 0.00 & 0.00 & 0.11 \\
0.50 & 0.00 & 0.20 & 0.24 & 0.16 & 0.44 & 0.00 & 0.00 & 0.22 & 0.00 & 0.00 & 0.00 & 0.33 & 0.22 & 0.00
\end{bmatrix}$$

Table 9 Clustering results based on MCL

Module No.	DSM clustering	MCL				
	Architecture (A) (based on physical relationship)	Architecture (B) (based on physical relationship)	Architecture (C) (based on interface complexity)	Architecture (D) (based on material similarity)	Architecture (E) (based on lifespan similarity)	Architecture (F) (based on weighted average)
1	1 4 15	1 2 3 4 6 14 15	1 4 13 15	1 2 4 15	1 2 3 4 6 15	1 2 4 13 15
2	2 3 14	5	2 3 5 6	3 5 6 14	5 14	3 5 6 14
3	7 8 9 10 13	7 8	7 8 9 10	7 8	7 8	7 8
4	11 12	11 12	11 12	11 12	11 12	11 12
5	5 6	9 10 13	14	9 10 13	9 10 13	9 10

4.4 Assessment of modular architecture

The modularity for the EOL stage (M_{EOL}) was applied to assess each module and architecture. Additionally, the architecture (A) developed by DSM clustering algorithm in Table 9 was also assessed to determine the degree of modularity for product recovery.

From the results of the module independence assessment, architectures (A) and (D) had the highest value with 0.54, while architecture (E) had the lowest value with 0.41 as shown in Table 10. This means that architectures (A) and (D) contain low interface complexity between the modules. From the results of the material similarity assessment, architecture (C) in Table 10 had the highest similarity with 0.85 because all modules contained components with a similar or the same materials, except for the second module. In architecture (C) in Table 9, the second module inevitably contained components with different materials, such as the hot plate (3), heater (5), power cord (6), and decanter (14), to meet the functionality of the product. For results of the lifespan similarity assessment, architecture (C) was also the best architecture with the highest lifespan similarity value, 0.9; this was because almost all the modules contained components with a similar or same lifespan.

Table 10 Modularity assessment results

	MI	MS_material	MS_lifespan	M_{EOL}	MDL
Architecture A	0.54	0.56	0.89	0.66	13.56
Architecture B	0.43	0.69	0.69	0.60	25.12
Architecture C	0.53	0.85	0.90	0.76	24.45
Architecture D	0.54	0.65	0.75	0.64	21.78
Architecture E	0.41	0.50	0.70	0.54	23.12
Architecture F	0.53	0.66	0.76	0.65	24.45

Based on the individual results from the module independence, material similarity, and lifespan similarity, we represented the results on a scatter plot of categorized modular architectures in Figure 10. Two thresholds could be decided to identify four regions of the architectures according to design strategies of a company including customer demands for recovered products, recovery profit, and so on. In this case study, the two thresholds were assumed as (θ_1, θ_2) = (0.7, 0.5). Region A contained architectures (A), (C), (D), and (F) with lifespan similarity. However, as the material similarity of theses architectures was lower than θ_1, except for architecture (C), these architectures had to be disassembled at the component level for recycling. Therefore, in this case study, architecture (C) would be the best architecture due to high module similarities within modules and high module independence between modules shown in Figure 10. Also, the M_{EOL} showed that architecture (C) was the best eco-modular architecture with the highest M_{EOL} value in Table 10. For architectures (B) and (E), the designer can plan to dispose of products because product recovery cost and time will be high. The proposed methodology was developed based on physical interrelationships in a product and resulted in the eco-modular architecture illustrated in the functional diagram of architecture (C) in Figure 11. Accordingly, architecture (C) will maintain its functions and enhance the product recovery process. Architecture (C) of this coffee maker could be recycled and reused without further disassembling the modules into components or materials. The characteristics of modules 1, 4, and 5 in architecture (C) could lead to reuse and recycle options because the modules consisted of components with the same material and lifespan. However, as modules 2 and 3 contained components with different materials and lifespans, these modules would have to be disassembled at the component or material levels for product recovery.

Figure 10 will be inserted here

Figure 11 will be inserted here

In order to compare a conventional method to the proposed assessment method, we used the minimum description length (MDL) to measure modularity by calculating the coupling complexity of the coffee maker (Chiriac et al. 2011) as follows:

$$MDL = \frac{1}{3}\left(n_c \log_2 n_n + \log_2 n_n \sum_{i=1}^{n_c} cl_i \right) + \frac{1}{3} S_1 + \frac{1}{3} S_2 \qquad (17)$$

where, n_c is the number of modules; n_n is the number of components in DSM; and cl_i is the number of components in module *i*. S_1 is the number of empty cells in a module, while S_2 is the number of cells that have a link between modules. The lower the MDL, the better the modularity of a product. When calculating the MDL, values from the proposed modular drivers were not considered, but the number and size of modules and components were considered. By using the theoretic information, we could identify well-established modules according to the ideal modularity concept that had high internal couplings within modules and low external couplings between modules. In terms of the result from the MDL calculation in Table 10, architecture (A)'s modularity was the best architecture. This was because architecture (A) had less empty cells within modules than other architectures. From the results of Table 10, we could see that different viewpoints of modularity assessments resulted in different priorities of the architectures. With regard to the ideal modularity concept, architecture (A) had the best modules, while architecture (C) had good modularity for the EOL stage. It was difficult to compare the MDL and M_{EOL} because of the different purposes of the modularity. However, the M_{EOL} would be a good starting point to assess modularity for sustainable product design.

Additionally, after deciding the eco-modular product architecture, design improvement in a module level would be needed to produce various product variants including environmentally-friendly products. So, lifecycle assessment was applied to measure the environmental impact of the modules and to redesign the modules regarding material usage. Eco-Indicator 99 was used as the lifecycle assessment tool (Baayen 2000). The Eco-Indicator 99 can measure three types of damage: 1) human health; 2) ecosystem quality (e.g. acidification); and 3) resources.

Table 11 Environmental impact changes after material reduction

Module No.	Part No.	Part name	Material	Weight (g)	EI (mPt)	Weight change (g)	EI change (mPt)
1	1	Bottom cover	PP	80.5	11.350	-10	-1.41
	4	Casing for heater	PP	55.2	7.783	0	0
	13	Lid of coffee maker	PP	72.6	10.237	0	0
	15	Bottom casing	PP	155.0	21.855	-25	-3.53
2	2	Silicon ring	Silicon	1.8	0.178	0	0
	3	Hot plate	Al	61.3	6.183	0	0
	5	Heater	Al	118.2	11.922	-18.2	-2.40
	6	Power cord	Copper	120.0	168.240	0	0
3	7	Water tube set	PP	14.7	2.073	0	0
	8	Silicon tube	Silicon	21.4	2.119	0	0
	9	Water reservoir	PP	243.2	34.291	-43.2	-6.09
	10	Steam sprout	PP	11.5	1.621	0	0
4	11	Filter basket	PP	79.1	11.153	0	0
	12	Filter	PP	12.0	1.692	0	0
5	14	Decanter	Glass	214.5	6.435	-14.5	-0.44
Sum				**1261**	**294.08**	**-100.9**	**-13.87**

A scenario was assumed to enhance the product design to include product recovery. For the product recovery, material selection plays a key role in eco-product design because the materials not only can affect the health of consumers but also affect the depletion of natural resources through overconsumption. The material selection relies on expert knowledge in terms of material property, manufacturing process, and safety for users. In this case study, as almost all materials in the coffee maker were recyclable, we assumed that the bottom cover and bottom casing, heater, water reservoir, and decanter were redesigned to reduce material usage by minimizing their size and thickness. Accordingly, the weight was reduced as shown in Table 11. As the result, the weight changes of the materials decreased the environmental impacts after re-design.

5 Closing remarks and future work

In this research, we introduced a methodology to determine the eco-modular product architecture and its modularity assessment for product recovery processes. The proposed methodology contributes to facilitating the product recovery processes by recycling, reusing, and disposing modules without further disassembly at the component or material levels. Also, the methodology helps designers quantify the degree of modularity of the eco-modular architecture from the viewpoint of the end-of-life stage.

In this research, modular product design was extended to consider the product recovery processes which aim to restore the inherent value of retired products. Three sustainable modular drivers were proposed to consolidate components with high similarity and interface complexity into modules, aiming to recover the modules without further disassembly at component or material levels. Markov Cluster Algorithm was applied to identify alternative eco-modular architectures based on these the sustainable modular drivers. To check whether or not the eco-modular architectures reflect the intent of the sustainable modular drivers, a novel methodology has been developed to assess the degree of modularity from the viewpoint of product recovery. The degree of modularity supported the selection of the eco-modular architecture based on the module independence metric between modules and the module similarity metric. To demonstrate the usefulness of the proposed methodology, we performed a case study of a coffee maker.

The limitation of this proposed methodology is that it considers modular architecture for only a single product. More case studies need to be performed with different products to show the scalability and applicability of the proposed methodology. Also, the methodology will be extended to product family architecture design to improve the reusability of the eco-modules by sharing and exchanging the eco-modules in a product family.

Acknowledgements

This work was supported by an AcRF Tier 1 grant (RG94/13) from Ministry of Education, Singapore.

References

Asikoglu, O., & Simpson, T. W. (2012). *A new method for evaluating design dependencies in product architectures.* Paper presented at the 12th AIAA Aviation Technology, Integration, and Operations (ATIO) Conference and 14th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference Indianapolis, Indiana, USA, September 17-19, 2012.

Baayen, H. (2000). Eco-indicator 99 manual for designers. https://www.pre-sustainability.com/download/EI99_Manual.pdf. Accessed 1 December 2015. Ministry of Housing, Spatial Planning and the Environment, Hague, Netherlands.

Behdad, S., Berg, L. P., Thurston, D., & Vance, J. (2014). Leveraging virtual reality experiences with mixed-integer nonlinear programming visualization of disassembly sequence planning under uncertainty. *Journal of Mechanical Design, 136*(4), 041005, doi:10.1115/1.4026463.

Behdad, S., Kwak, M., Kim, H., & Thurston, D. (2010). Simultaneous selective disassembly and end-of-life decision making for multiple products that share disassembly operations. *Journal of Mechanical Design, 132*(4), 041002-041002, doi:10.1115/1.4001207.

Bernardes, J. S., Vieira, F. R., Costa, L. M., & Zaverucha, G. (2015). Evaluation and improvements of clustering algorithms for detecting remote homologous protein families. *BMC Bioinformatics, 16*(1), 1-14, doi:10.1186/s12859-014-0445-4.

Borjesson, F., & Hölttä-Otto, K. (2014). A module generation algorithm for product architecture based on component interactions and strategic drivers. *Research in Engineering Design, 25*(1), 31-51, doi:10.1007/s00163-013-0164-2.

Chiriac, N., Hölttä-Otto, K., Lysy, D., & Suh, E. S. (2011). Level of modularity and different levels of system granularity. *Journal of Mechanical Design, 133*(10), 101007.

Chun-Che, H., & Kusiak, A. (1998). Modularity in design of products and systems. *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on, 28*(1), 66-77, doi:10.1109/3468.650323.

Dahmus, J. B., Gonzalez-Zugasti, J. P., & Otto, K. N. (2001). Modular product architecture. *Design Studies, 22*(5), 409-424, doi:http://dx.doi.org/10.1016/S0142-694X(01)00004-7.

Dobberfuhl, A., & Lange, M. W. (2009). *Interfaces per module: Is there an ideal number?* Paper presented at the ASME 2009 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, San Diego, California, USA, August 30-September 2, 2009.

Du, G., Jiao, R. J., & Chen, M. (2014). Joint optimization of product family configuration and scaling design by Stackelberg game. *European Journal of Operational Research, 232*(2), 330-341, doi:http://dx.doi.org/10.1016/j.ejor.2013.07.021.

Erixon, G. (1996). *Modular Function Development MFD, Support for Good Product Structure Creation.* Paper presented at the DS 53: Proceedings of the 2nd WDK Workshop on Product Structuring Delft University of Technology, Netherlands, June 03-04

Erixon, G. (1998). *Modular Function Deployment - A method for product modularisation*. PhD thesis, The Royal Institute of Technology, Sweden.

European Commission (2011). Preparatory studies for ecodesign requirements of EuPs (III): Lot 25: Non-tertiary coffee machines, Task 2: Economic and market analysis.

Fixson, S. K. (2005). Product architecture assessment: a tool to link product, process, and supply chain design decisions. *Journal of Operations Management, 23*(3-4), 345-369, doi:10.1016/j.jom.2004.08.006.

Fujita, K., & Yoshida, H. (2004). Product Variety Optimization Simultaneously Designing Module Combination and Module Attributes. *Concurrent Engineering, 12*(2), 105-118, doi:10.1177/1063293x04044758.

GÜngÖr, A., & Gupta, S. M. (2001). Disassembly sequence plan generation using a branch-and-bound algorithm. *International Journal of Production Research, 39*(3), 481-509, doi:10.1080/00207540010002838.

Helmer, R., Yassine, A., & Meier, C. (2010). Systematic module and interface definition using component design structure matrix. *Journal of Engineering Design, 21*(6), 647-675, doi:10.1080/09544820802563226.

Hirtz, J., Stone, R. B., McAdams, D. A., Szykman, S., & Wood, K. L. (2002). A functional basis for engineering design: Reconciling and evolving previous efforts. *Research in Engineering Design, 13*(2), 65-82, doi:10.1007/s00163-001-0008-3.

Hölttä, K. M. M., & Otto, K. N. (2005). Incorporating design effort complexity measures in product architectural design and assessment. *Design Studies, 26*(5), 463-485, doi:http://dx.doi.org/10.1016/j.destud.2004.10.001.

Ilgin, M. A., & Gupta, S. M. (2010). Environmentally conscious manufacturing and product recovery (ECMPRO): A review of the state of the art. *Journal of Environmental Management, 91*(3), 563-591, doi:10.1016/j.jenvman.2009.09.037.

Ishii, K., Eubanks, C. F., & Di Marco, P. (1994). Design for product retirement and material life-cycle. *Materials & Design, 15*(4), 225-233, doi:http://dx.doi.org/10.1016/0261-3069(94)90007-8.

Jose, A., & Tollenaere, M. (2005). Modular and platform methods for product family design: literature analysis. *Journal of Intelligent Manufacturing, 16*(3), 371-390, doi:10.1007/s10845-005-7030-7.

Jung, S., & Simpson, T. W. (2014). *A Clustering Method Using New Modularity Indices and Genetic Algorithm with Extended Chromosomes.* Paper presented at the DSM 14 Proceedings of the 16th International DSM conference: Risk and Change management in complex systems, Paris, France, July 2-4, 2014.

Jung, S., Simpson, T. W., & Asikoglu, O. (2014). *Using interfaces to drive module definition: investigating the impact of a new design dependency measure.* Paper presented at the ASME 2014 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference (DETC2014-34555), Buffalo, New York, USA, August 17–20, 2014.

Kara, S., Mazhar, M., Kaebernick, H., & Ahmed, A. (2005). Determining the Reuse Potential of Components Based on Life Cycle Data. *CIRP Annals - Manufacturing Technology, 54*(1), 1-4, doi:http://dx.doi.org/10.1016/S0007-8506(07)60036-5.

Kim, S., Baek, J. W., Moon, S. K., & Jeon, S. M. (2015). A New Approach for Product Design by Integrating Assembly and Disassembly Sequence Structure Planning. In *Proceedings of the 18th Asia Pacific Symposium on Intelligent and Evolutionary Systems* (Vol. 1, pp. 247-257): Springer International Publishing Switzerland.

Krause, D., Beckmann, G., Eilmus, S., Gebhardt, N., Jonas, H., & Rettberg, R. (2014). Integrated Development of Modular Product Families: A Methods Toolkit. In T. W. Simpson, J. Jiao, Z. Siddique, & K. Hölttä-Otto (Eds.), *Advances in Product Family and Product Platform Design* (pp. 245-269): Springer New York.

Kwak, M. J., Hong, Y. S., & Cho, N. W. (2009). Eco-architecture analysis for end-of-life decision making. *International Journal of Production Research, 47*(22), 6233-6259, doi:10.1080/00207540802175329.

Lambert, A. J. D. (2002). Determining optimum disassembly sequences in electronic equipment. *Computers & Industrial Engineering, 43*(3), 553-575, doi:http://dx.doi.org/10.1016/S0360-8352(02)00125-0.

Lei, X., Wang, F., Wu, F.-X., Zhang, A., & Pedrycz, W. (2016). Protein complex identification through Markov clustering with firefly algorithm on dynamic protein–protein interaction networks. *Information Sciences, 329*(0), 303-316, doi:http://dx.doi.org/10.1016/j.ins.2015.09.028.

Lindahl, M., Sundin, E., Östlin, J., & Björkman, M. (2006). Concepts and definitions for product recovery Analysis and clarification of the terminology used in academia and industry. In *Innovation in Life Cycle Engineering and Sustainable Development* (pp. 123-138): Springer Netherlands.

Ljungberg, L. Y. (2007). Materials selection and design for development of sustainable products. *Materials & Design, 28*(2), 466-479, doi:http://dx.doi.org/10.1016/j.matdes.2005.09.006.

Mazhar, M. I., Kara, S., & Kaebernick, H. (2007). Remaining life estimation of used components in consumer products: Life cycle data analysis by Weibull and artificial neural networks. *Journal of Operations Management, 25*(6), 1184-1193, doi:http://dx.doi.org/10.1016/j.jom.2007.01.021.

Meng, X., Jiang, Z., & Huang, G. Q. (2007). On the module identification for product family development. *The International Journal of Advanced Manufacturing Technology, 35*(1-2), 26-40, doi:10.1007/s00170-006-0712-2.

Moon, S. K., & McAdams, D. A. (2012). A Market-Based Design Strategy for a Universal Product Family. *Journal of Mechanical Design, 134*(11), 111007, doi:10.1115/1.4007395.

Moon, S. K., Park, K. J., & Simpson, T. W. (2014). Platform design variable identification for a product family using multi-objective particle swarm optimization. *Research in Engineering Design, 25*(2), 95-108, doi:10.1007/s00163-013-0166-0.

Moon, S. K., Simpson, T. W., & Kumara, S. R. T. (2010). A methodology for knowledge discovery to support product family design. *Annals of Operations Research, 174*(1), 201-218, doi:10.1007/s10479-008-0349-7.

Newcomb, P. J., Bras, B., & Rosen, D. W. (1998). Implications of modularity on product design for the life cycle. *Journal of Mechanical Design, 120*(3), 483-490.

Otto, K. N., & Wood, K. L. (2001). *Product design : techniques in reverse engineering and new product development*: Upper Saddle River, NJ : Prentice Hall.

Pimmler, T. U., & Eppinger, S. D. (1994). *Integration analysis of product decompositions.* Paper presented at the Proceedings of the ASME Design Engineering Technical Conferences - 6th International Conference on Design Theory and Methodology Minneapolis, MN, USA, 1994.

Ramani, K., Ramanujan, D., Bernstein, W. Z., Zhao, F., Sutherland, J., Handwerker, C., et al. (2010). Integrated Sustainable Life Cycle Design: A Review. *Journal of Mechanical Design, 132*(9), 091004, doi:10.1115/1.4002308.

Smith, S., & Chen, W.-H. (2011). Rule-based recursive selective disassembly sequence planning for green design. *Advanced Engineering Informatics, 25*(1), 77-87, doi:10.1016/j.aei.2010.03.002.

Smith, S., Smith, G., & Chen, W.-H. (2012). Disassembly sequence structure graphs: An optimal approach for multiple-target selective disassembly sequence planning. *Advanced Engineering Informatics, 26*(2), 306-316, doi:http://dx.doi.org/10.1016/j.aei.2011.11.003.

Sosa, M. E., Eppinger, S. D., & Rowles, C. M. (2003). Identifying Modular and Integrative Systems and Their Impact on Design Team Interactions. *Journal of Mechanical Design, 125*(2), 240-252, doi:10.1115/1.1564074.

Thebeau, R. E. (2001). *Knowledge management of system interfaces and interactions from product development processes*. M.S. Thesis, Massachusetts Institute of Technology, USA.

Thierry, M., Salomon, M., van Nunen, J., & van Wassenhove, L. (1995). Strategic Issues in Product Recovery Management. *California Management Review, 37*(2), 114-135.

Tseng, H.-E., Chang, C.-C., & Li, J.-D. (2008). Modular design to support green life-cycle engineering. *Expert Systems with Applications, 34*(4), 2524-2537, doi:http://dx.doi.org/10.1016/j.eswa.2007.04.018.

Ulrich, K. T. (1995). The role of product architecture in the manufacturing firm. *Research Policy, 24*(3), 419-440, doi:http://dx.doi.org/10.1016/0048-7333(94)00775-3.

Ulrich, K. T., & Eppinger, S. D. (2008). *Product design and development* Boston : McGraw-Hill Higher Education, c2008. 4th ed.

Van Dongen, S. M. (2000). *Graph clustering by flow simulation*. Ph.D. Thesis, Utrecht University, Nederlands.

Yan, J., Feng, C., & Cheng, K. (2012). Sustainability-oriented product modular design using kernel-based fuzzy c-means clustering and genetic algorithm. *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture, 226*(10), 1635-1647.

Yigit, A. S., Ulsoy, A. G., & Allahverdi, A. Optimizing modular product design for reconfigurable manufacturing. [journal article]. *Journal of Intelligent Manufacturing, 13*(4), 309-316, doi:10.1023/a:1016032714680.

Yu, T.-L., Yassine, A., & Goldberg, D. (2007). An information theoretic method for developing modular architectures using genetic algorithms. *Research in Engineering Design, 18*(2), 91-109, doi:10.1007/s00163-007-0030-1.

List of Figures

Figure 1 The proposed process for eco-modular architecture design

Figure 2 An example of product architecture

Figure 3 Three sustainable modular drivers

Figure 4 An example of DSM with (a) physical relationship and (b) wMCS

Figure 5 Clustering process for identifying the eco-modules by the MCL

Figure 6 The assessment range of the proposed modularity metrics

Figure 7 Modular architecture categories and their product recovery options in product lifecycle

Figure 8 Functional diagram of the coffee maker

Figure 9 Modules of the coffee maker identified by DSM clustering algorithm

Figure 10 Categorized modular product architectures of the coffee maker in four regions according to the module independence and module similarity metrics

Figure 11 The eco-modular product architecture (C)

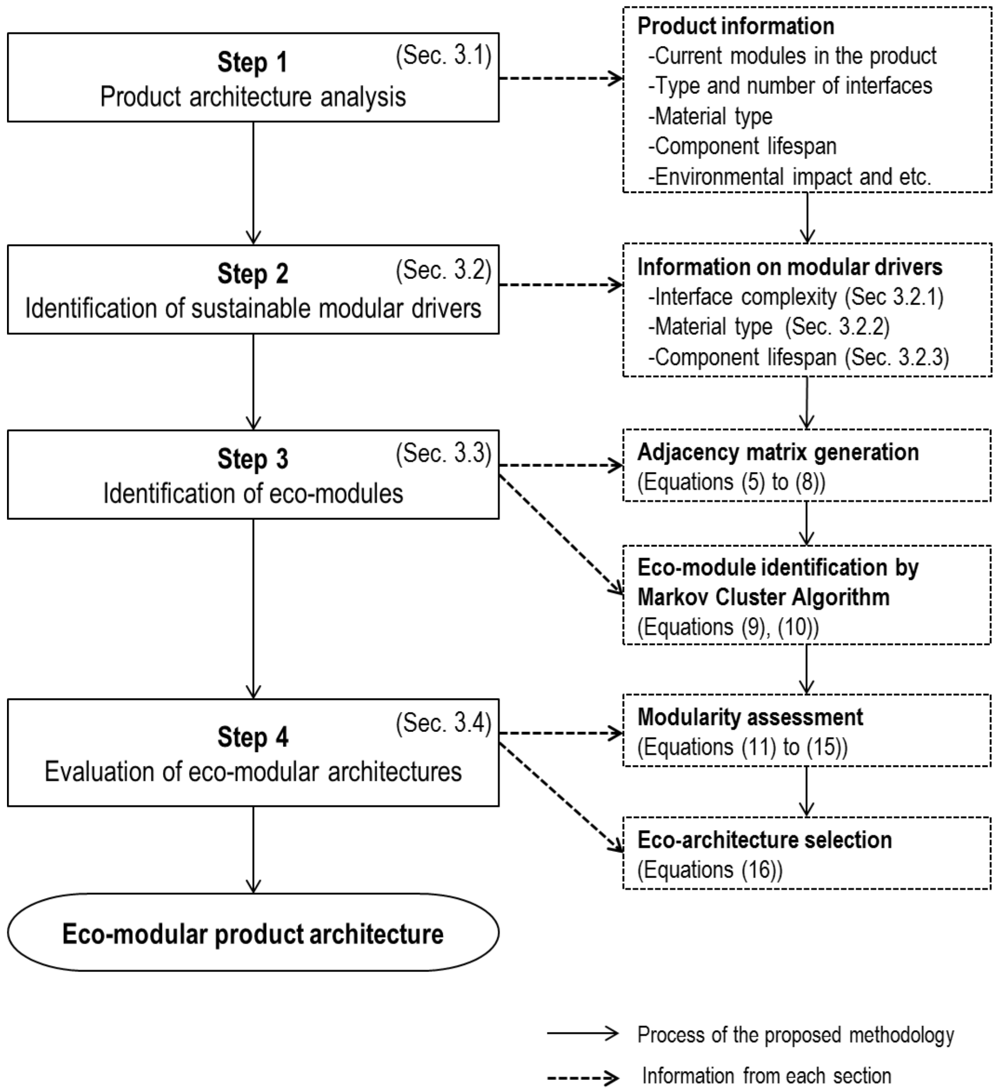


Figure 1 The proposed process for eco-modular architecture design

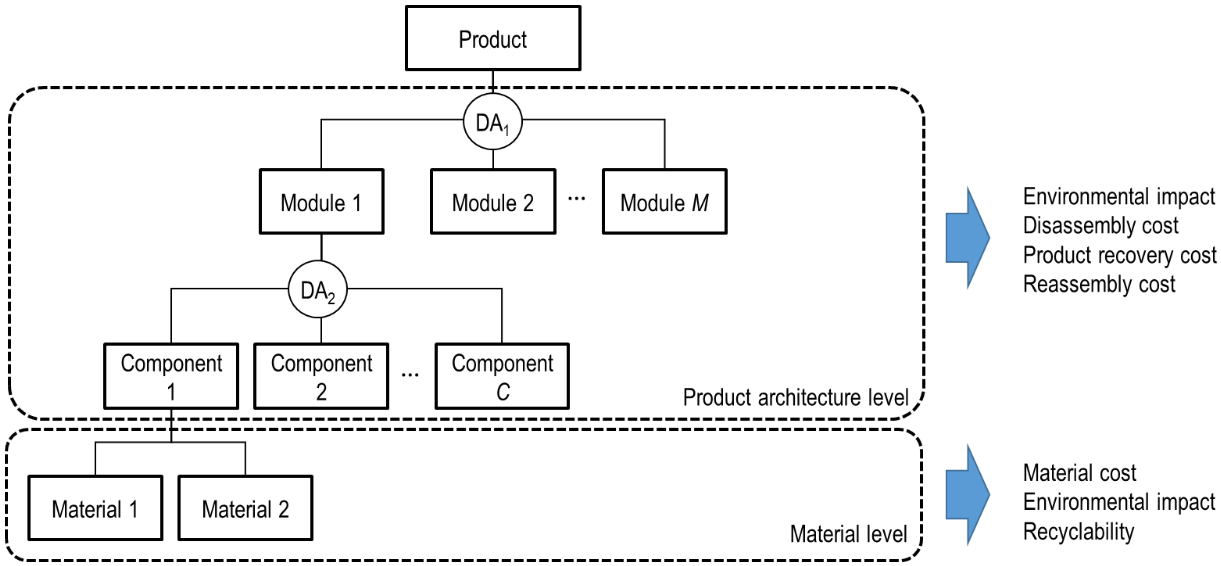


Figure 2 An example of product architecture

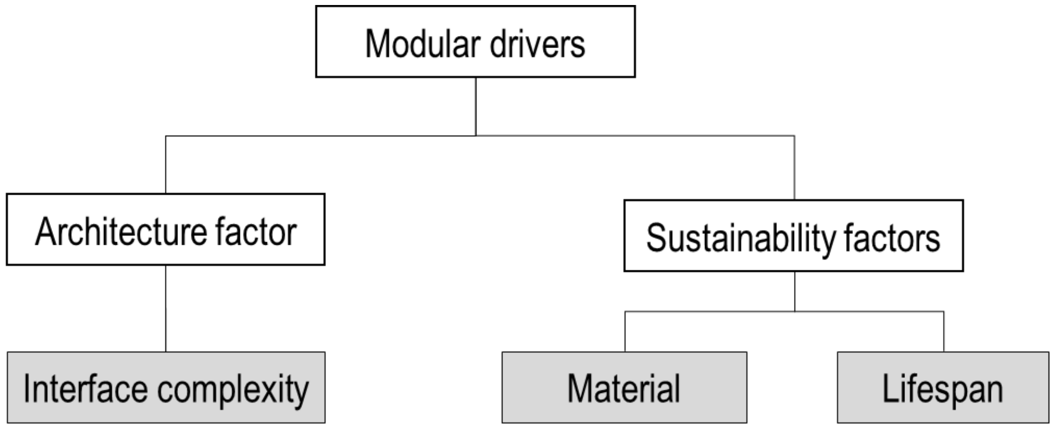


Figure 3 Three sustainable modular drivers

	A	B	C	D	E
A		1	1		
B	1		1		
C	1	1		1	
D			1		1
E				1	

(a)

	A	B	C	D	E
A		A	A(2),S		
B	A		AS		
C	A(2),S	AS		S	
D			S		A
E				A	

	A	B	C	D	E
A		2.974	7.743		
B	2.974		9.538		
C	7.743	9.538		1.795	
D			1.795		2.974
E				2.974	

(b)

Figure 4 An example of DSM with (a) physical relationship and (b) wMCS

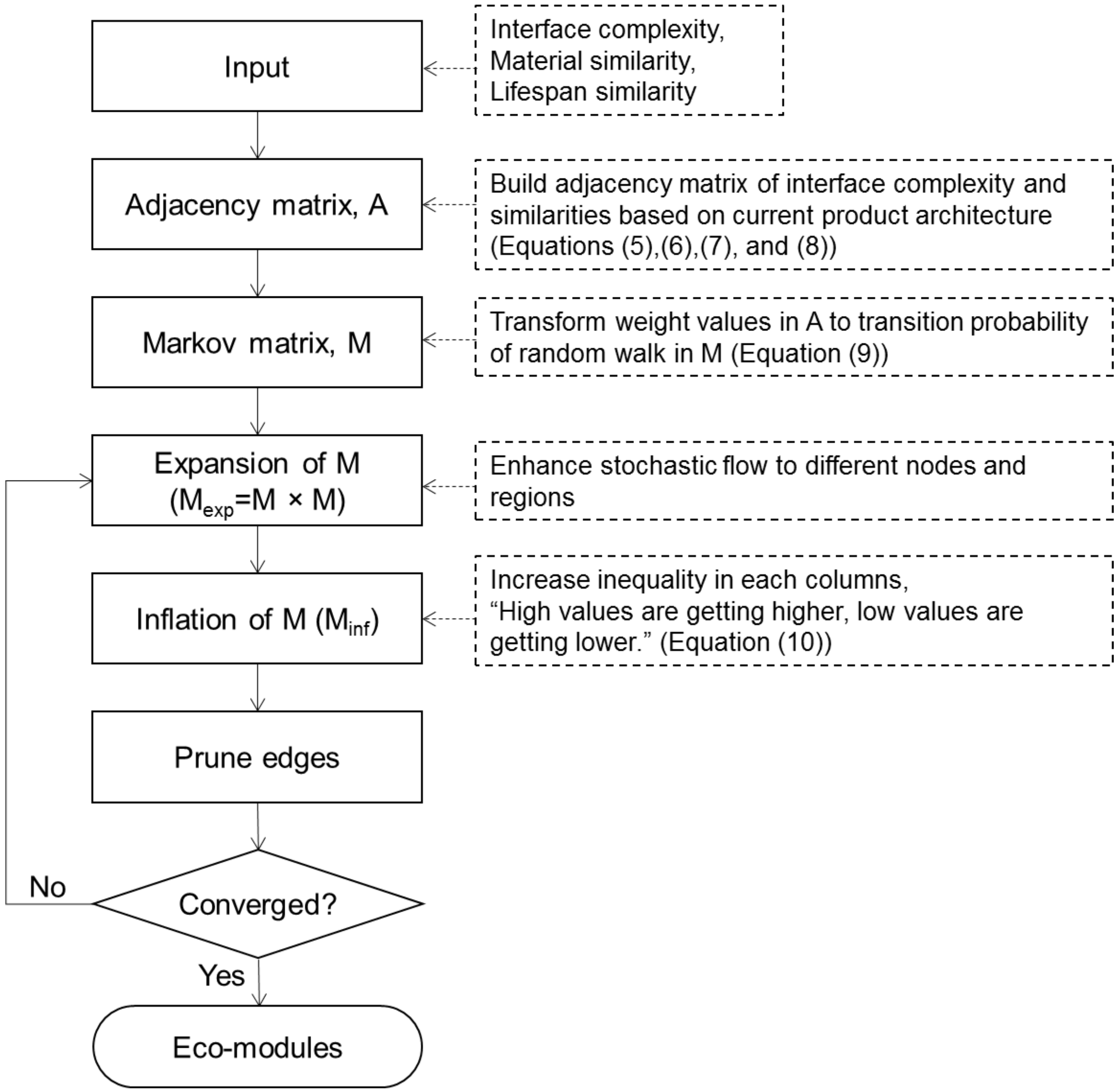


Figure 5 Clustering process for identifying the eco-modules by the MCL

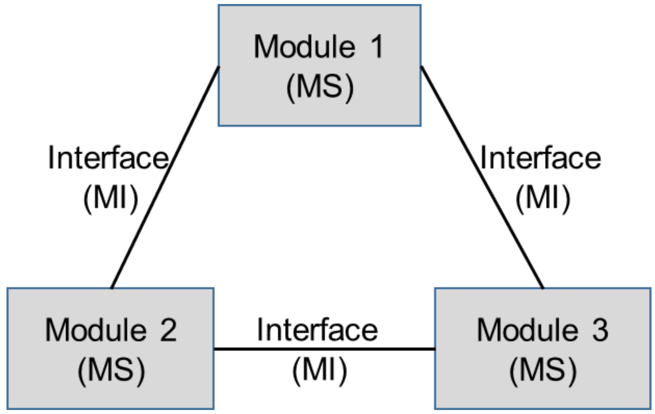
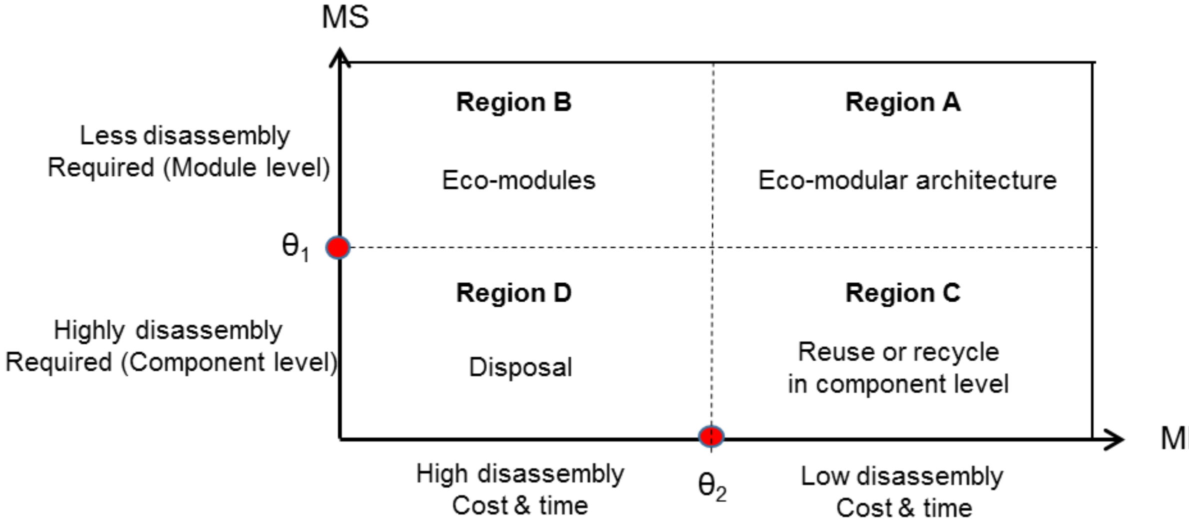
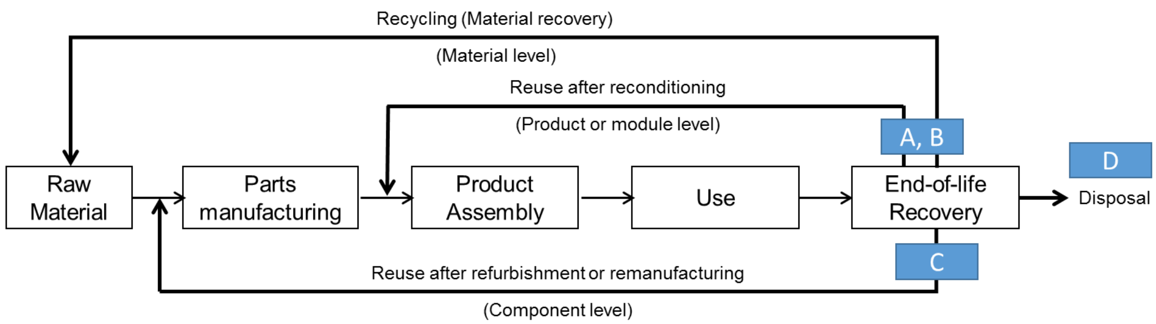


Figure 6 The assessment range of the proposed modularity metrics



(a) Four regions of modular product architecture



(b) Product recovery options of the modular product architecture in the four regions in product lifecycle

Figure 7 Modular architecture categories and their product recovery options in product lifecycle

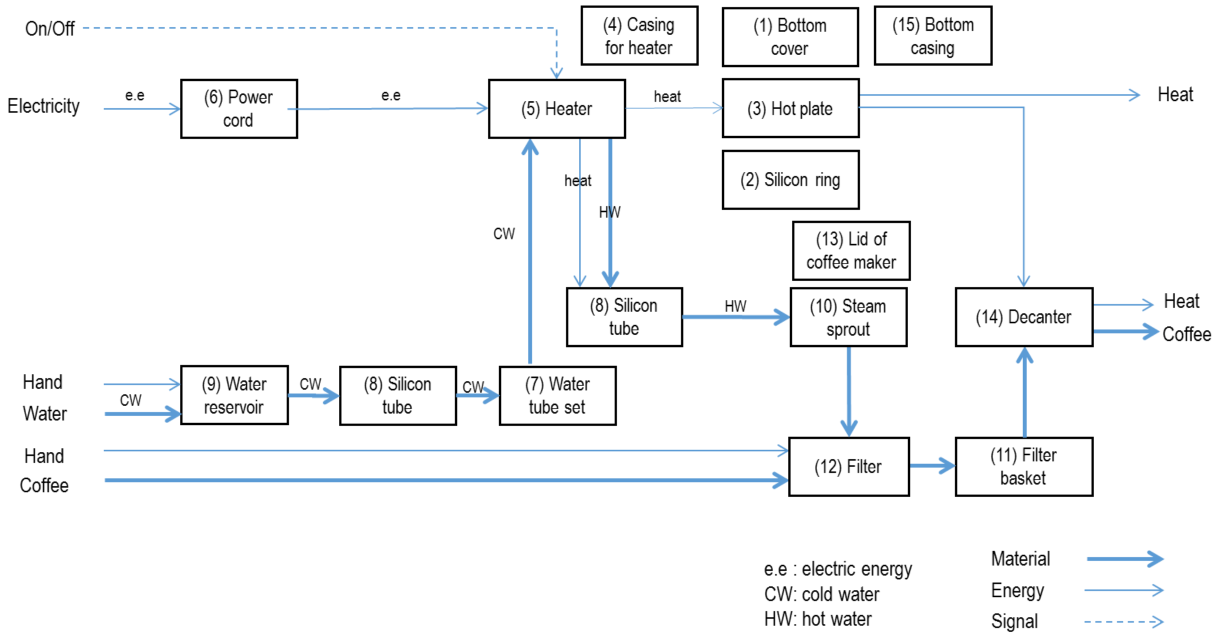


Figure 8 Functional diagram of the coffee maker

Module No.		1st			2nd			3rd					4th		5th	
DSM		1	4	15	2	3	14	7	8	9	10	13	11	12	5	6
1	Bottom cover		1	1												
4	Casing for heater	1		1	1	1									1	
15	Bottom casing	1	1			1	1			1		1			1	1
2	Silicon ring		1			1										
3	Hot plate		1	1	1		1								1	
14	Decanter			1		1							1		1	
7	Water tube set								1							
8	Silicon tube							1		1	1				1	
9	Water reservoir			1					1		1	1	1			
10	Steam sprout								1	1		1	1			
13	Lid of coffee maker			1						1	1					
11	Filter basket						1			1	1			1		
12	Filter												1			
5	Heater		1	1		1	1		1							1
6	Power cord			1											1	

Figure 9 Modules of the coffee maker identified by DSM clustering algorithm

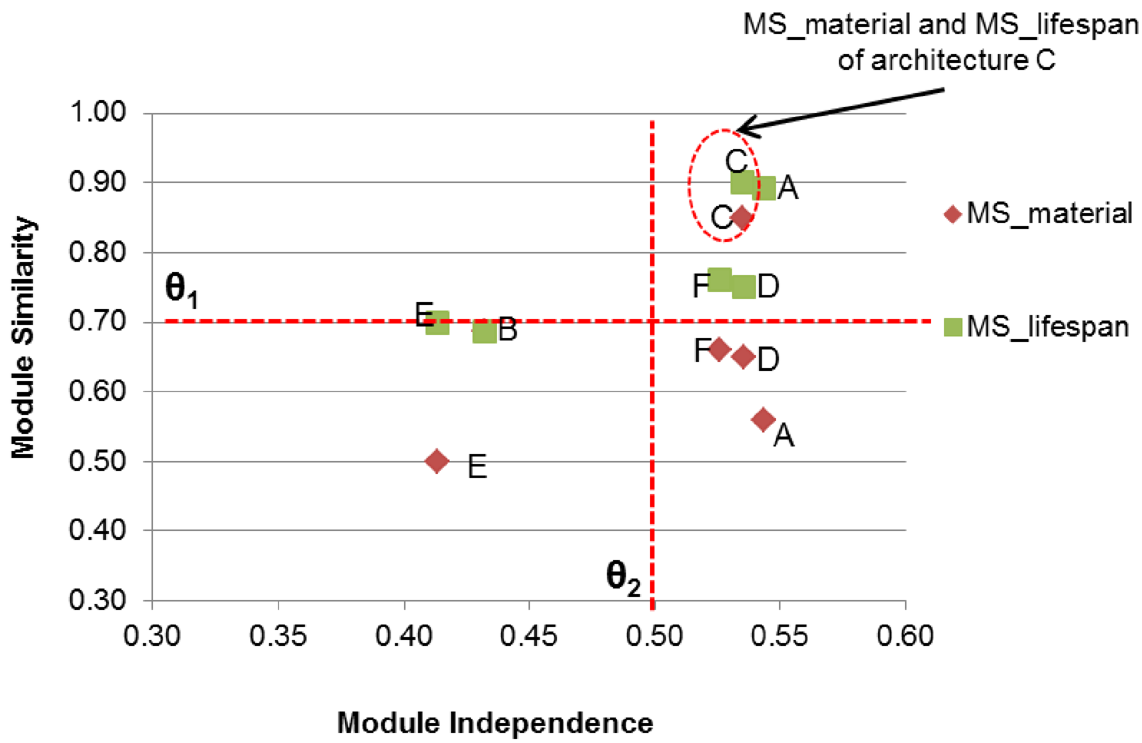


Figure 10 Categorized modular product architectures of the coffee maker in four regions according to the module independence and module similarity metrics

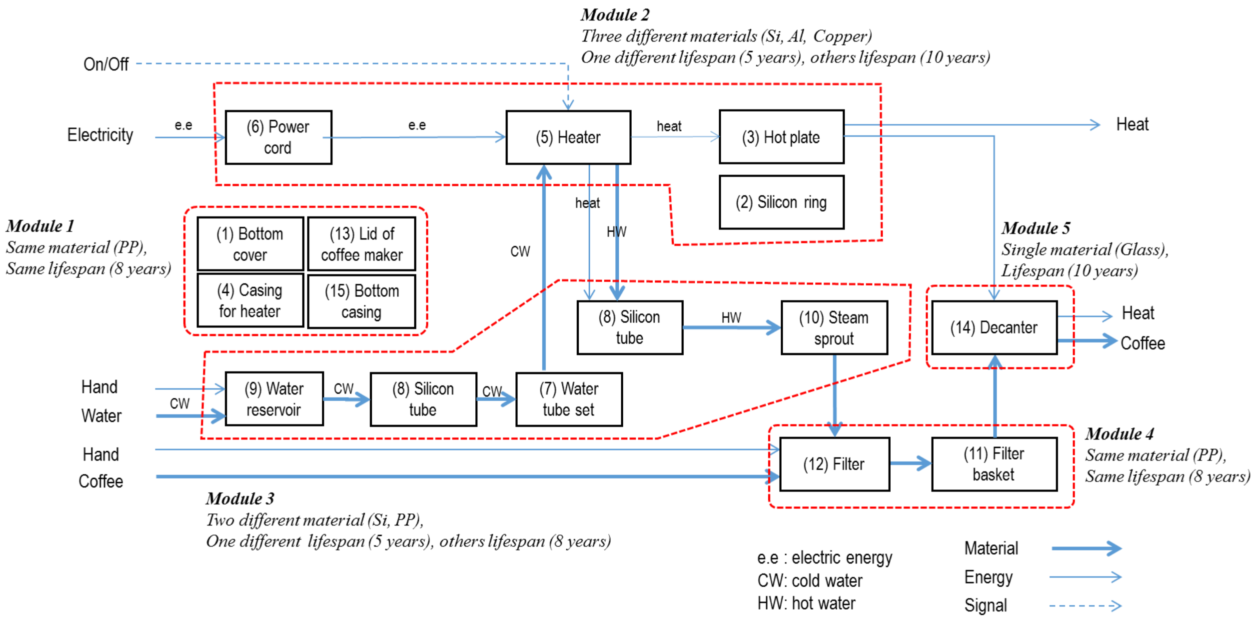


Figure 11 The eco-modular product architecture (C)