

네트워크 흐름 모델을 활용한 모듈러 제품 설계를 위한 컴포넌트 군집화

손지양, 유재욱*
동아대학교 경영학과

Components Clustering for Modular Product Design Using Network Flow Model

Jiyoung Son, Jaewook Yoo*

Department of Business Administration, Dong-A University

요약 모듈러 제품 설계는 유연한 제품 수정, 제품 개발 및 생산 리드타임 감소, 제품의 다양성 증대와 같은 많은 장점들이 있다. 모듈러 제품 설계의 목적은 여러 개의 모듈들로 구성된 제품의 아키텍처를 효율적으로 개발하는 것인데, 이들 모듈들은 컴포넌트들 간 물리적, 기능적 상호관계 분석을 토대로 컴포넌트들 간의 유사성을 최대화함으로써 만들어 질 수 있다. 본 연구에서는 모듈러 제품 설계의 핵심 작업인 모듈화를 위하여 3개 단계로 이뤄진 체계적인 절차를 제시하고자 한다. 첫 번째 단계는 컴포넌트들 간 물리적, 기능적 상호관계 분석을 통한 컴포넌트들 간 상관 관계 매트릭스를 구성하는 것이고, 두 번째 단계는 컴포넌트들 간 상관 관계를 최대화하는 컴포넌트들의 군집들을 찾아내기 위하여 네트워크 흐름으로 모델링하는 것이다. 마지막으로 세 번째 단계에서는 선형 계획 모형인 네트워크 흐름 모델을 풀어서 컴포넌트들의 군집들을 찾아내고 이들을 모듈화 하는 것이다. 본 연구에서 제시한 절차의 이해와 실제 적용을 위하여 진공 청소기 모듈화 사례에 적용해 보고 절차의 타당성을 보여준다.

Abstract Modular product design has contributed to flexible product modification and development, production lead time reduction, and increasing product diversity. Modular product design aims to develop a product architecture that is composed of detachable modules. These modules are constructed by maximizing the similarity of components based on physical and functional interaction analysis among components. Accordingly, a systematic procedure for clustering the components, which is a main activity in modular product design, is proposed in this paper. The first phase in this procedure is to build a component-to-component correlation matrix by analyzing physical and functional interaction relations among the components. In the second phase, network flow modeling is applied to find clusters of components, maximizing their correlations. In the last phase, a network flow model formulated with linear programming is solved to find the clusters and to make them modular. Finally, the proposed procedure in this research and its application are illustrated with an example of modularization for a vacuum cleaner.

Keywords : clustering, module, modularization, modular product design, network flow modeling

1. 서론

제품 수명주기가 짧아지는 시장 환경에 대응하기 위

해 대량생산체제의 규모의 경제는 유지하면서, 동시에
고객의 다양한 요구를 만족시킬 수 있는 대량고객화
(mass customization)를 실현하기 위한 여러 방법들 중

본 논문은 동아대학교 학술연구비 지원으로 연구되었음.

*Corresponding Author : Jaewook Yoo(Dong-A Univ.)

Tel: +82-51-200-7469 email: jyoo@dau.ac.kr

Received May 10, 2016

Revised (1st July 1, 2016, 2nd July 6, 2016)

Accepted July 7, 2016

Published July 31, 2016

의 하나로 모듈러 제품 설계(modular product design)가 인식되어져 왔다[1-3].

모듈러 제품 설계의 목적은 여러 개의 모듈(module)로 구성된 제품 아키텍처(product architecture)를 효율적으로 개발하는 것이다[4-6]. 유사한 기능을 수행하는 컴포넌트들의 군집으로 모듈을 정의함으로써 하나의 모듈 내에 존재하는 컴포넌트들 간의 유사성은 최대화되고, 서로 다른 모듈에 존재하는 컴포넌트들 간의 유사성은 최소화 된다[7, 8].

이와 같은 모듈의 특징은 다른 모듈에 영향을 미치지 않으면서 동시에 특정 모듈을 수정할 수 있으므로, 모듈러 제품 설계에는 유연한 제품 수정, 제품 개발 및 생산 리드타임(production lead time) 감소, 제품 다양성 증대와 같은 장점들이 있다[4, 6, 9].

모듈러 제품 설계에서 제품 수정은 컴포넌트들을 분리하거나 병합함으로써 이루어진다. 이 때, 기능적, 물리적으로 교류가 있는 컴포넌트들을 분리하거나 병합하면 제품개발 단계에서 계획했던 제품의 기능에 영향을 미칠 수도 있으므로 컴포넌트들 간의 기능적, 물리적 상호작용을 고려해야 한다. 기능적 상호작용은 컴포넌트들 사이에 자재(material), 에너지(energy), 신호(signal)를 교

환하는 것을 의미하며 제품의 전체적인 성능에 기여한다. 물리적 상호작용은 공간적(spatial), 기하학적(geometrical) 관계를 나타내며 컴포넌트들 간의 결합(attachment), 배치(arrangement), 동작(motion)의 형태로 표현된다.

모듈러 제품 설계에 관한 연구는 많이 이루어져왔다 [9, 10]. 본 논문에서는 모듈러 제품 설계에 관한 선행연구의 고찰은 모듈화를 위한 지표의 매트릭스 (index matrix) 개발과 컴포넌트들의 군집방법에 초점을 두고자 한다. Table 1은 모듈화를 위한 지표 개발과 컴포넌트들의 군집방법에 대한 선행연구들을 정리한 것이다. Table 1에서 볼 수 있듯이 대부분의 선행연구에서 지표는 기능적, 물리적 상호작용을 고려한 지표를 주로 사용하였고, 각 논문의 특징에 맞게 새로운 지표를 개발하여 사용하였다. 또한, 군집 방법으로는 대부분 선행연구에서 스와핑 알고리즘(swapping algorithm), 유전자 알고리즘(genetic algorithm)과 시뮬레이티드 어니일링(simulated annealing) 기법 등과 같은 휴리스틱(heuristic) 기법을 사용한 것을 볼 수 있다.

Table 1의 선행연구를 좀더 상세히 보면, Pimmmler와 Eppinger [11]는 제품 아키텍처를 재설계하거나 제품 개

Table 1. The list of literature

no	title	author	used index	clustering methodology
1	Integration analysis of product decompositions(1994)	Pimmmler & Eppinger[11]	- functional, physical interaction	heuristic swapping algorithm
2	Macro level product development using design for modularity(1999)	Sa'ed & Kamrani [12]	- functional, physical characteristic - product requirements - similarity index	P-median model
3	Modular product design with grouping genetic algorithm—a case study(2004)	Kreng & Lee [6]	- functional, physical interaction	grouping genetic algorithm
4	An information theoretic method for developing modular architectures using genetic algorithms(2007)	Yu & Yassine [13]	- judgement of expert	genetic algorithm MDL DSM
5	An integrated modular design methodology for life-cycle engineering(1997)	Gu & Sosal [5]	- assembly procedure, assembly line - functional, physical interaction	genetic algorithm
6	Product modularization for life cycle engineering(1999)	Gu & Sosal [4]	- assembly procedure, assembly line - functional, physical interaction	simulated annealing
7	Product modular design incorporating life cycle issues-Group Genetic Algorithm (GGA) based method(2011)	Yu & Yang [14]	- functional, physical interaction - lifetime, compatibility, recyclability	grouping genetic algorithm
8	A heuristic method for identifying modules for product architectures (2000)	Stone & Wood [15]	-functional model	3steps heuristic method

발품을 재구성하기 위해 제품을 컴포넌트로 분해하고 제품의 구조적 배열, 에너지 교환, 정보 교환, 자재 교환의 네 가지 관점에서 컴포넌트들 간의 기능적, 물리적 관계를 측정된 지표들에 가중치를 부여하여 하나의 매트릭스로 통합시켰다. 이 매트릭스를 기반으로 휴리스틱 스와핑 알고리즘(heuristic swapping algorithm)을 사용하여 컴포넌트를 군집시키고 모듈을 정의하였다.

Sa'ed와 Kamrani[12]는 제품의 물리적, 기능적 특징에 대한 각 컴포넌트들 간의 관계, 제품의 요구사항과 컴포넌트 간의 관계, 컴포넌트들 간 유사성 지표(similarity index)를 통합하여 하나의 매트릭스(matrix)로 나타내고 유사성의 합을 최대화하는 최적화 모델인 P-median 모델을 활용하여 컴포넌트들을 군집하였다.

Victor B. Kreng과 Tseng-Pin Lee[6]는 컴포넌트들 간의 기능적, 물리적 상관관계를 매트릭스 형태로 표현하고 모듈의 개수에 대한 제약을 반영하여 그룹핑 유전자 알고리즘(grouping genetic algorithm)을 사용하여 모듈을 탐색하였다.

Yu와 Yassine[13]는 유전자 알고리즘(genetic algorithm)과 최소묘사길이(minimum description length)에 기반한 새로운 Design Structure Matrix(DSM)을 컴포넌트들의 군집 방법으로 개발하였다.

Gu와 Sosale[4, 5]는 제품의 수명주기를 고려하기 위해 조립 절차, 조립 라인을 모듈화하는 목적으로 이들 각각의 목적에 따른 컴포넌트들 간의 물리적, 기능적 관계를 측정하여 하나의 매트릭스로 통합한 후 유전자 알고리즘[4]과 시뮬레이티드 어니일링(simulated annealing) 기법[5]으로 모듈을 개발하였다.

Yu와 Yang[14]의 연구에서도 MDF(Modular Driving Forces)로써 컴포넌트들 간의 물리적, 기능적 관계와 컴포넌트의 수명, 호환성, 재활용도를 고려한 후 각각의 MDF에 가중치를 부여하여 통합 매트릭스를 생성하고 그룹 유전자 알고리즘(group genetic algorithm)을 활용하여 모듈을 정의하였다.

한편, 컴포넌트간의 물리적, 기능적 관계를 활용한 컴포넌트 군집화와 같은 모듈러 제품 설계 이외에도 Stone과 Wood[15]는 제품의 기능 흐름(functional flow)에 기반하여 3단계 휴리스틱 방법(dominant flow, branching flow, conversion)으로 모듈을 정의하였고, Erixon[16]은 품질기능전개(quality function deployment)를 활용하여 모듈을 정의하는 세 단계를 개발하였다. 또한, Hölttä

[17]는 제품군을 형성하기 위해 군집 5단계 알고리즘을 개발하고 계통수(dendrogram)를 그려 제품 플랫폼의 공통 모듈을 찾았다. Ericsson과 Erixon[18]은 기업의 특정한 전략을 달성하기 위하여 인터페이스를 정의하고 모듈화 하였다.

위의 선행연구의 고찰에서 보면 모듈러 제품 설계의 컴포넌트 모듈화에 대한 기존의 대부분의 연구들은 휴리스틱(Heuristic) 접근법으로 모듈을 정의함으로써 근사 최적 모듈러 아키텍처(near-optimal modular architecture) 수준의 해답을 찾는데 한정되어왔다. [12]에서는 최적 모듈을 찾을 수 있는 P-median 모델을 활용하였으나, 모듈의 개수를 미리 정해야 하는 제약이 있다.

이에 본 논문에서는 네트워크 흐름 모델(network flow model)을 이용한 컴포넌트들의 군집절차를 제시함으로써 모듈화를 위한 최적 해를 제시하는 데 본 연구의 기여가 있고, 또한 보다 실질적인 개발 환경인 이미 시장에 출시된 제품을 대상으로 고객의 새로운 요구사항에 맞게 제품을 수정하거나 또는 새로운 시장에 제품을 출시하기 위해 제품을 재설계하는 Bottom-Up 개발 방식에서 본 연구에서 제시한 모델이 적용 가능하여 상당히 실용적인 방법론이라 볼 수 있겠다.

본 논문의 제 2장에서는 제품의 컴포넌트들 간의 상호작용을 최대화하기 위해 네트워크 흐름 기법을 사용한 컴포넌트 군집 절차를 제시한다. 제 3장에서는 본 연구에서 제시한 군집화 기법을 사례연구에 적용해봄으로써 실제 기업에서의 적용가능성을 살펴본다. 제 4장에서는 본 연구의 결론과 향후 연구방향을 제시한다.

2. 네트워크 흐름 모델을 활용한 컴포넌트 군집 절차

본 연구에서는 컴포넌트들 간의 기능적, 물리적 상호작용(유사성)을 측정하고 컴포넌트들을 모듈로 군집하기 위한 방법으로 네트워크 흐름 모델을 활용한다. 네트워크 흐름 모델을 활용한 컴포넌트의 군집 절차는 (1) 컴포넌트 간 상호작용 분석(interaction analysis)을 하고, (2) 유사한 기능을 가진 컴포넌트들의 군집을 위하여 capacitated - circulation 네트워크 흐름으로 모델링하여, (3) capacitated-circulation 네트워크 흐름 문제에 대한 수리모형을 세우고 문제를 해결함으로써 컴포넌트들을

군집하고 모듈을 정의한다.

2.1 컴포넌트 간 상호작용 분석

대부분의 모듈 설계 방법론들은 유사한 기능을 가진 컴포넌트들을 군집하여 모듈화 한다. 이때 각 모듈은 특정 기능을 수행하기 위해 각 모듈 내의 컴포넌트들 간의 유사성은 최대화 하고, 다른 모듈 간 컴포넌트들의 유사성은 최소화 하는 모듈을 찾는다. 이와 같이 모듈화의 첫 번째 단계는 컴포넌트들 간의 물리적 상호작용 매트릭스와 기능적 상호작용 매트릭스를 만드는 것이다. 그리고 서로 다른 컴포넌트들 간의 관계의 중요도를 나타내는 가중 매트릭스(weight matrix)를 형성하고, 상호작용 매트릭스와 가중 매트릭스를 곱하여 상관관계 매트릭스(correlation matrix)를 형성한다. 이는 결국 컴포넌트들 간의 유사성(similarity)을 수치로 나타내며 물리적, 기능적 관계에 따른 근거가 된다. 상관관계 매트릭스의 형성 과정을 식으로 표현하면 다음과 같다.

$$CV_{ij} = W_{ij}^P * PI_{ij} + W_{ij}^F * FI_{ij} \quad (1)$$

CV_{ij} 는 컴포넌트 i 와 j 의 상관관계 값(correlation value)이다. PI_{ij} 는 컴포넌트 i 와 j 의 물리적 상관관계(physical interaction)이며 FI_{ij} 는 컴포넌트 i 와 j 의 기능적 상관관계(functional interaction)이다. PI_{ij} 와 FI_{ij} 는 Table 2와 Table 3에서와 같이 물리적, 기능적 상호작용 매트릭스의 원소로 나타낼 수 있다. 한편, PI_{ij} 와 FI_{ij} 의 값은 컴포넌트들 간의 상호작용측정 기준에 따라 달라질 수 있다.

W_{ij}^P 는 컴포넌트 i 와 j 의 물리적 관계의 중요도를 나타내는 물리적 가중치(physical weight)이며 W_{ij}^F 는 컴포넌트 i 와 j 의 기능적 관계의 중요도를 나타내는 기능적 가중치(functional weight)이다. W_{ij}^P , W_{ij}^F 는 Table 4와 Table 5에서와 같이 물리적, 기능적 상호작용에 대한 가중 매트릭스의 원소로 나타낼 수 있다. $i \neq j$ 인 경우, 각 i 에 대해 $\sum_{j=1}^N W_{ij}^P$, $\sum_{j=1}^N W_{ij}^F$ 가 각각 0.5가 되도록 W_{ij}^P 와 W_{ij}^F 의 값을 0~0.5 사이의 값으로 측정한다. $i = j$ 인 경우, W_{ij}^P , W_{ij}^F 는 각각 0.5의 값을 가진다. 따

라서 각 i 에 대한 $\sum_{j=1}^N W_{ij}^P$, $\sum_{j=1}^N W_{ij}^F$ 는 모두 1이 된다.

모든 CV_{ij} 값을 식(1)을 이용하여 구하고 Table 6과 같은 상관관계 매트릭스를 만든다. 1단계의 결과로 얻은 상관관계 매트릭스는 컴포넌트들을 군집하기 위한 정보가 된다.

Table 2. The physical interaction matrix

component	1	2	3	...	N
1	PI_{11}	PI_{12}	PI_{13}	...	PI_{1N}
2	PI_{21}	PI_{22}	PI_{23}	...	PI_{2N}
3	PI_{31}	PI_{32}	PI_{33}	...	PI_{3N}
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$
N	PI_{N1}	PI_{N2}	PI_{N3}	...	PI_{NN}

Table 3. The functional interaction matrix

component	1	2	3	...	N
1	FI_{11}	FI_{12}	FI_{13}	...	FI_{1N}
2	FI_{21}	FI_{22}	FI_{23}	...	FI_{2N}
3	FI_{31}	FI_{32}	FI_{33}	...	FI_{3N}
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$
N	FI_{N1}	FI_{N2}	FI_{N3}	...	FI_{NN}

Table 4. The weight matrix for physical interaction

component	1	2	3	...	N
1	W_{11}^P	W_{12}^P	W_{13}^P	...	W_{1N}^P
2	W_{21}^P	W_{22}^P	W_{23}^P	...	W_{2N}^P
3	W_{31}^P	W_{32}^P	W_{33}^P	...	W_{3N}^P
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$
N	W_{N1}^P	W_{N2}^P	W_{N3}^P	...	W_{NN}^P

Table 5. The weight matrix for functional interaction

component	1	2	3	...	N
1	W_{11}^F	W_{12}^F	W_{13}^F	...	W_{1N}^F
2	W_{21}^F	W_{22}^F	W_{23}^F	...	W_{2N}^F
3	W_{31}^F	W_{32}^F	W_{33}^F	...	W_{3N}^F
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$
N	W_{N1}^F	W_{N2}^F	W_{N3}^F	...	W_{NN}^F

Table 6. The correlation matrix

component	1	2	3	...	N
1	CV_{11}	CV_{12}	CV_{13}	...	CV_{1N}
2	CV_{21}	CV_{22}	CV_{23}	...	CV_{2N}
3	CV_{31}	CV_{32}	CV_{33}	...	CV_{3N}
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$
N	CV_{N1}	CV_{N2}	CV_{N3}	...	CV_{NN}

2.2 capacitated-circulation 네트워크 수립

Capacitated-circulation 네트워크 $G=(V,E)$ 는 $N \times N$ 상호작용 매트릭스에 기반하여 형성된다. V 는 노드(vertex 또는 node)의 집합이고, E 는 아크(edge 또는 arc)의 집합이며 이 네트워크 모델에서 컴포넌트 i 는 직선으로 된 아크(i_a, i_b)에 연결된 두 노드 i_a, i_b 로 표현된다.

네트워크 모델을 이용하여 N 개 컴포넌트들의 모듈화 문제를 풀기 위해 다음과 같은 절차로 네트워크를 형성한다.

- 1) $i_a, i_b, i=1, \dots, N$ 으로 표현되는 $2N$ 개의 노드를 만든다. 노드 $1_b, 1_a$ 를 각각 네트워크의 소스 노드(source node), 싱크 노드(sink node)라 한다.
- 2) 네트워크의 나머지는 다음 원칙에 따라 구성된다. 각 아크에는 그 아크에 흐르는 흐름의 상한(upper bound), 하한(lower bound), 단위당 흐름비용 $[U, L, C]$ 값이 할당된다.
 - (2.1) 각 노드 $i_a (i \neq 1)$ 에 대해 i_a 에서 i_b 로 연결되는 아크를 생성한다. 이 때 $[U, L, C] = [1, 1, CV_{ii}]$ 이며 CV_{ii} 는 상관관계 매트릭스에서 대각선에 있는 값으로, 컴포넌트들 간의 상관관계 계수가 가질 수 있는 최댓값이다. 이는 컴포넌트 i 자신을 의미한다.
 - (2.2) 각 노드 $i_b (i=2, \dots, N)$ 에 대해 i_b 에서 노드 $i_a (i=2, \dots, N)$ 로 연결되는 $(N-2)$ 개의 아크를 생성한다. 이 때 $[U, L, C] = [1, 0, CV_{ij}]$ 이며, $i \neq j$ 이다.
- 3) 1_b 에서 j_a 로 연결되는 $[U, L, C] = [1, 0, CV_{1j}]$, $j \neq 1$ 인 아크를 생성하고 i_b 에서 1_a 로 연결되는 $[U, L, C] = [1, 0, CV_{i1}]$, $i \neq 1$ 인 아크를 생성한다.

- 4) 네트워크를 circulation 네트워크로 만들기 위해 $[U, L, C] = [1, 1, CV_{11}]$ 인 리턴(return) 아크($1_a, 1_b$)를 생성한다. 이때 CV_{11} 도 (2.1)에서와 같이 최댓값을 가진다.

Fig. 1은 앞서 설명한 circulation 네트워크 구성을 나타낸 그림이다. 이 네트워크 흐름 모델에서는 흐름량 보존 조건(flow conservation condition)에 의해 i_b 가 노드 $j_a (j \neq i)$ 로 1 단위의 흐름을 보낼 수 있도록 아크(i_a, i_b)에 대한 흐름은 항상 1이다. 각 노드 i_b 에서 각 노드 j_a 로 한 단위의 흐름을 흘림으로써 발생하는 값은 컴포넌트 i 와 j 의 상관관계 매트릭스에 정의된 값이다($j \neq i$).

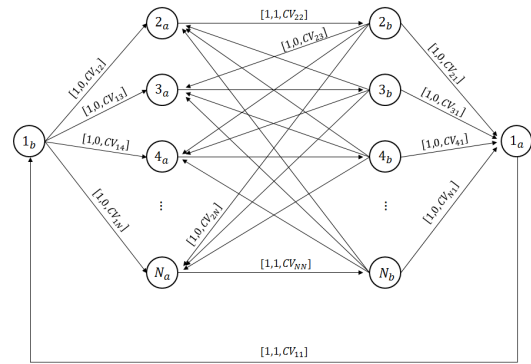


Fig. 1. The capacitated-circulation network

2.3 컴포넌트 군집화

capacitated-circulation 네트워크가 형성되고 나면 네트워크에 대한 수리모형을 세우고 네트워크 문제를 해결함으로써 각 모듈 내의 상호작용 즉, CV_{ij} 가 최대가 되는 군집을 찾는다. 수리모형은 다음과 같다.

목적함수

$$\text{Max} \sum_{i=1}^N \sum_{j=1}^N CV_{ij} f(i_b, j_a) \quad (2)$$

제약식

$$\sum_{i=1}^N f(i_b, j_a) = 1 \text{ for all } j_a \in V, i \neq j \quad (3)$$

$$\sum_{j=1}^N f(i_b, j_a) = 1 \text{ for all } i_b \in V, i \neq j \quad (4)$$

$$0 \leq f(i_b, j_a) \leq 1 \text{ for all } (i, j) \in E \quad (5)$$

$f(i_b, j_a)$ 는 아크(i, j)를 지나는 흐름이다. 목적 함수 (2)는 부품 간의 유사성을 최대화 하며 제약식 (3),(4)는 각 노드에 대한 흐름량 보존 조건을 나타낸다. 제약식 (5)는 아크의 흐름에 대한 용량 조건이다. 이 수리모형은 선형 계획 모델(Linear programming model)이며, 본 논문에서는 capacitated-circulation 네트워크 문제를 풀기 위해 Matlab의 최적화 도구를 사용하여 네트워크내 컴포넌트들 간의 상관관계를 최대화하는 최적해를 찾게 되며, 그 해들로 인해 컴포넌트들의 군집이 형성된다. 이때 군집된 컴포넌트들이 하나의 모듈이 된다.

3. 사례연구

Kreng 과 Lee[6]의 연구에서 다룬 기업A의 제품 모듈화 사례를 본 연구에서 제시하고 있는 네트워크 흐름 모델을 활용한 컴포넌트 군집 절차에 적용해 보고자 한다.

타이완의 가전 제품 제조기업인 A기업은 진공 청소기 ‘TK-N929’모형을 일본 시장에 출시하기 위해 제품 수정이 필요하다. Table 7에 있는 35개의 컴포넌트들은 진공 청소기 ‘TK-N929’본체의 주요 기능을 고려하여 흡입튜브(sucking tube)와 헤드(head)를 제외한 컴포넌트들이다.

Table 7. The components list of ‘TK-N929’

No.	Components name	No.	Components name
1	Handle	19	Front cushion
2	Winder pedal	20	Inlet filter
3	Power pedal	21	Inlet filter cover
4	Indicator light	22	Back wheel
5	Top body	23	Base body
6	Indicator	24	Fixed seal
7	Storage cover	25	Front cover
8	Bumper	26	Front wheel base
9	Exhaust filter cover	27	Front wheel
10	Exhaust filter	28	Cord reel
11	Cord gate	29	Actuation shaft
12	Receiver pcb	30	Pbc shell
13	Small pcb box	31	Motor driver pbc
14	Control pcb	32	Heat sinking board
15	Rear cushion	33	Up housing
16	Small sound proof sponge	34	Down housing
17	Sound proof sponge	35	Dust container
18	Motor		

3.1 컴포넌트 간 상호작용 분석

진공청소기의 컴포넌트들 간의 물리적, 기능적 상관관계 지표인 PI_{ij} , FI_{ij} 가 없음, 약함, 중간, 강함을 나타내는 0, 1, 3, 9의 값을 가질 때, 상관관계 값 CV_{ij} 는 식 (1)에 의해 Table 8과 같이 계산된다[5]. 컴포넌트 17(soundproof sponge)과 18(motor)을 예로 들자면, 물리적 가중치(W_{1718}^P), 기능적 가중치(W_{1718}^F), 물리적 상

Table 8. The correlation matrix of ‘TK-N929’

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35
1	9	0.3	0.3	0.3	9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0.3	9	1.4	3.9	6	0	0	0	0	0	0	0	0	1.2	0	0	0	0	0	0	0	0	0	0	0	0	0	1.2	0	0	0	0	0	0	0
3	0.3	1.4	9	3.9	2.1	0	0	0	0	0	0	0	0	1.2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1.2	0	0	0
4	0.3	3.9	3.9	9	6	0	0	0	0	0	0	0	0.6	0.3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.3	0	0	0
5	9	6	2.1	6	9	0	0	1.2	0	0.9	0.3	0.3	0.1	0	0	0	0	0	0	0	0	0	4.2	0	0	0	0	0.2	0.3	1.2	0	0	0.3	0.3	0
6	0	0	0	0	0	9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.3	0.3
7	0	0	0	0	0	9	9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	4.5	6	9	0	0	0	0	0	0	0	0	0	9
8	0	0	0	0	0	0	9	0	0	0	0	0	0	0	0	0	0	0	0	0.6	0.6	0	9	0	1.5	0	0	0	0	0	0	0	0.3	0.3	0
9	0	0	0	0	1.2	0	0	9	9	0	0	0	0	1.5	5.7	8.1	2.1	0.5	4.8	4.8	0	0.9	0	0	0	0	0	0	0	0.6	0	9	9	0	0
10	0	0	0	0	0	0	9	9	0	0	0	0	0	1.5	5.7	5.7	1.1	0.5	3.9	3.9	0	0.1	0	0	0	0	0	0	0	0	0	0	5.7	5.7	0
11	0	0	0	0	0.9	0	0	0	0	9	0	0	0	0	0	0	0	0	0	0	0	0.6	0	0	0	0	0	9	9	0	0	0	0	0	0
12	0	0	0	0.6	0.3	0	0	0	0	0	0	9	7.2	9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	8.1	4.8	1.5	0	0
13	0	0	0	0	0.3	0	0	0	0	0	0	0	7.2	9	8.1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	8.1	4.8	1.5	0	0
14	0	1.2	1.2	0.3	0.1	0	0	0	0	0	0	9	8.1	9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	9	4.6	1.3	0	0	0
15	0	0	0	0	0	0	0	1.5	1.5	0	0	0	0	9	0.5	0.5	9	9	0.5	0.5	0	6.8	0	0	0	0	0	0	0	0	0	0	0.5	0.5	0.1
16	0	0	0	0	0	0	0	5.7	5.7	0	0	0	0	0.5	9	9	1.5	0.5	4.8	4.8	0	0.2	0	0	0	0	0	0	0	0	0	0	8.1	8.1	0
17	0	0	0	0	0	0	0	8.1	5.7	0	0	0	0	0.5	9	9	2.1	0.5	4.8	4.8	0	0.2	0	0	0	0	0	0	0	0	0	0	5.7	5.7	0
18	0	0	0	0	0	0	0	2.1	1.1	0	0	0	0	9	1.5	2.1	9	9	1.1	2.1	0	1.6	0	0	0	0	0	0	0	0.6	0.3	0	1.5	1.5	0.6
19	0	0	0	0	0	0	0	0.5	0.5	0	0	0	0	9	0.5	0.5	9	9	1.5	1.5	0	0.9	0	0	0	0	0	0	0	0	0	0.5	0.5	0.9	
20	0	0	0	0	0	0	0	0.6	4.8	3.9	0	0	0	0.5	4.8	4.8	1.1	1.5	9	9	0	0.2	0.6	0.6	0	0	0	0	0	0	0	5.1	5.1	0.8	
21	0	0	0	0	0	0	0	0.6	4.8	3.9	0	0	0	0.5	4.8	4.8	2.1	1.5	9	9	0	1.2	0.6	0.6	0	0	0	0	0	0	0	8.1	8.1	1.8	
22	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	9	9	0	0	5.4	4.7	0	0	0	0	0	0	0	0
23	0	0	0	0	4.2	0	4.5	9	0.9	0.1	0.6	0	0	0	6.8	0.2	0.2	1.6	0.9	0.2	1.2	9	9	1.2	3.3	9	9	1.5	0.6	4.2	0.1	0	0.9	0.9	1.2
24	0	0	0	0	0	6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.6	0.6	1.2	9	9	0	0	0	0	0	0	0.3	0.3	9
25	0	0	0	0	0	9	1.5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.6	0.6	3.3	9	9	0	0	0	0	0	0	0.3	0.3	9
26	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	5.4	9	0	0	9	9	0	0	0	0	0	0	0	0
27	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	4.7	9	0	0	9	9	0	0	0	0	0	0	0	0
28	0	1.2	0	0	0.2	0	0	0	0	9	0	0	0	0	0	0	0	0	0	0	0	1.5	0	0	0	9	9	0	0	0	0	0	0	0	0
29	0	0	0	0	0.3	0	0	0	0	9	0	0	0	0	0	0	0	0	0	0	0	0.6	0	0	0	9	9	0	0	0	0	0	0	0	0
30	0	0	0	0	1.2	0	0	0	0.6	0	8.1	8.1	9	0	0	0	0.6	0	0	0	0	4.2	0	0	0	0	0	0	9	0.7	0.3	0	0.2	0	0
31	0	0	1.2	0.3	0	0	0	0	0	0	0	4.8	4.8	4.6	0	0	0	0.3	0	0	0	0.1	0	0	0	0	0	0	0.7	9	9	0	0	0	
32	0	0	0	0	0	0	0	0	0	0	1.5	1.5	1.3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.3	9	9	0	0	
33	0	0	0	0	0.3	0	0.3	9	5.7	0	0	0	0	0.5	8.1	5.7	1.5	0.5	5.1	8.1	0	0.9	0.3	0.3	0	0	0	0	0	0	0	9	9	0.3	
34	0	0	0	0	0.3	0	0.3	9	5.7	0	0	0	0	0.5	8.1	5.7	1.5	0.5	5.1	8.1	0	0.9	0.3	0.3	0	0	0	0	0	0.2	0	9	9	0.3	
35	0	0	0	0	0	3.6	9	0	0	0	0	0	0	0	0.1	0	0	0.6	0.9	0.8	1.8	0	1.2	9	9	0	0	0	0	0	0	0.3	0.3	9	9

관관계 (PI_{1718}), 기능적 상관관계 (FI_{1718})가 각각 0.4, 0.3, 3, 3의 값을 가진다. 따라서 식(1)에 의하여 상관관계 값 CV_{1718} 은 2.1이 된다.

3.2 capacitated-circulation 네트워크 수립

35개의 컴포넌트들을 $i_a, i_b, i = 1, \dots, 35$ 인 $2 \times 35 = 70$ 개의 노드를 만들고 2장에서 소개한 capacitated-circulation 네트워크 수립 절차에 따라 네트워크를 만든다. 앞에서 언급하였듯이 이때 단위당 흐름 값 CV_{ij} 는 1단계에서 구한 상관관계 계수이다.

3.3 컴포넌트 군집화

본 사례를 수리모형식 (2)~(5)에 적용하여 Matlab의 최적화 도구를 사용하여 문제를 푼 결과, Table 9과 같이 최적 흐름 값과 이들 최적 값들을 토대로 15개의 루프(loop)를 형성하였다. Fig. 2는 Table 9의 결과를 네트워크 흐름 모델로 보여준다.

Table 9. The optimal flow and loop of ‘TK-N929’

Loop no.	Optimal Flow			Sum
1	$f(1_b, 5_a)$	$f(5_b, 1_a)$		18
	9	9		
2	$f(2_b, 3_a)$	$f(3_b, 4_a)$	$f(4_b, 2_a)$	9.2
	1.4	3.9	3.9	
3	$f(6_b, 7_a)$	$f(7_b, 6_a)$		18
	9	9		
4	$f(8_b, 23_a)$	$f(23_b, 8_a)$		18
	9	9		
5	$f(9_b, 10_a)$	$f(10_b, 9_a)$		18
	9	9		
6	$f(11_b, 29_a)$	$f(29_b, 28_a)$	$f(28_b, 11_a)$	27
	9	9	9	
7	$f(12_b, 14_a)$	$f(14_b, 12_a)$		18
	9	9		
8	$f(13_b, 30_a)$	$f(30_b, 13_a)$		16.2
	8.1	8.1		
9	$f(15_b, 18_a)$	$f(18_b, 19_a)$	$f(19_b, 15_a)$	27
	9	9	9	
10	$f(16_b, 17_a)$	$f(17_b, 16_a)$		18
	9	9		
11	$f(20_b, 21_a)$	$f(21_b, 20_a)$		18
	9	9		
12	$f(22_b, 26_a)$	$f(26_b, 27_a)$	$f(27_b, 22_a)$	19.1
	5.4	9	4.7	
13	$f(24_b, 25_a)$	$f(25_b, 35_a)$	$f(35_b, 24_a)$	27
	9	9	9	
14	$f(31_b, 32_a)$	$f(32_b, 31_a)$		18
	9	9		
15	$f(33_b, 34_a)$	$f(34_b, 33_a)$		18
	9	9		
Total				287.5

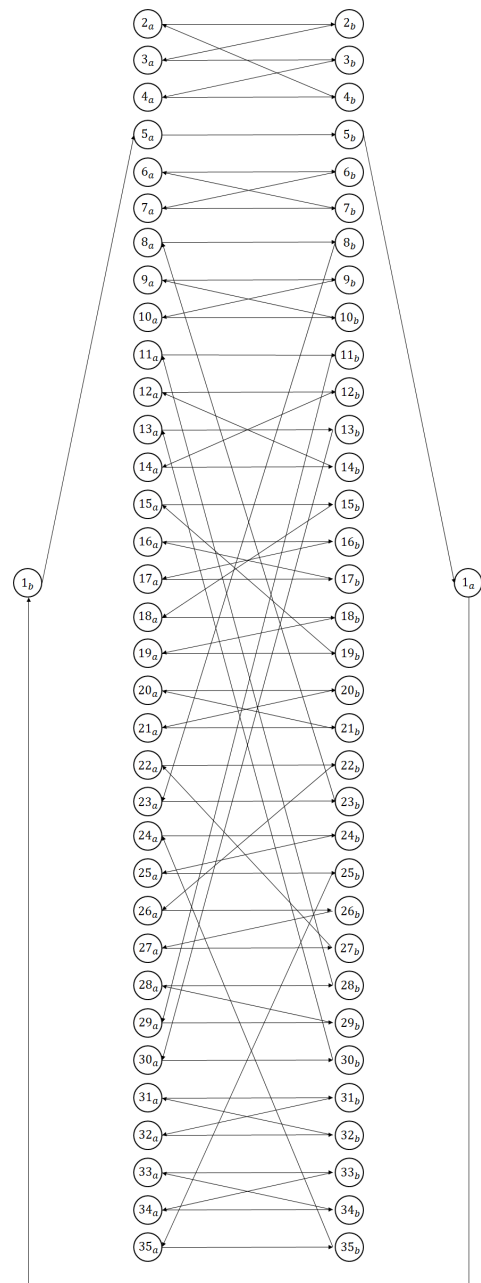


Fig. 2. The optimal network flow of ‘TK-N929’

Table 9의 각각의 루프(loop)가 컴포넌트들의 군집을 나타내며 각각의 군집은 하나의 모듈이 된다. 따라서, 본 연구에서 제시한 capacitated-circulation 네트워크 모델을 활용하여 구한 진공 청소기 ‘TK-N929’의 모듈들은 Table 10과 같이 나타낼 수 있다.

Table 10. The optimal flow and loop of ‘TK-N929’

Module No.	Component No.	Module No.	Component No.
1	1, 5	9	15, 18, 19
2	2, 3, 4	10	16, 17
3	6, 7	11	20, 21
4	8, 23	12	22, 26, 27
5	9, 10	13	24, 25, 35
6	11, 28, 29	14	31, 32
7	12, 14	15	33, 34
8	13, 30		

Fig. 3은 컴포넌트들의 군집 결과를 모듈로 나타낸 것이다. 동시에 서로 다른 모듈 사이에 Table 8에서의 상관관계 값이 6 이상인 컴포넌트들을 표시함으로써 모듈간 상호작용(inter-modular interaction)을 나타내었으며, 이 값은 개발 전문가의 판단으로 결정된다. 모듈간 상호작용은 군집 절차를 거쳐 서로 다른 두 모듈에 속해 있지만 컴포넌트간의 상관관계 값이 큰 컴포넌트들을 표시한 것으로 모듈간의 물리적, 기능적 접점인 인터페이스(interface)와 연관이 있다[5].

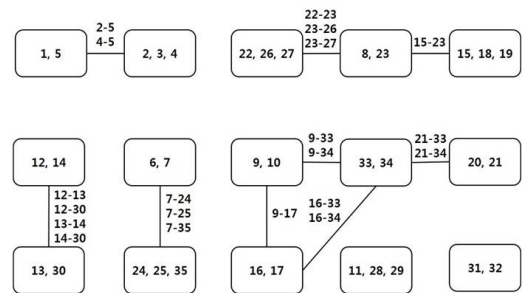
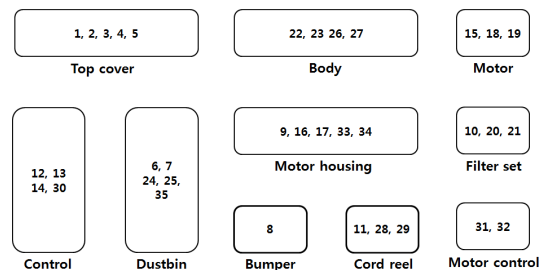
모듈의 구성과 크기, 개수는 제품에 대한 요구사항에 따라 결정되는데, 이는 모듈을 합치거나 분리함으로써 조정할 수 있다. 이와 같이 모듈을 수정할 때 모듈간 상호작용 정보를 활용하여 유연하게 모듈을 설계할 수 있다[5]. 예를 들어, Fig. 3에서 컴포넌트 2와 5, 컴포넌트 4와 5를 모듈간 상호작용이 있는 것으로 나타냈는데 이는 진공청소기의 물리적, 기능적 인터페이스가 유지되는 한 제품의 정상적인 작동과 기능에 영향을 미치지 않으므로 모듈의 개수를 줄이거나 모듈의 크기를 키워야 할 경우에 모듈(1, 5)와 모듈(2, 3, 4)를 합쳐서 하나의 모듈(1, 2, 3, 4, 5)로 재설계 할 수 있다. 이와 같이 모듈 간 상호작용 정보를 활용하여 모듈의 개수가 많게는 15개에서 적게는 7개로 줄일 수 있겠다.

본 연구에서 제시한 컴포넌트 군집 절차의 특성을 알아보기 위해 본 논문의 사례연구 결과와 Kreng 과 Lee[6]의 사례 연구결과를 비교하고자 한다. Fig. 4는 본 연구에서 활용한 진공 청소기‘TK-929’의 사례에 Kreng 과 Lee[6]가 제시한 유전자 알고리즘을 기반으로 한 군집절차를 이용하여 얻은 결과이다.

본 논문의 사례연구 결과에 대한 목적식 값은 Table 9와 같이 287.5이며, Kreng 과 Lee[6]의 사례연구 결과에 대한 목적 값은 Fig. 4의 결과를 본 논문에서 제시한 수리모형 (2)~(5)에 대입하여 그 값이 257임을 알 수 있

다. 목적 값이 클수록 하나의 모듈로 군집된 컴포넌트들 간의 유사성이 크며, 모듈 간 상호작용은 작다는 것을 의미한다. 본 논문에서 제시한 네트워크 흐름 모델을 활용한 군집방법은 최적해를 보장하므로 Kreng 과 Lee[6]가 제시한 유전자 알고리즘을 기반으로 한 군집방법이 제시한 해보다 우수한 것을 알 수 있다.

또한, 본 논문에서 제시하는 네트워크 흐름 모델을 활용한 컴포넌트 군집 절차는 모듈러 제품 설계에서 가장 기본적이고 주요한 최적의 모듈들을 찾아내고, 본 절의 앞에서 언급하였듯이 이들 모듈 간 상호작용 지표를 활용하여 보다 유연하게 모듈을 설계하고 수정할 수 있도록 제품 개발 관련 부서와 개발 전문가에게 정보를 제공하여 제품 수정 유연성과 제품 다양성 증대에 기여하는 바이다.

**Fig. 3.** The result of clustering(modules) and interfaces**Fig. 4.** The result of Kreng & Lee's case study

4. 결론

본 연구에서는 이미 시장에 출시되어 있는 제품을 새로운 시장에 출시하기 위하여 모듈 기반의 Bottom-up 제품군 개발방식으로 접근하여 그 시장의 요구사항에 맞게 새롭게 컴포넌트들을 모듈화하기 위해 네트워크 흐름 모델을 활용한 컴포넌트 군집 절차를 제시하였다. 컴포

넛트들을 군집하기 위한 기준으로 컴포넛트들 간의 상관관계 매트릭스를 만들고, capacitated-circulation 넛트워크 흐름 모델을 수립하였다. Matlab의 최적화 도구를 활용하여 군집을 형성하였다.

본 논문에서 제시한 군집 절차의 유효성을 입증하기 위해 A기업의 진공 청소기 사례에 넛트워크 흐름 모델을 활용한 군집 절차를 적용하였다. 35개의 컴포넛트에 대해 각각 물리적, 기능적 상관관계와 물리적, 기능적 상관관계의 중요도를 측정하여 군집 기준이 되는 하나의 최종 상관관계 매트릭스를 만들었다. 본 논문에서 제시한 넛트워크 흐름 모델을 통하여 15개의 군집들이 15개의 모듈들을 형성할 수 있고 모듈 간 인터페이스에 대한 정보를 나타냄으로써 모듈 설계에서 기본적으로 주요한 모듈들을 파악할 수 있었다.

또한, 동일한 사례를 다룬 Kreng 과 Lee[6]의 유전자 알고리즘 기반의 군집절차와 본 논문에서 제시한 군집절차가 각각 구한 값을 비교함으로써 본 논문의 군집절차가 최적 해를 찾는다는 것을 확인하였고, 모듈간 상호작용 정보를 활용하여 유연하게 모듈을 설계하고 수정할 수 있도록 제품 개발 관련 부서와 개발 전문가에게 정보를 제공하는 것을 알 수 있었다.

향후 연구 분야로, 기업의 마케팅부서와 개발부서, 생산부서가 협업하여 제품 개발을 수행하는 보다 현실적인 개발환경을 반영하기 위해 컴포넛트 군집 기준에 고객의 요구사항이나 제조 설비 라인 등과 같은 마케팅과 생산 부서의 입장을 동시에 고려할 수 있는 컴포넛트들 간의 상관관계 지표 개발이 필요해 보인다. 또한 넛트워크 흐름 문제를 푸는 알고리즘을 탐색하거나 개발하여 본 논문에서 제시한 컴포넛트 군집 절차가 얼마나 큰 크기의 문제를 다룰 수 있는지 실험해 보는 것도 상당히 의미 있는 연구주제로 보인다. 마지막으로, 본 논문에서 제시한 군집 절차의 결과를 토대로, 이미 형성된 군집 간 상관관계를 측정하여 3장의 사례에서 보였듯이 여러 군집을 재 군집하거나 또는 분리하는 등 보다 유연한 모듈설계를 위한 체계적인 절차를 수립하는 작업도 상당히 흥미로운 연구주제로 보인다.

References

- [1] ElMaraghy, H., Schuh, G., ElMaraghy, W., Piller, F., Schönsleben, P., Tseng, M., & Bernard, A. (2013). Product variety management. *CIRP Annals-Manufacturing Technology*, vol. 62, no. 2, pp. 629-652. DOI: <http://dx.doi.org/10.1016/j.cirp.2013.05.007>
- [2] Kidd, Paul T., "Agile manufacturing: a strategy for the 21st century", *Agile Manufacturing (Digest No. 1995/179), IEE Colloquium on. IET*, 1995. DOI: <http://dx.doi.org/10.1049/ic:19951097>
- [3] Jiao, J., &Tseng, M. M., "Understanding product family for mass customization by developing commonality indices," *Journal of Engineering Design*, vol. 11, no. 3, pp.225-243, 2000. DOI: <http://dx.doi.org/10.1080/095448200750021003>
- [4] Gu, P., Hashemian, M., Sosale, S., & Rivin, E., An integrated modular design methodology for life-cycle engineering. *CIRP Annals-Manufacturing Technology*, vol. 46, no. 1, pp. 71-74, 1997. DOI: [http://dx.doi.org/10.1016/S0007-8506\(07\)60778-1](http://dx.doi.org/10.1016/S0007-8506(07)60778-1)
- [5] Gu, P., &Sosale, S., "Product modularization for life cycle engineering," *Robotics and Computer-Integrated Manufacturing*, vol. 15, no. 5, pp.387-401, 1999. DOI: [http://dx.doi.org/10.1016/S0736-5845\(99\)00049-6](http://dx.doi.org/10.1016/S0736-5845(99)00049-6)
- [6] Kreng, V. B., &Lee, T., "Modular product design with grouping genetic algorithm—a case study," *Computers &Industrial Engineering*, vol. 46, no. 3, pp. 443-460, 2004. DOI: <http://dx.doi.org/10.1016/j.cie.2004.01.007>
- [7] Eppinger, S.D. &Ulrich, K.T., *Product design and development*, McGraw Hill New York, 1995.
- [8] Ulrich, K., *Fundamentals of product modularity*. p. 219-231, Springer, 1994. DOI: http://dx.doi.org/10.1007/978-94-011-1390-8_12
- [9] Huang, C., "Overview of modular product development," *Proceedings-National Science Council Republic of China Part a Physical Science and Engineering*, vol. 24, no. 3, pp.149-165, 2000.
- [10] Jose, A., &Tollenaere, M., "Modular and platform methods for product family design: Literature analysis," *Journal of Intelligent Manufacturing*, vol. 16, no. 3, pp.371-390, 2005. DOI: <http://dx.doi.org/10.1007/s10845-005-7030-7>
- [11] Pimmler, T. U., &Eppinger, S. D., "Integration analysis of product decompositions," 1994.
- [12] Sa'ed, M. S., &Kamrani, A. K., "Macro level product development using design for modularity," *Robotics and Computer-Integrated Manufacturing*, vol. 15, no. 4, pp.319-329, 1999. DOI: [http://dx.doi.org/10.1016/S0736-5845\(99\)00008-3](http://dx.doi.org/10.1016/S0736-5845(99)00008-3)
- [13] Yu, T. L., Yassine, A. A., & Goldberg, D. E., "An information theoretic method for developing modular architectures using genetic algorithms," *Research in Engineering Design*, vol. 18, no. 2, pp.91-109, 2007. DOI: <http://dx.doi.org/10.1007/s00163-007-0030-1>
- [14] Yu, S., Yang, Q., Tao, J., Tian, X., & Yin, F. "Product modular design incorporating life cycle issues-Group Genetic Algorithm (GGA) based method," *Journal of Cleaner Production*, vol. 19, no. 9, pp. 1016-1032, 2011. DOI: <http://dx.doi.org/10.1016/j.jclepro.2011.02.006>
- [15] Stone, R. B., Wood, K. L., & Crawford, R. H., "A heuristic method for identifying modules for product

architectures,” *Design studies*, vol. 21, no. 1, pp. 5-31, 2000.

DOI: [http://dx.doi.org/10.1016/S0142-694X\(99\)00003-4](http://dx.doi.org/10.1016/S0142-694X(99)00003-4)

- [16] Erixon, G., von Yxkull, A., & Arnstroem, A., “Modularity - the basis for product and factory reengineering,” *CIRP Annals-Manufacturing Technology*, vol. 45, no. 1, pp.1-6, 1996.
DOI: [http://dx.doi.org/10.1016/S0007-8506\(07\)63005-4](http://dx.doi.org/10.1016/S0007-8506(07)63005-4)
- [17] Hölttä, K., Tang, V., & Seering, W. P., “Modularizing product architectures using dendrograms,” *DS 31: Proceedings of ICED 03, the 14th International Conference on Engineering Design, Stockholm*, 2003.
- [18] Ericsson, A. & Erixon, G., *Controlling design variants: modular product platforms*, Society of Manufacturing Engineers, 1999.

손 지 양(Jiyang Son)

[준회원]



- 2014년 8월 : 동아대학교 경영학과 (경영학사)
- 2014년 8월 ~ 현재 : 동아대학교 일반대학원 경영학과 (경영학석사 과정)

<관심분야>

제품개발, 경영과학, 최적화

유 재 욱(Jaewook Yoo)

[정회원]



- 1990년 2월 : 한양대학교 산업공학과 (공학사)
- 1992년 2월 : 한양대학교 일반대학원 산업공학과 (공학석사)
- 2004년 5월 : Texas A&M Univ. 산업공학과 (공학박사)
- 2004년 4월 ~ 2011년 8월 : 삼성 전자 책임연구원

- 2011년 9월 ~ 현재 : 동아대학교 경영학과 교수

<관심분야>

경영과학, 제품개발 전략 및 계획, 공급사슬관리