ICED13
INTERNATIONAL CONFERENCE ON ENGINEERING DESIGN, ICED13
19-22 AUGUST 2013, SUNGKYUNKWAN UNIVERSITY, SEOUL, KOREA

AN ANALYSIS OF DECOMPOSITION APPROACH APPLICATIONS IN DESIGN ENGINEERING & SUGGESTIONS FOR IMPROVEMENT

Junfeng MA, Gul KREMER
The Pennsylvania State University, United States of America

ABSTRACT

Modular design attracts great attention because of widely implementation in industry and academe. The benefits of modular design include shorter assembly process, easier manufacturing and cheaper maintenance. The modular design also connects both the gains of standardization and customizations since it can reponse to market requirements rapidly. Therefore, many authors concentrate on developing modular design methods. The most widely used method is decomposition approach and one of this type method is classic decomposition approach which proposed by Huang and Kusiak in 1998. Classic decomposition approach provides a new perspective of modular design with designer's desire taken into account, however, there are still some potential improvements for this approach. In this paper, we figure out several limitations of CDA (classic decomposition approach) first and then the revised algorithm is offered regarding to some of these limitations.

Keywords: design engineering, design methodology, product architecture, modular design, decomposition approach

Contact:
Junfeng Ma
The Pennsylvania State University
Industrial Engineering
state college
16801
United States of America
jom5367@psu.edu

1 INTRODUCTION

Modular design, as its name suggests, subdivides a complex system or a product into several smaller parts or modules which can be treated independently, and then combines these modules based on varied design criteria to form different products or systems. Modular design has become more widely used in industry due to its shorter assembly process, easier manufacturing and cheaper maintenance (Okudan et al., 2012), especially for products with huge numbers of components, such as automobiles and aircrafts. Modular design is also an attempt to connect both the gains of standardization and customization (Gershenson et al., 2004) since the concept of modularity can provide the necessary foundation to design products or systems which can respond rapidly to market requirements and simultaneously allow changes in product or system design with cost savings. Moreover, modular design also benefits supply chain operations. Modularity reduces product development time and order lead time because it relies on dividing a product into components with a clear definition of the interfaces which permits performance of the design tasks concurrently, where standard components are inventoried (Kamrani and Salhieh, 2008).

In this paper, we first review the developments in modular design from a methodological perspective, and then focus on the Decomposition Approach (DA), one of the most widely implemented modularization methods. We specifically analyze the potential limitations of DA followed by suggestions for its improvement.

2 LITERATURE REVIEW IN MODULAR DESIGN

Zhang and Gershenson (2003) categorized modular design methods into two groups: (1) matrix based, and (2) function based. Matrix-based methods group components into separate modules using clustering methods implemented in a matrix. Function-based methods require some intrinsic knowledge of the complex product or system to direct function identification and decomposition.

Numerous matrix-based methods have evolved. Kusiak and Chow (1987) developed a cluster identification algorithm and a cost analysis algorithm to group components. In order to increase design efficiency, Kusiak and Wang (1993) developed the triangularization algorithm based on depth-first search and applied this method as well as the decoupling algorithm to group and optimize modules. Newcomb et al. (1998) defined two indexes: CR (Correspondence Ratio), and CI (Cluster Independence) to measure modularity and applied the cluster identification algorithm to redesign products. Gu et al. (1997) took several design criteria into account and used the integrated modular design methodology for life cycle engineering. Huang and Kusiak (1998) modified the triangularization algorithm by considering interaction and suitability matrices. Pimmler and Eppinger (1994) applied the heuristic swapping algorithm to measure interaction among components using five levels (-2 to 2), divided the interactions into four types and categorized them based on those interactions.

Among the prominent function-based methods are the following. Ishii et al. (1995) used the fishbone diagram to represent the relationships among modules. Marshall et al. (1998) investigated whether corporate goals and product requirements were matched or not in modular design. Stone et al. (1998) used the function structure diagram to identify dominant flows, branching flows and conversion transmission flows; where each flow is a potential module or module type.

The matrix-based modular design methods focus on the similarities and differences among components, but these methods give limited attention to the function level relationships. For example, Kusiak and Chow (1987) developed two matrix-based algorithms using only the perspective of components' physical relationships. In contrast, function-based modular design methods concentrate on functional or group relations while mostly ignoring component level properties. For example, Marshall et al. (1998) considered only functional criteria to check the implications of the modular design. Until recently, no method considered these two factors simultaneously. Table 1 summarizes selected published examples of matrix-based and function-based modular design methods.

In general, development of the modularity methods coincides with the increased popularity of mass customization and product families as modular designs enable increased product variety. Among the presented approaches in Table 1, most of the matrix-based approaches can be seen as decomposition approaches in their essence, and hence, we focus our analysis on this group.

Table 1 Modular Design Methods

Authors /Date	Type	Method	Merits	Limitations
Kusiak and Chow (1987)	Matrix-Based	Cluster Identification Algorithm + Cost Analysis Algorithm	Develop simpler group methods	No guarantee for more complex modular design problems
Kusiak and Wang (1993)	Matrix-Based	Triangularization Algorithm + Decoupling Algorithm	Two methods increase design efficiency	No guarantee for all cases
Pimmler and Eppinger (1994)	Matrix-Based	Heuristic Swapping Algorithm	Five levels interactions (-2~2); four types interactions(spatial, energy, information, materials); one entry with four numbers	Ignores similarity; no guarantee for the final form
Gu, Hashemian and Sosale (1997)	Matrix-Based	Simulated Annealing Algorithm	Integrated modular design methodology for life cycle engineering	No similarity notes among modules and component incompatibility
Huang and Kusiak (1998)	Matrix-Based	Triangularization Algorithm + deletion+duplication	Two matrices: interaction and suitability matrix	See Section 5
Newcomb, Bras and Rosen (1998)	Matrix-Based	Cluster Identification Algorithm	Two indexes CR (correspondence ratio) and CI (Cluster Independence)	The algorithm may not be feasible with no intermediate steps to allow a designer to move to a better design
Ishii, Juengel and Eubanks (1995)	Function-Based	Fishbone Diagram	Fishbone diagram represents module relationships	May not apply in other life cycle applications except product recycling
Marshall (1998)	Function-Based	Holonic Product Design Method	Check the matches of corporate goals and product requirements	No specific modularization method
Stone, Wood and Crawford (1999)	Function-Based	Heuristic Method+ Function Structure Diagram	Module identification is unique; each function is a potential module or module	The heuristic approach is not easy to quantify

3 DECOMPOSITION APPROACH AND RELEVANT PRIOR STUDIES

Decomposition Approach is one of most widely used modular design methods. Initially, this approach was proposed by Steward (1965) who introduced the philosophy of system partition and testing. Eppinger et al. (1990) used matrix representation to capture both sequence and technical relationships among many design tasks to be performed. The relationships are analyzed in order to find alternative sequences and/or definitions of the design tasks. Kusiak and Park (1990) developed two methods for decomposition of design activities: (1) one based on the product structure, and (2) one based on the precedence relationship between activities. Both of these allow effective organization of resources required in the design process and simplify the management of design activities. Selected published cases of the DA approach in products are summarized in Table 2.

Table 2 Cases Studies for Decomposition Approach

Authors	Product	Compt. Numbers	Group Technology
Pimmler and Eppinger (1994)	Automotive Climate Control System	16	Heuristic Swapping Algorithm
Gu, Hashemian and Sosale (1997)	Vacuum Cleaner	24	Simulated Annealing Algorithm
	Starter	25	
Huang and Kusiak (1998)	Electrical Product	14	Triangulization Algorithm;
	Mechanical Product	14	
Newcomb, Bras and Rosen (1998)	Center Console in Chrysler LHS	19	Cluster Identification Algorithm
Sosa, Eppinger and Rowles (2000)	Large Commercial Aircraft Engine	54	Heuristic Swapping Algorithm
Okudan, Lin and Chiu (2011)	Refrigerator	26	Triangulization Algorithm+ Multivariate Cluster Application

The case studies in Table 2 adapted DA-based modular design methods with respect to different design criteria. Huang and Kusiak (1998) considered two different criteria: one is component-based, and the other is designer's judgment-based. Since this method takes both the product itself and the design into account, it is most widely used. We discuss this method in detail in the section 4.

4 CLASSIC DECOMPOSITION APPROACH AND A CASE STUDY

Huang and Kusiak (1998) contributed to the application and method development for decomposition approach by taking two matrixes into account: the interaction matrix and the suitability matrix. The interaction matrix represents physical characteristics of components, and the suitability matrix shows how other factors affect components, such as designer preferences and cost consideration. Following are the steps of Huang and Kusiak's classic decomposition approach (referred to as CDA below).

Step 0: Initialization: Initialize the interaction and suitability matrices. Specify the upper bound N_V on the number of components in a module and budget B.

Step 1: Triangularization: Triangularize the interaction matrix A into matrix $A^{'}$ using triangularization algorithm.

Step 2: Rearrangement: Rearrange the suitability matrix B into matrix $B^{'}$ so that sequence of columns and rows in matrix $B^{'}$ is the same as in matrix $A^{'}$.

Step 3: Combination: Combine the matrix $A^{'}$ and the matrix $B^{'}$ into the modularity matrix $(A^{'}|B^{'})$. Identify modules corresponding to the groups in $A^{'}$. In suitability matrix, A means strongly desired, O means strongly undesired, E means desired and U means undesired.

Step 4: Deletion: Remove a component from a module if it satisfies Condition 1, and place it in the last column of the modularity matrix. Repeat this step until no more components can be removed.

Step 5: Duplication: Duplicate a component that satisfies Condition 2, and repeat this step until no more components can be duplicated.

Step 6: Classification: Analyze the modularity matrix to classify the modules.

Step 7: Termination: Stop and output the results.

Condition 1: Remove a component k, if the following conditions are satisfied.

1) Component k and any other component l that appear in the same module are strongly undesired for inclusion in the module.

2) Component k interacts with the remaining components in the module to a lesser degree than component l.

3) None of the sub-matrices violates constraints C1 and C2.

Condition 2: Duplicate the component if the following conditions are satisfied.

1) The component that is used and strongly desired for inclusion in two modules simultaneously.
2) None of the sub-matrices violates constraints C1 and C2.
Constraint C1: Empty modules of components are not allowed,
Constraint C2: The number of components in a module cannot exceed the upper bound N_V, and the total cost of the components duplicated cannot exceed B (Adapted from Huang and Kusiak, 1998).
Okudan et al. (2012) provided a case study implementing CDA after dissecting a refrigerator. Table 3 shows parts of the dissected refrigerator. By applying the CDA algorithm, the corresponding modules in both the interaction and suitability matrices are derived, which are shown in Figure 1.

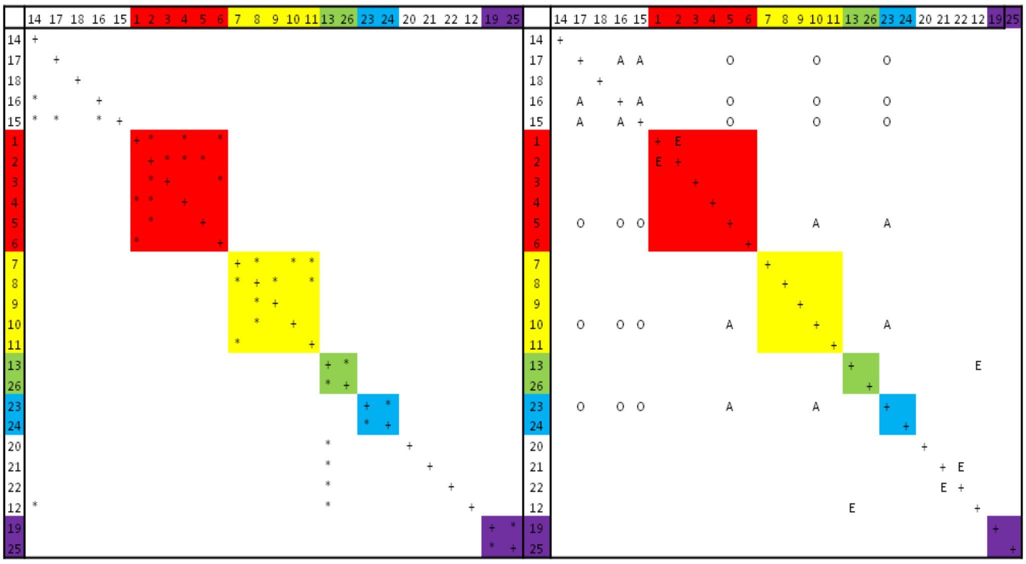

Figure 1 Interaction and Suitability Matrixes

The interaction matrix is setup based on physical interaction considerations, and the suitability matrix is formed by taking components' end of life options into account. The modules in the suitability matrix correspond to the modules in the interaction matrix.

Table 3 Major Parts of the Dissected Refrigerator

Main Structure	Major Part	# of Comp	Part #	Material	EOL Options	Weight (g)
Left-Hand Side Door (with water/ice supply system)	Outer Housing (L)	1	1	Al&Steel	Recycle	2,693
	Inner Housing (L)	1	2	Plastic	Recycle	2,236
	Inner Partitions (L)	3	3	Plastic	Recycle	40.7
	Water Supply Parts (L)	1	4	Plastic	Recycle	1028
	Rubber Strip (L)	1	5	Rubber	Disposal	50
	Handle (L)	1	6	Plastic	Recycle	287.4
Right-Hand Side Door	Outer Housing (R)	1	7	Plastic	Recycle	4,331
	Inner Housing (L)	1	8	Plastic	Recycle	4,472
	Inner Partitions (L)	4	9	Plastic	Recycle	351.2
	Rubber Strip (L)	1	10	Rubber	Disposal	60
	Handle (L)	1	11	Plastic	Recycle	29
Main Body	Housing (M)	1	12	Steel	Recycle	23,606
	Inner Housing (M)	1	13	Plastic	Recycle	26,313
Cooling System	Base Pan (B2)	1	14	Plastic	Recycle	1,240
	Compressor	1	15	Multiple Material	Reuse	7,985
	Dryer	1	16	Multiple Material	Reuse	111
	Condenser	1	17	Multiple	Reuse	2,669

				Material		
	Fan	1	18	Plastic	Recycle	483

Table 4 (continued) Major Parts of the Dissected Refrigerator

Main Structure	Major Part	# of Comp	Part #	Material	EOL Options	Weight (g)
Evaporator and Inner Partition	Evaporator	1	19	Plastic	Recycle	532
	Water Tank	1	20	Plastic	Recycle	1,412
	Shelves	5	21	Plastic	Recycle	1,000
	Crisper (115*2)	2	22	Plastic	Recycle	266
	Auger Motor	1	23	Multiple Material	Reuse	483
	Relay Capacitor	1	24	Plastic	Recycle	177
	Evaporator Cover	1	25	Plastic	Recycle	897
	Back Inner	1	26	Steel	Recycle	986

5 LIMITATIONS OF THE CLASSIC DECOMPOSITION APPROACH

Although the CDA method by Huang and Kusiak (1998) is applied widely in academia and industry, it has some limitations. Its major limitations are listed below:

1. The entries in the interaction matrix are binary (0 or 1) (or can be replaced by empty cells or stars) which may not reflect the actual interaction level. As shown in Figure 1, component 1 (Outer Housing) and component 4 (Water Supply Parts) have a significant amount of physical interaction (contact area), while for its interaction with component 6 (Handle), component 1 has only a very small area. However, in Huang and Kusiak's method, all interactions are represented by star (or binary number), which does not reflect real interaction levels. Figure 2 shows this situation.
2. Suitability matrix may not contribute to the module forming in the interaction matrix, because the strongly desired or strongly undesired components may not exist in the same modules corresponding to the interaction matrix. For example, in Figure 1, the strongly desired or strongly undesired entries are not in the same module within the suitability matrix; thus they have no effect on the interaction matrix.

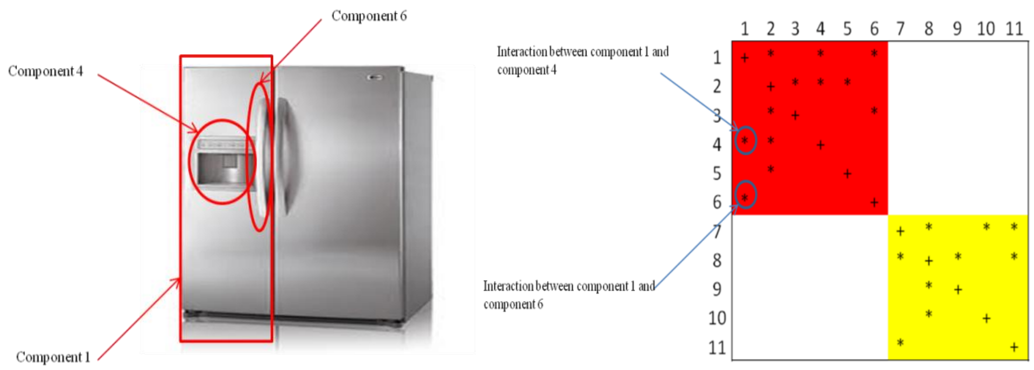

Figure 2 Actual Interactions and Interaction Matrix

3. The original CDA does not provide an assignment method for strongly desired components in the suitability matrix. In the red and yellow modules of Figure 1, components 5 and 10 are strongly desired. However, they belong to two separate modules, and there is no provision in the CDA method to handle this case. This case is shown in Figure 3.
4. The deletion step may not always work since condition 1 may not cover all strongly undesired situations. Using the case study from section 4 as an example we show this limitation. Assume that the interaction between components 1 and 2 are strongly undesired based on manufacturing considerations; the suitability matrix for this case is shown in Figure 4. Note that Figure 4 only shows the red module without showing others. Component 1 and component 2 are strongly undesired, which satisfies rule 1 of condition 1. However, both component 1 and component 2

have three interactions with the remaining components in the module, thus rule 1 of condition 2 is not satisfied. Therefore, deletion step cannot be applied in this case.

5. The duplication step may not always work either, since condition 2 does not cover all strongly desired situations, and there might be additional cases beyond the definition of condition 2. Let's revisit the case study in Figure 1 as an example. Assume component 8 is strongly desired to be with both component 9 and component 13, but strongly undesired to be co-located with component 26 based on the end-of-life considerations. This case is shown in Figure 5 with the appropriate updates in the suitability matrix. Duplication step cannot be applied in this case because duplicating and inserting component 8 into the same module with components 13 and 26, strongly undesired co-location condition of components 8 and 26 is violated.

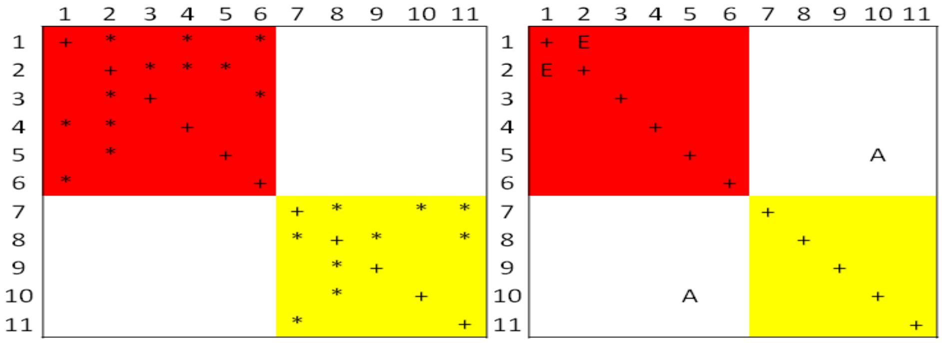

Figure 3 Strongly Desired Components in Two Modules

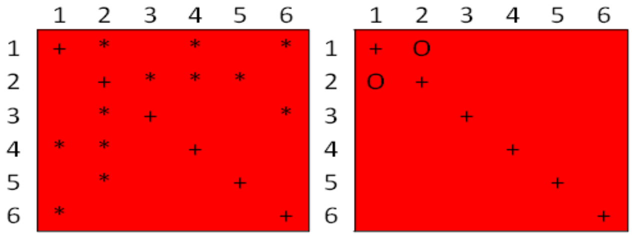

Figure 4 Anti-Deletion Case

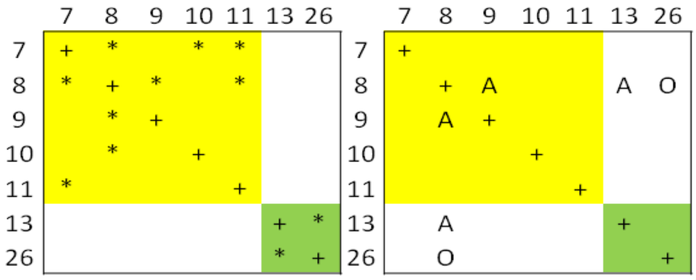

Figure 5 Anti-Duplication Case

6. Suitability matrix considers only one suitability factor, such as preferences of the design decision maker. In real life cases, many factors need to be specifically taken into account to reach a decision.
7. The cycle forming process in the triangularization algorithm is depth-first search, which may not work in some cases, especially when entry density is very high. In the example of Figure 6, the entry density is high, the module formed by triangularization algorithm includes all entries, and suitability matrix does not contribute to the module forming, which is not useful in practice since putting all entries into one module is equal to putting no entries into one module.

6 A PRELIMINARY REVISION OF CLASSIC DECOMPOSITION APPROACH ALGORITHM

A revised algorithm has been developed based on the consideration of limitations presented in section 5. Since modification of all seven limitations might further present new problems (e.g., how to guarantee one algorithm solves both limitations 1 and 7), this revised algorithm only considers limitations 3, 4, 5 and 6. The steps are the following:

Step 0: Initialization: Initialize the interaction and suitability matrices. The suitability matrix is based on first priority suitability factor. Specify the upper bound N_V on the number of components in a module and budget B.

	1	2	3	4	5	6
1	+	*		*		*
2		+	*	*	*	
3		*	+			*
4	*	*		+		
5		*			+	
6	*					+

	1	2	3	4	5	6
1	+	E				
2	E	+				
3			+			
4				+		
5					+	
6						+

Figure 6 Anti-Case of Cycle Forming

Step 1: Triangularization: Triangularize the interaction matrix A into matrix A' using triangularization algorithm.

Step 2: Rearrangement: Rearrange the suitability matrix B into matrix B' so that sequence of columns and rows in matrix B' is the same as in matrix A'.

Step 3: Combination: Combine the matrix A' and the matrix B' into the modularity matrix ($A'|B'$). Identify modules corresponding to the groups in A'. In suitability matrix, A means strongly desired, O means strongly undesired, E means desired and U means undesired.

Step 4: Re-module: Regroup the components based on consideration of strongly desired as per the suitability matrix and minimization of information loss in the interaction matrix. Make sure that strongly desired components should be in the same module.

Step 5: Deletion: Remove a component from a module if it satisfies Condition 1, and place it in the last column of the modularity matrix. Repeat this step until no more components can be removed.

Step 6: Duplication: Duplicate a component that satisfies Condition 2, and repeat this step until no more components can be duplicated.

Step 7: Division: Divide the module into sub-modules if it satisfies condition 3. Repeat this step until no more modules can be divided.

Step 8: Classification: Analyze the modularity matrix to classify the modules.

Step 9: Termination: Stop and output the results. Decouple the components in each module. Then use the module matrix as interaction matrix, and form suitability matrix for second priority suitability factor, and repeat steps 1 to 9 until all suitability factors are considered.

Condition 1: Remove a component *k*, if the following conditions are satisfied.

1) Component *k* and any other component *l* of the same module are strongly undesired for inclusion in the module.

2) Component *k* interacts with the remaining components in the module to a lesser degree than component *l*.

3) None of the sub-matrices violates constraints C1 and C2.

Condition 2: Duplicate the component if the following conditions are satisfied.

1) The component that is used and strongly desired for inclusion in two modules simultaneously; and none of the sub-matrices violates constraints C1 and C2.

2) In condition 1, rule 1) if both component *k* and *l* have the same number of interactions with remaining components in the module, and component *k* and *l* have common entries, duplicate these common entries, and assign them into both *k* module and *l* module.

*Condition 3***:** Divide the module if following conditions are satisfied.

1) In condition 1, rule 1) if both components *k* and *l* have same number of interactions with remaining components in the module and component *k* and *l* have no common entries, divide this module based on component *k* and *l*.

2) None of the sub-matrices violates constraints C1 and C2.

Constraint C1: Empty modules of components are not allowed.

Constraint C2: The number of components in a module cannot exceed the upper bound N_V, and the total cost of the components duplicated cannot exceed *B*.
From the revised algorithm, the limitation examples in Figures 3-5 can be solved as the following.

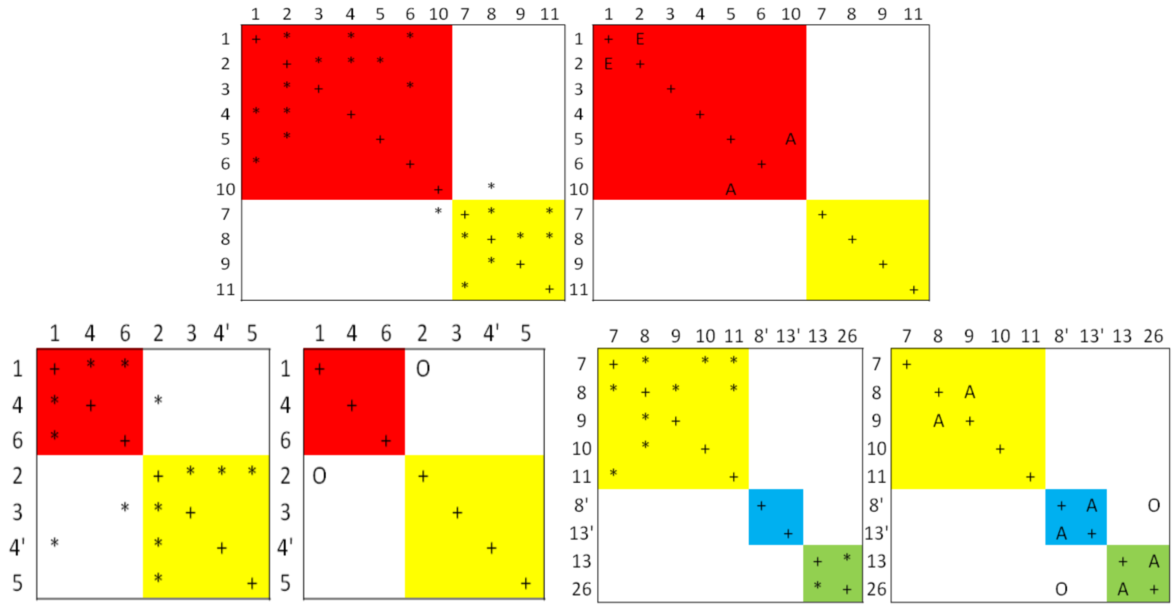

Figure 7 Revised Examples corresponding to Fig 3~5

In Fig 3, component 1~6 form a module under the original CDA; when applying revised decomposition approach (upper section of Fig 7), component 1~6 and component 10 are grouped into one module by considering original constraints that component 5 and 10 are strongly desired in the suitability matrix of Fig 1. In Fig 4, under the assumption that components 1 and 2 are strongly undesired, the original CDA still puts components 1~6 into one module, but when applying the revised approach, components 1, 4 and 6 are in one module, and components 2, 3, 4' and 5 are located in another module. This case is shown on the lower left section of Fig 7. In Fig 5, based on the assumption that component 8 is strongly desired to be co-located with both components 9 and 13, but strongly undesired to be modularized with component 26, the original CDA suggests that component 7~11 should be grouped in one module, component 13 and 26 are in another module. When using the new approach, shown in the lower right section of Fig 7, component 7~11 are in one module, 8' and 13' are in another modular, and 13 and 26 are in a third module. It is obvious that adapting the revised decomposition approach could make product modules more reasonable adhering to design requirements.

7 CONCLUSION AND FUTURE WORK

The classic decomposition approach developed by Huang and Kusiak (1998) is broadly used in industry and academia. Originally, this approach provided a new perspective to the modular design area by taking designer's judgment into account. However, from the discussion presented in this paper, it is seen that there are several limitations with it; the classical decomposition approach has room for potential improvements. This is the motivation of this research. It is difficult to eliminate all limitations in one revision because combining possible solutions for some limitations may lead to new limitations. For example, limitation 1, as discussed, relates to binary representations in both interaction and suitability matrices, and limitations 3~6 mention several missing coordination issues between interaction and suitability matrices. If we replace binary representations in the interaction matrix with numerical representations, and correct limitations 3~6 by methods mentioned in section 6, new limitations might come up. Therefore, we separate our revision of the method to correct the limitations in order to simplify the problem. The revision one of the classic decomposition approach only concentrates on limitations 3~6 with no attention to limitations 1, 2, and 7. Some possible future works include replacing binary entries by numerical entries and developing a new algorithm based on numerical entries, developing a new clustering or grouping method to form cycles, and trying to connect these algorithms together with no conflicts.

REFERENCE

Eppinger, S.D, Whitney, D.E, Smith, R.O. and Gebala, D.A. (1990) Organizing the tasks in complex design projects. *ASME conference-Design Theory Methodology,* Chicago, IL, Sep.16-19, New York: ASME, pp. 39-46.

Gershenson, J.K, Prasad, G.J. and Zhang, Y. (2004) Product Modularity: Measures and Design Methods. *Journal of Engineering Design*, Vol.15, No.1, pp. 33-51.

Gu, P., Hashemian, M. and Sosale, S. (1997) An integrated modular design methodology for life cycle engineering. *Annals of the CIRP*, Vol.46, No.1, pp.71–74.

Huang, C.C. and Kusiak, A. (1998) Modularity in design of products and systems. *IEEE Transactions on Systems, Man, and Cybernetics, Part A, Systems and Humans*, Vol.28, No.1, pp.66–77.

Ishii, K., Juengel, C. and Eubanks, C. F. (1995) Design for product variety: key to product line structuring. *ASME Design Engineering Technical Conferences—7th International Conference on Design Theory and Methodology*, Boston, MA, Sep 17-20, New York: ASME, pp. 499-506.

Kamrani, A.K. and Salhieh,S.E.M. (2008) Modular Design. In Kamrani, A.K. and Salhieh,S.E.M, (ed[s]), *Collaborative Engineering: Theory and Practice,* Springer, pp.207-227.

Kusiak, A. and Chow, W.S. (1987) Efficient solving of the group technology problem. *Journal of Manufacturing Systems*, Vol.6, No.2, pp.117–124.

Kusiak, A. and Park, K. (1990) Concurrent engineering: Decomposition and scheduling of design activities. *International Journal of Production Research*, Vol. 28, No. 10, pp. 1883-1900.

Kusiak, A. and Wang, J. (1993) Efficient organizing of Design Activities. *International Journal of Production Research*, Vol. 31, No.4, pp.753-769.

Marshall, R., Leaney, P.G. and Botterell, P. (1998) Enhanced product realisation through modular design: an example of product/process integration. *Journal of Integrated Design and Process Technology*, Vol.3, No.4, pp.143–150.

Newcomb, P.J., Bras, B. and Rosen, D.W. (1998) Implications of modularity on product design for the life cycle. *Journal of Mechanical Design*, Vol.120, No.3, pp.483-491.

Okudan, G.E, Lin, T.K. and Chiu, M.C. (2012) Carbon Footprint Implications of Modularity and Projections for the Reverse Logistics. *International Workshop on Green Supply Chain*, Arras, France, Jun 21-22.

Pimmler, T.U. and Eppinger, S.D. (1994) Integration analysis of product decompositions. *ASME Design Engineering Technical Conferences—6thInternational Conference on Design Theory and Methodology*, Minneapolis, MN, Sep.11-14, New York: ASME, pp. 343-351.

Sosa, M.E., Eppinger, S.D. and Rowles, C.M. (2000) Design modular and integrative systems, *ASME International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, Baltimore, Maryland, September 10-13.

Steward, D.V. (1965) Partition and Testing Systems of Equations. *Journal of the Society for Industrial and Applied Mathematics: Series B, Numerical Analysis*, Vol.2, No.2, pp. 345-365.

Stone, R.B., Wood, K.L. and Crawford, R.H. (1998) A heuristic method to identify modules from a functional description of a product. *ASME Design Technical Conferences—11th International Conference on Design Theory and Automation*, Las Vegas, NV.

Zhang, Y. and Gershenson, J.K. (2003) An initial Study of Direct Relationships between Life-Cycle Modularity and Life-Cycle Cost. *Concurrent Engineering*, Vol.11, No.2, pp.121-128.