NordDesign 2016
August 10 – 12, 2016
Trondheim, Norway

Trade-offs Among System Architecture Modularity Criteria

Roozbeh Sanai[1], Kevin Otto[2], Kristin Wood[1], Katja Hölltä-Otto[2]

[1]SUTD-MIT International Design Centre, Singapore University of Technology and Design
roozbeh_sanaei@mymail.sutd.edu.sg
kristinwood@sutd.edu.sg

[2] Department of Engineering Design and Production, Aalto University
kevin.otto@aalto.fi
katja.holtta-otto@aalto.fi

Abstract

Modularity is a key design research area of the Nordic countries, with a history of intensive work over the last twenty years. In the last decade, there has been considerable research in product modularity, measuring the level of modularity and various procedures for searching for ideal modular architectures. Different manual heuristics and computer clustering algorithms have been developed to search ideal modular architectures by optimizing a modularity metric. However, the different criteria can be in conflict and improving one criteria may drive another infeasible, without an appropriate compromising effect or insights into decoupling conflicts / contradictions. We pose here the research question as to how to visualize product architectural design criteria and trade-offs in the early conceptual or configurational design phase. We analyze correlations between different system architecture modularity criteria provided in the research field, namely the intra-cluster, extra-cluster costs, number of modules and the variance in the size of modules. We demonstrate that these criteria trade-off with each other, and therefore one cannot be improved without affecting the other. We also show that several of these metrics are directly correlated; for example, the variance in the size of modules can be controlled through the intra-cluster cost. Finally we observe that, although typically proposed agglomerative or divisive hierarchical clustering algorithms might able to obtain optimal architecture when only extra-cluster cost is of concern, such algorithms are not able to find optimal cluster when both extra-cluster and intra-cluster cost are matter of interest. Overall, as minimizing intra-cluster cost is in charge of proper sizing of modules, well-sized modules cannot be obtained through conventional algorithms such as K-means clustering or similar.

Keywords: Product modularity, clustering algorithms, trade-off analysis

Nomenclature

Pareto-frontier	Pareto Frontier, is the subset of solutions including all solutions that at least one of their objectives is optimized while the other objectives are kept constant.
Utopia-Point	Solution that optimizes all objectivs, such a point often doesn't exist, but there are other solutions that keep other objectives constant while optimizing one.
Design Structure Matrix (DSM)	Matrix of Interactions between pairs of Components/Functions
ClusterSize	Number of Elements in Cluster
IntraClusterCost	Overall cost of Interactions between Elements that belong to the same cluster
ExtraClusterCost	Overall cost of Interactions between Elements that belong to different Clusters
TotalCost	Sum of all Intra and Extra-cluster costs
$DSM_{i,j}$	Interaction between elements i and j
$ClusterSize_y$	Number of elements in cluster y
Powcc	Exponent that penalizes the size of clusters in the formula for *TotalCost*
Powbid	Exponent that promotes the cluster size in the ClusterBid formula
Granularity	The extent to which a system is composed of identifiable elements.
Hierarchy	When every low-level module in the architecture is sub-module to another higher level module.

1 Introduction

Modularity is a key design research area of the Nordic countries, with a history of intensive work over the last twenty years. A *module* is an independent building block of a larger system with a specific or primary function and well-defined interface (Otto & Wood, 2001). The aim is to design the components of each module to be highly interdependent on each other, while having weak dependencies on elements outside the module (Baldwin & Clark, 2000). There are various benefits in using modular product architectures. The variety in customer needs of the marketplace can be responded to faster and more effectively without significant development effort in terms of time and cost (Du, Jiao, & Tseng, 2001; Gershenson, Prasad, & Zhang, 2003; Hölttä & Otto, 2005; Marshall & Leaney, 1999). New functionalities can be added by replacement of modules, parts replacement is far more convenient when they wear out, and updating the system is possible without swapping unnecessary components (Dahmus & Otto, 2001). Moreover, design teams can focus on design of certain modules which allows parallelization of design process. Modules designed for other products of a same product family or earlier designs of similar products can be utilized in new designs, significantly reducing design cycle time(Sanchez & Mahoney, 1996). Skania, Nokia, Volvo, and Electrolux are some of the many Nordic companies making use of modularity.

In practice, modularization procedures are implemented either by using manual heuristic rules (Stone, Wood, & Crawford, 1998, 2000a, 2000b; Sudjianto & Otto, 2001; Zamirowski & Otto, 1999), clustering algorithms, or a combination. Product architecture can be represented as a graph where nodes of graph are components or functions of the product and edges of graph are interactions between functions or components (Hirtz, Stone, McAdams, Szykman, & Wood, 2002; Stone & Wood, 2000). The adjacency matrix of such a graph would construct the Design Structure Matrix (DSM) of a product. Here we use DSMs as a representation since they are practicable when the system of interest is of considerable size and can be entered into computer algorithms easier.

In most modularization algorithms, having maximum inter dependencies within each module and minimum dependencies among modules are combined together in a single objective variable (Hölttä-Otto, Chiriac, Lysy, & Suk Suh, 2012). Another concern is the sizing of the modules, where it is often desirable to have more or less a similar number of components within modules, for supplier and logistic concerns. Generating many small modules connected to one or two large modules is not desired. We can also seek the minimum variance of module size as another objective. In this paper, we show how the objective on module size can be controlled easily within the commonly used DSM algorithms, by minimizing a variable in the algorithm, the *intra-cluster cost*. Doing so can allow a designer to interactively determine a trade-off on granularity of an architecture with an overall cost measure. The main point is to enable interactive trade-offs between the algorithm results and designer judgement.

The subsequent sections of this paper are organized as follows. First we introduce procedures to visualize trade-offs between criteria. We then demonstrate the use of the procedures on a vacuum cleaner product example, providing a discussion of trade-off decision making between different architectural measures.

2 Methodology

We propose to visualize a product architectural trade-off space of DSM alternatives through executing a stochastic hill climbing algorithm for a large number of times. This approach will stochastically generate a wide range of solutions in the neighborhood of the Pareto-frontier. A difficulty is that the parameters of the modularity optimization are discrete, and so the Pareto-frontier curve is not necessarily convex. Therefore executing a typical minimization of a weighted summation of the parameters can lead to an aggregation of points in certain regions of Pareto frontier and not others. The stochastic nature of the proposed hill climbing algorithm, however, leads to points over the entire Pareto frontier. This allows a designer to visualize trade-off between the parameters.

The IGTA+ algorithm developed by Börjesson (Borjesson & Hölttä-Otto, 2012) while working with Nordic companies and KTH appears to be a suitable algorithm for this procedure. IGTA+ is built upon IGTA (Idicula-Gutierrez-Thebeau Algorithm)[1], one of most commonly used algorithms in product modularization. IGTA+ randomly choses individual components and determines whether they must be moved into another module. Components will be moved to clusters if the fit between a selected component (later referred to as *ClusterBid*) and each of the existing clusters is improved. IGTA+ follows our desired stochastic hill-climbing approach, with a certain probability the algorithm will select the cluster with second-highest ClusterBid

[1] The initial version of IGTA was developed by John Idicula (Idicula, 1995) and then later refined by Gutierrez Fernandez (Fernandez, 1998) and Thebeau (Thebeau, 2001). Gutierrez Fernandez converted the program into C and integrated it with the Excel environment augmented it with more user-control parameters. The current MATLAB version of IGTA is written by Thebeau. He also conducted experiments to determine values for several strategic algorithm control parameters. Thebeau‘s Matlab code is freely downloadable (“DSMweb.org: MATLAB Macro for Clustering DSMs,” n.d.).

rather than highest bid. Only if the new clustering solution has a lower objective variable value compared with an updated "best solution" will the solution be accepted as a new best solution. The IGTA+ objective function can be expanded in the form

$$TotalCost = ExtraClusterCost + \ IntraClusterCost \tag{1}$$

where *ExtraClusterCost* is the total number of interactions between different modules, weighted by the size of the DSM, and *IntraClusterCost* is the number of interactions within individual modules weighted by the sizes of the module. These quantities are as defined in (Fernandez, 1998) as follows:

$$IntraClusterCost = \\ \sum_{i=1}^{n_{clusters}} \left(\sum_{j,k\ \in Cluster_i} DSM_{i,k} + DSM_{k,i} \right). ClusterSize_i^{powcc} \tag{2}$$

$$ExtraClusterCost = \\ \sum_{i=1}^{n_{clusters}} \left(\sum_{j,k\ \notin Cluster_i} DSM_{i,k} + DSM_{k,i} \right). DSMSize^{powcc} \tag{3}$$

Here we assume each component can only be assigned to one cluster, and *ExtraClusterCost* and $IntraClusterCost$ can take form expressible in matrix form and so be quickly computed (Borjesson & Hölttä-Otto, 2012). Solutions generated by the IGTA+ algorithm are mainly in the neighborhood of the utopia point, and do not distribute very well through Pareto-frontier. Therefore, we modify the definition of total cost into

$$TotalCost = \min(p\ IntraClusterCost + (1-p)ExtraClusterCost); \quad (0 < p < 1) \tag{4}$$

Where p is the weighting parameter, we increase p from 0 to close to 1 incrementally in the computations (computation time increases substantially for values of$p$ close to 1). So the points will be pushed into two extremes of Pareto-frontier (corresponding to p = 0 and p = 1) rather than concentrating in the middle of Pareto-curve (corresponding to p=0.5) and this results into more even distribution of points over the curve. In addition, other parameters of the IGTA+ algorithm can also tuned to obtain a clearer picture of a larger portion of Pareto curve.

We tuned the IGTA+ parameters before running examples. Although Thebeau (Thebeau, 2001) suggested values of *pow_bid* = 1 and *pow_cc* =1 as they worked reasonably well in their Utopia point search cases, for our Pareto search case these values lead to small modules where all architectures had more than 4 modules. Through adjustment of these values we rewarded larger clusters. Here through experimentation search, we found that *pow_bid* = -2 and *pow_cc* = -1 produce better sized modules with broader range of number of modules for architectures across the Pareto frontier.

3 Results and Discussion

We now demonstrate the approach on a Black & Decker Dustbuster shown in Figure 1, a vacuum cleaner with 57 components and 89 interactions, originally developed by Borjesson (Borjesson & Hölttä-Otto, 2012). We implemented our methodology on different examples in different ranges of complexity, such as Jet engines, Power-screwers, motor controllers, MRI contrast injectors and printers and similar trends have been overserved in all these examples. These examples are not shown for sake of brevity. Design structure and partial disassembly of Black & Decker Dustbuster can be found in (Borjesson & Hölttä-Otto, 2012).

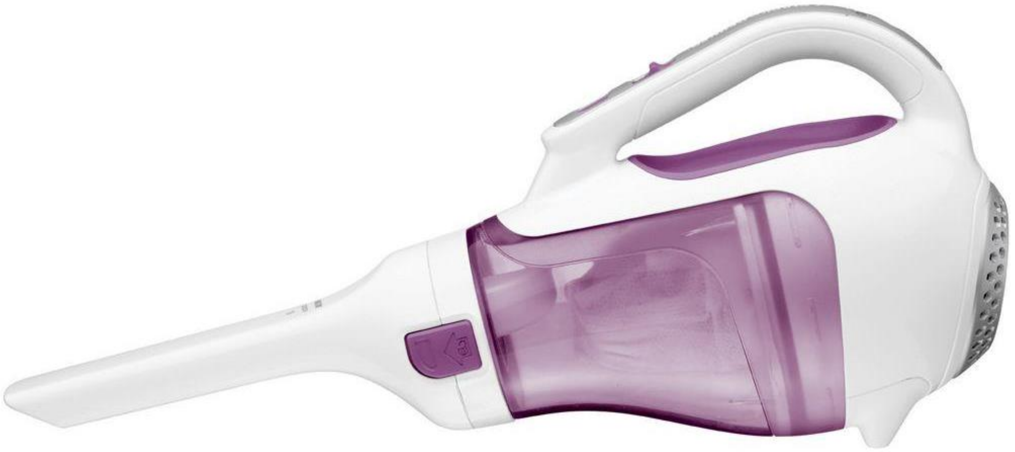

Figure 1. Black & Decker CHV1210 Dustbuster 12-Volt Cordless Cyclonic Hand Vaccum Cleaner (www.homedepot.com)

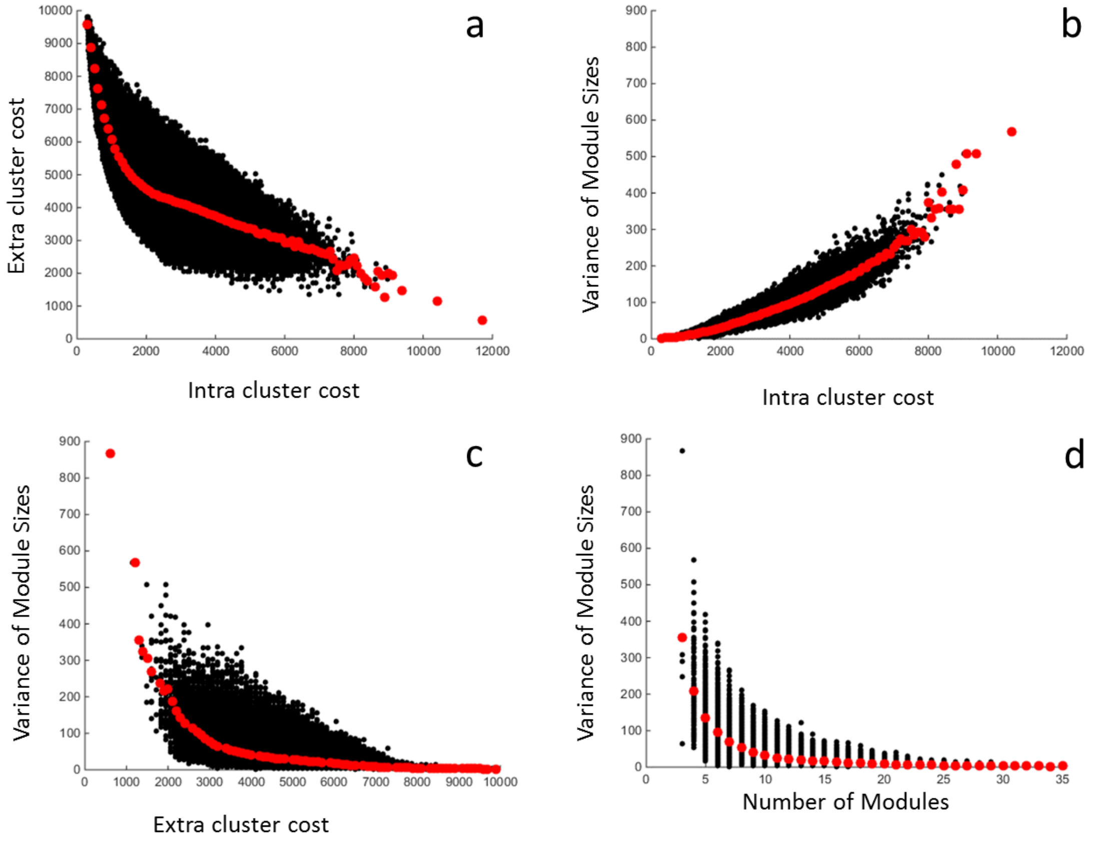

Figure 2. The correlation between different system architecture modularity parameters.

Correlation between different recorded parameters can be observed in Figure 1. The red-colored dots denote the average value of the parameter represented by the horizontal axis for a certain value of vertical axis variable.

Figure 2.a shows the trade-off between intra-cluster cost and extra-cluster cost. The conflict between intra-cluster and extra-cluster costs persists when the number of modules is kept fixed. Notice the trending relationship between the variance of the modules sizes and the other modularity factors. The trends drive the architectural modularity sizes to be larger.

As shown in Figure 2.b, the maximum variance is reduced by increasing the number of modules. For larger number of modules, there is a trend toward overall smaller number of components within each module. Therefore there is a decrease in variance as well.

As illustrated in Figure 2.c, both the minimum and maximum values of variance increase with intra-cluster cost. Intra-cluster cost has positive correlation, and extra-cluster cost has negative correlation with number of modules (Sanaei, Otto, Hölttä-Otto, & Luo, 2015). Thus, to see if this increase is due to increasing the number of modules or is an independent effect, we keep the number of modules fixed and observed the difference in the trend. We observed that both maximum and minimum values of variance of module sizes are positively correlated with intra-cluster cost. Intra-cluster cost is a driver of modular size variance within an architecture.

Finally, we consider observations on the relationship of extra-cluster cost for different number of modules. We observe negative correlation between extra-cluster and maximum variance of modules sizes. Again, to see if this negative correlation is attributable to the change in number of modules, we keep the number of modules fixed and observe the difference. We see that unlike the case of intra-cluster cost, both maximum and minimum values remain unchanged for a fixed number of modules with change in the variance.

In an earlier paper (Sanaei et al., 2015), we observed that the optimal modular architectures parameterized with an increasing number of modules does not constitute a hierarchical structure. Therefore, hierarchical clustering approaches such as cluster trees that assume a hierarchy behind optimal architecture do not necessarily provide the optimal architectures in terms of connectivity. Simply speaking, breaking single modules up and combining them into a new module could be less expensive that combining two modules already formed in one level of a nested hierarchy.

Here we observed the same result. There is at least one hierarchical architecture with minimum extra-cluster cost, but this particular architecture(s) might not have minimum intra-cluster cost as well. So if we are only concerned about extra-cluster cost, hierarchical clustering cost might be good enough. However, given that minimum intra-cluster cost corresponds to minimal variance of modules size, we can conclude that modules produced hierarchically might not come with proper sizing. Making the modules more or less of equal size has a compromising effect on extra-cluster cost. Therefore a better approach is to consider the trade-off between costs, module sizes and the number of modules, and interactively compare architectures on the Pareto frontier.

4 Limitations

Here we observed a correlation between variance in size of modules and intra-cluster cost, there are other factor to consider that may help understanding this relationship better. First, we observe a correlation between variance of modules and number of modules, to exclude this dependency from the relationship, it might be helpful to use standard deviation in place of variance and see the difference. Furthermore, the effect of *Powcc* on this relationship is another significant factor to consider.

5 Conclusions

Here we show that variance of size of modules can be controlled through intra-cluster cost, even when the number of modules is fixed. Furthermore, we observe that, in all examples we tried, optimal architecture in terms of extra-cluster cost constitute a hierarchy and therefore such optimal modularization can be obtained by agglomerative or divisive hierarchical clustering but optimal architecture in terms of both extra-cluster and intra-cluster cost do not constitute such a hierarchy and therefore hierarchical clustering algorithms are not able to find optimal clustering when both of these factors are of concern.

Acknowledgement

This work was supported by SUTD-MIT International Design Centre (IDC) (https://idc.sutd.edu.sg/). Any opinions, findings, or recommendations are those of the authors and do not necessarily reflect the views of the sponsors.

References

Baldwin, C. Y., & Clark, K. B. (2000). *Design rules: The power of modularity* (Vol. 1). MIT press.

Borjesson, F., & Hölttä-Otto, K. (2012). Improved clustering algorithm for design structure matrix. In *ASME 2012 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference* (pp. 921–930). American Society of Mechanical Engineers.

Dahmus, J. B., & Otto, K. N. (2001). Incorporating lifecycle costs into product architecture decisions. In *Proc 2001 ASME Design Engineering Technical Conferences*.

DSMweb.org: MATLAB Macro for Clustering DSMs. (n.d.). Retrieved January 10, 2015, from http://www.dsmweb.org/en/dsm-tools/research-tools/matlab.html

Du, X., Jiao, J., & Tseng, M. M. (2001). Architecture of product family: fundamentals and methodology. *Concurrent Engineering*, *9*(4), 309–325.

Fernandez, C. I. G. (1998). Integration analysis of product architecture to support effective team co-location. *ME Thesis, MIT, Cambridge, MA.*

Gershenson, J. K., Prasad, G. J., & Zhang, Y. (2003). Product modularity: definitions and benefits. *Journal of Engineering Design*, *14*(3), 295–313.

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.

Hölttä, K. M., & Otto, K. N. (2005). Incorporating design effort complexity measures in product architectural design and assessment. *Design Studies*, *26*(5), 463–485.

Hölttä-Otto, K., Chiriac, N. A., Lysy, D., & Suk Suh, E. (2012). Comparative analysis of coupling modularity metrics. *Journal of Engineering Design*, *23*(10–11), 790–806.

Idicula, J. (1995). Planning for concurrent engineering. *Researchreport, Gintic Institute, Singapore*.

Marshall, R., & Leaney, P. G. (1999). A systems engineering approach to product modularity. *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture*, *213*(8), 847–851.

Otto, K. N., & Wood, K. L. (2001). *Product design*. Prentice hall.

Sanaei, R., Otto, K., Hölttä-Otto, K., & Luo, J. (2015). Trade-Off Analysis of System Architecture Modularity Using Design Structure Matrix. In *ASME 2015 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference* (p. V02BT03A037-V02BT03A037). American Society of Mechanical Engineers.

Sanchez, R., & Mahoney, J. T. (1996). Modularity, flexibility, and knowledge management in product and organization design. *Strategic Management Journal*, *17*(S2), 63–76.

Stone, R. B., & Wood, K. L. (2000). Development of a functional basis for design. *Journal of Mechanical Design*, *122*(4), 359–370.

Stone, R. B., Wood, K. L., & Crawford, R. H. (1998). A heuristic method to identify modules from a functional description of a product. In *Proceedings of DETC98* (pp. 1–11).

Stone, R. B., Wood, K. L., & Crawford, R. H. (2000a). A heuristic method for identifying modules for product architectures. *Design Studies*, *21*(1), 5–31.

Stone, R. B., Wood, K. L., & Crawford, R. H. (2000b). Using quantitative functional models to develop product architectures. *Design Studies*, *21*(3), 239–260.

Sudjianto, A., & Otto, K. (2001). Modularization to support multiple brand platforms. In *Proc. ASME International Design Engineering Technical Conferences, Design Theory and Methodology*.

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

Zamirowski, E. J., & Otto, K. N. (1999). Identifying product family architecture modularity using function and variety heuristics. In *11th International Conference on Design Theory and Methodology, ASME, Las Vegas*.

http://www.homedepot.com/p/BLACK-DECKER-12-Volt-Dustbuster-Hand-Vac
CHV1210/203157739