

KEYWORDS ■ Product development ■ Product architecture ■ Modularity
■ Resource allocation ■ Design rules

INVESTMENT DECISIONS IN MODULAR PRODUCT DEVELOPMENT

✓ **Ali A. Yassine**

Engineering Management Program, Faculty of Engineering & Architecture, American University of Beirut, Beirut, Lebanon
ali.yassine@aub.edu.lb

ABSTRACT

Developing new products is critical to firms' success and is the key for sustaining competitive advantage. Firms engaging in product development (PD) face the important problem of allocating scarce development resources to a multitude of opportunities. In this paper, we propose a mathematical formulation to optimize product development (PD) investment decisions. The model maximizes the performance of a product under development based on its product architecture and the firm's resource constraints. The analysis of the model shows that the architecture of a product plays an essential role in affecting the optimal resource allocation to various product modules.

1. Introduction

Many believe that in developing complex products, we can individually design or improve each module's performance separately, but this may affect the behavior or performance of other dependent modules. This is due to a known or unknown common function, feature, or interaction in the product which is implemented by more than one module or component. As opposed to perfect modularity, integral systems involve a strong dependency between individual modules where changes made to any component (*to improve its performance*) may deteriorate or

improve the performance of others. Consequently, in an integral architecture an optimal performance for each individual module or component may not necessarily lead to a global optimal performance for the whole product or system due to the complex interactions between the various modules (Mihm et al., 2003).

Although modular systems have many time and cost benefits (*e.g., parallel module development which saves development time and module commonality across multiple product lines which saves development cost*), a perfectly modular design may not always be achievable due to business and technical constraints (Ulrich, 1995; Holtta et al., 2005). Cutherell (1996) argues that integral architecture is often

driven by product performance or cost and modular architecture by variety, product change, engineering standards, and service requirements. Along similar lines, Whitney (2004) argues that modularity is not always a desirable property; a modular product is likely to be larger, heavier, and less energy efficient.

This paper aims to develop a model of resource allocation in PD to maximize product performance (*by investing in modules and in design rules*) where the topology of the product architecture will be taken into account. Our intention is to build a foundation for a formal theory that can link PD product development investment decisions to product architecture and to ultimately explain the

integral-modular dynamics in product architecture as envisioned in **Figure 1**. The figure shows that PD investments can be placed on individual modules and/or on developing design rules that govern the relationships between these modules. Two hypotheses can be derived from this figure. First, the product architecture affects resource allocation decisions and ultimately product performance. Second, for the same product architecture, there is a shift in the temporal allocation of resources from design rules to individual modules; thus, supporting the move from integral to modular architectures as the product evolves. In this paper, we are set to address the first hypothesis.

Substantial empirical work supports the notion of a product architecture evolution from integral to modular product architectures (Fixson and Park, 2008; Schilling, 2000). The double helix model (Fine, 1998) illustrates how product (*or industry*) structure evolves from integral (*or vertical*) to modular (*horizontal*). Fine (1998) explains that starting with an industry exhibiting a vertical structure and integrated product architecture, a number of forces push toward the disintegration of the product architecture into a more modular one. On the other hand, with a modular architecture, numerous other forces push toward the integration of product architecture and industry structure. Similarly, Christensen et al. (2002) explain how integral and modular product architectures change over time when the performance of sustaining and disruptive technologies is considered.

Ethiraj (2007) argues and empirically demonstrates that the interactions between components in a product system condition the R&D investment incentives of firms. Along similar lines, Ethiraj and Posen (2013) iterated the same message that R&D effort and performance returns that accrue to that effort depend on the

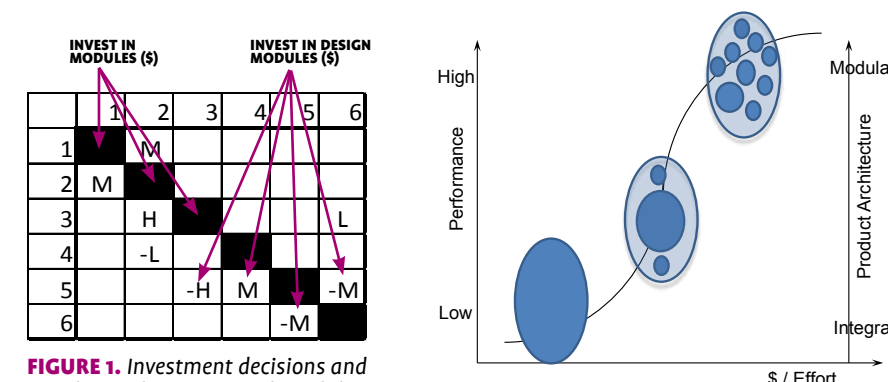


FIGURE 1. Investment decisions and its relationship to integral-modular dynamics

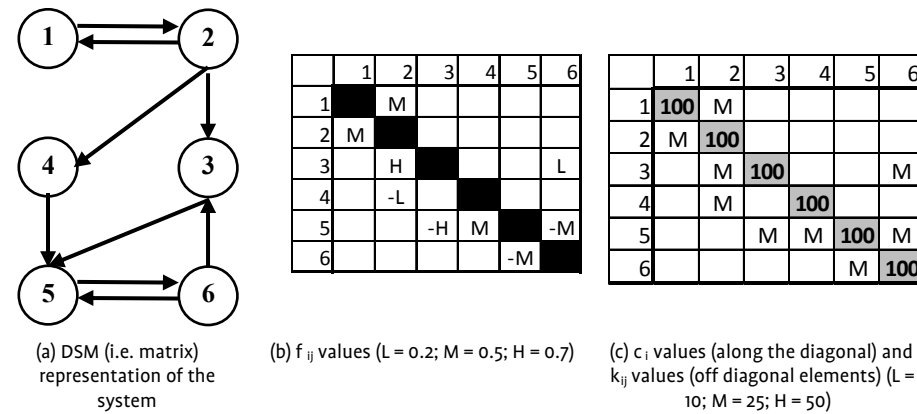
dependencies between components of a complex product and the changes in these dependencies over time.

At the surface, this paper proposes a useful resource allocation model for PD projects. However, the importance of the model stems from its capability to serve as a basis for developing analytical models that capture many of the qualitative observations in the literature such as comparing investment profiles for modular and integrated architectural configurations; in addition to a comparison between investment profiles based on module connectivity. More importantly, the model can shed light on several insights regarding the behavior of modular and integrated systems. For instance, we may be able to explain why and under what conditions integrated systems can attain higher performance levels during development compared to modular systems.

2. The General Model

In this section we present a mathematical programming formulation for the optimal investment decisions in PD. The overall product (*or system*) performance can be described as the total performance of the individual modules. Performance is defined as a measure of the product's fidelity with respect to its requirements (Joglekar et al., 2001). (*Two examples of fidelity can be the speed of a microprocessor and the number of bugs eliminated from a new software release.*) Furthermore, the performance of each individual module is dependent on the amount of resources invested in that particular module. However, the interdependencies among the different modules require that any investment made in one particular module to be accompanied by additional investments in dependent modules in order to ensure their compatibility with each other. We therefore consider that the interdependency can be reduced by investing in the design rules (*defining the connections or relationships between these modules*); thus, reducing the level of interdependency among the modules. Hence, the problem is stated as finding the optimal investment amounts that should be made in each of the individual modules and in the design rules in order to maximize the overall product performance. The following decision variables are then used to model the problem: α_i is the amount invested in module i , and θ_{ij}

FIGURE 2.
Six module
example



is the amount invested in the design rules for the connection between module i and module j . The overall system performance P_T is then expressed as the sum of the performance of the M individual modules such that:

$$P_T = \sum_{i=1}^M P_i(\alpha) \quad (1)$$

where P_i is the performance of module i that is an increasing function of the total investment in module i . That is, the more budget spent working on a module, the higher the level of performance that can be achieved. Furthermore, since in an integrated system modules impact each other, we assume that investing in a module also affects (i.e. either improves or deteriorates) the performance of other dependent modules. However, the performance, P_i , cannot be negative because we can always choose to keep the current design of module i and only incur a compatibility cost as discussed next. Hence, the performance of each module is an increasing function of investments in the module itself and in other dependent modules; thus, performance of module i , $P_i(\alpha)$, is a function of α which is the investment vector $(\alpha_1, \dots, \alpha_M)$ in all modules.

In addition to the amount invested to improve the performance, additional investments are required to insure compatibility among the modules (Smith and Eppinger, 1997; Yassine et al., 2003). The investment that is due to compatibility, l_{ij} , is a function $L_{ij}(\alpha_i, \theta_{ij})$ that is increasing in α_i and decreasing in θ_{ij} . Thus, for two dependent modules (say module j depends on i), investing α_i in a module i necessitates an additional investment in module j that is proportional to α_i . Furthermore, the additional investment $L_{ij}(\alpha_i, \theta_{ij})$ can be reduced by investing θ_{ij} in the design rules between modules i and j . Finally, a maximum investment budget B is considered to be available throughout the development process. Therefore, the investment decision problem is:

$$\begin{aligned} \max P_T &= \sum_{i=1}^M P_i(\alpha), \quad (2) \\ \text{s.t.} \quad \sum_{i=1}^M \alpha_i &+ \sum_{i=1}^M \sum_{j=1; j>i}^M \theta_{ij} + \sum_{i=1}^M \sum_{j=1; j \neq i}^M l_{ij} \leq B, \quad (3) \\ l_{ij} &= L_{ij}(\alpha_i, \theta_{ij}), \quad (4) \\ \alpha_i, \theta_{ij}, l_{ij} &\geq 0 \quad \forall (i, j). \quad (5) \end{aligned}$$

The auxiliary variable l_{ij} denotes the necessary investment needed by module i to insure compatibility with module j after α_i is invested in module j . Functional forms for $P_i(\alpha)$ and $L_{ij}(\alpha_i, \theta_{ij})$ are discussed in the following section and a stylized model that is used for testing is presented next.

2.1 The Stylized Model

In this section, we consider functional forms for $P_i(\alpha)$ and $L_{ij}(\alpha_i, \theta_{ij})$ and present a stylized model based on (2)–(5). Particularly, the performance function $P_i(\alpha)$ of each module i follows a production function that is increasing in terms of the resource α_j (Joglekar et al., 2001). The slope of the production function is also a function of the resources α_j that are invested in other modules. $P_i(\alpha)$ is then of the following form:

$$P_i(\alpha) = \max \left\{ 0, \left(C_i + \sum_{j=1; j \neq i}^M C_j f_{ij} \alpha_j \right) \alpha_i \right\}. \quad (6)$$

The module performance is zero when we have zero investment in the module and C_i is a measure of complexity for module i . C_i is hence a proxy for the design complexity (or module size as in Baldwin and Clark (2000)) of each of the modules where a simple module (i.e. small C_i value) will have a small impact on the overall system performance while a complex module (i.e. large C_i value) has a higher impact. It is

assumed that each module has its specific C_i which is known in advance.

The factor f_{ij} represents the strength of dependency among modules i and j , where $-1 \leq f_{ij} \leq 1$. We assume a symmetrical reciprocal dependency structure; that is, $f_{ij} = f_{ji}$. We note that if $f_{ij} = 0$ then module i is independent from module j and hence investments in modules j ($j \neq i$) will have no impact on module i . In the cases where $f_{ij} > 0$, investments in modules j increase the slope of the performance function of module i thus increasing the performance of i . Finally, when $f_{ij} < 0$, investments in module j decrease the slope of the performance function of module i . Note that in order to capture the synergistic effect of integrality between modules i and j , concurrent investment must be made in the modules. Thus, the term $C_j f_{ij} \alpha_j$ is multiplied by α_i in Equation (6).

In order to ensure compatibility after upgrades among the interconnected modules, an investment in one module requires additional investments in the other dependent modules (that is, directly connected to it). Therefore, the development manager can elect to invest in the design rules in order to reduce or even eliminate the dependencies among the modules (Baldwin and Clark, 2000). Investing in the design rules reduces the amount that needs to be spent on dependent modules. For instance in practice, an investment may be made in the connections between the modules in order to make them more generic. The compatibility among the interconnected modules is modeled as follows. An investment α_i in module i requires an equivalent investment $L_{ij}(\alpha_i, \theta_{ij})$ to be made in each module j to insure j 's compatibility (with changes in i). A decreasing function $L_{ij}(\alpha_i, \theta_{ij})$ of the following form is considered:

$$L_{ij}(\alpha_i, \theta_{ij}) = \left(\frac{C_j}{C_i} \right) (|f_{ij}| e^{-(k_{ij})\theta_{ij}}) \alpha_i \quad \forall (i, j); \quad (7)$$

where θ_{ij} is the amount invested in the design rules for the connection between modules i and j , and $k_{ij} \geq 0$, reflects a design knowledge parameter that alters the sensitivity of the function $L_{ij}(\alpha_i, \theta_{ij})$ depending on the designer's prior knowledge about the connection between modules i and j . Note that the compatibility cost is incurred regardless of the sign of f_{ij} ; thus, the absolute value used in equation (7).

Following equation (7), an investment α_i in module i , requires an additional proportional investment $\left(\frac{C_j}{C_i} \right) (|f_{ij}| e^{-(k_{ij})\theta_{ij}})$ in module j to ensure compatibility of module j with the updated module i . We note the scaling factor $\frac{C_j}{C_i}$ is included to account for the fact that the investment in each module should be proportional to its complexity. Therefore, the total investment in module i includes the amount α_i for updating module i in addition to an amount $\left(\frac{C_i}{C_j} \right) (|f_{ij}| e^{-(k_{ij})\theta_{ij}})$ for every α_j invested in the other modules

(impacting module i) to ensure the compatibility of the system. The investment decision problem is then formulated as follows:

$$\begin{aligned} \max P_T &= \sum_{i=1}^M P_i(\alpha) = \sum_{i=1}^M \max \left\{ 0, \left(C_i + \sum_{j=1; j \neq i}^M C_j f_{ij} \alpha_j \right) \alpha_i \right\} \quad (8) \\ \text{s.t.} \quad \sum_{i=1}^M \alpha_i &+ \sum_{i=1}^M \sum_{j=1; j>i}^M \theta_{ij} + \sum_{i=1}^M \sum_{j=1; j \neq i}^M l_{ij} \leq B \quad (9) \\ l_{ij} &= \left(\frac{C_j}{C_i} \right) (|f_{ij}| e^{-(k_{ij})\theta_{ij}}) \alpha_i \quad \forall (i, j) \quad (10) \\ \alpha_i, \theta_{ij}, l_{ij} &\geq 0 \quad \forall (i, j) \quad (11) \end{aligned}$$

The objective function (8) maximizes the total weighted performance of all the modules. Note that the objective function is a convex function. However, since we are maximizing, then the problem is a non-convex optimization problem. Constraint (9) is a budget constraint which limits the total investments in the individual modules, the total investment in the design rules, in addition to the investment that is required to ensure compatibility to a maximum budget B . (Constraint (9) is neither convex nor concave.) Constraint (10) insures the compatibility of the system by forcing an investment module i following an investment in module j . It is worth noting that since the objective function is increasing in α and all the coefficients of α in the budget constraint are non-negative, it can be easily verified that the budget constraint is binding at optimality.

2.2 An Illustrative Example

In order to demonstrate the methodology presented in Section 2.1, we introduce and solve an illustrative example consisting of six interdependent modules. The architectural configuration of this system is represented by the DSM in Figure 2. Solving this example, we get an investment of 0.396 in module 4 and 0.396 in module 5. Investment in all other modules is zero.

3. Sensitivity Analysis & Discussion

We start the sensitivity analyses by considering a base example that has a purely modular architecture (as shown in example 1 in Figure 3a). Then, we progressively add more connections between the modules to systematically decrease modularity. A sample of potential examples that could result from this experimental setup is shown in Figure 3. In this experimental setup, we are trying to make the base example (example 1) more integral by simultaneously increasing coupling and connectivity between modules. This process results in examples 2 through 6, as shown in Figure 3. These exam-

ples can be used to carry statistical analysis on the relationship between investment decisions (*i.e., in various modules*) and product architecture. In the base example (**Figure 3a**), all the modules are identical so the budget is divided equally on all the modules (0.1667/module). Then, in example 2 (*in Figure 3b*), we set all f_{ij} to 0.5 and change f_{12} between -1 and 1. We get the results shown in **Figure 4**. The figure shows that, if all things are equal, investment will be allocated to depend-

ent modules (*i.e., modules 2, 4 and 6 in this example*). However, as one of the dependencies becomes high enough, the dependency benefit (*in terms of overall system performance*) justifies spending the compatibility cost and investment is made in the pair of dependent modules (*i.e., modules 1 and 2*) and ignoring the other modules. Similar analysis must be carried out in order to develop an understanding of two important questions: (*a*) do modules attract investment based on

their connectedness to the rest of the system and how? And (*b*) do certain architectural properties of the system impose specific investment profiles and how? In another experiment on the examples of **Figure 3**, we increased the budget between 1 and 25 and plotted the performance for examples 1, 2, 5, 6. Examples 1 and 2 have high modularity while 5 and 6 are low modularity. We observed that for low modularity the performance increased much faster

than for high modularity as depicted in **Figure 5**. We also compared the investment profiles for example 2 (*which is high modularity*) to example 4 (*which is low modularity*) and the results are shown in **Figure 6**. **Figure 6** shows the difference in investment profiles, for a modular versus more integrated systems, between direct investments in module development versus investment in design rules. The figure shows that higher amounts are invested in design rules in integrated systems compared to simple systems. This means that, in integrated systems, it is optimal to initially invest more in design rules, which is consistent with the literature (*e.g., Baldwin and Clark 2000*).

maximizes total product performance based on the product architecture. Such allocations can target either the modules themselves or the design rules that dictate the dependency strength amongst these modules. Managerial guidelines that inform product development management can result from the analysis of such a model. These guidelines shed some light on the evolution of product architecture within the integral- modular spectrum and assist in explaining some of the intrinsic properties of modular and integral systems. Most importantly, the model shows that development managers must understand the architecture of the product when making resource allocation decisions. Specifically, when we compare the performance of the simple and the integrated systems, we observe that integrated systems tend to have higher performance. Furthermore, in integrated systems we tend to have higher investments in the links (*i.e. design rules*). In future work, our model can benefit from real world design network structures to validate the resultant investment profiles. One approach is to use DSM networks that have been analyzed in the literature (*e.g., Eppinger and Browning, 2012*). Also, by using one of the ‘modularity’ indices in the literature, one can then compare the performance of various DSM structures that exhibit different levels of modularity.

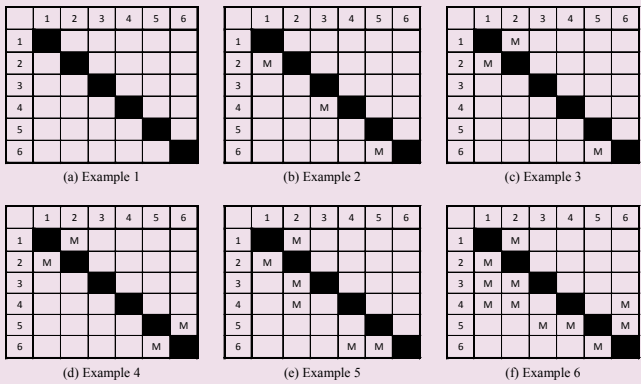


FIGURE 3. Progressively more integral architectures with increased coupling

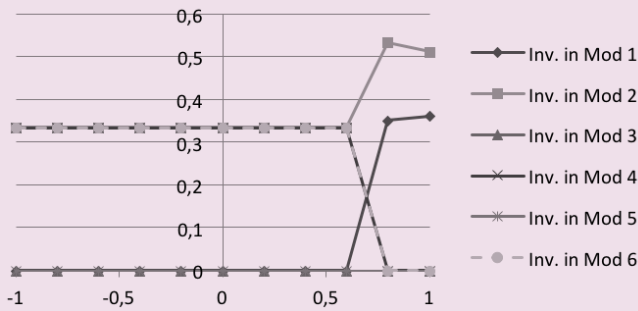


FIGURE 4. Investment profiles for example 2 as we vary f_{12} between -1 and 1

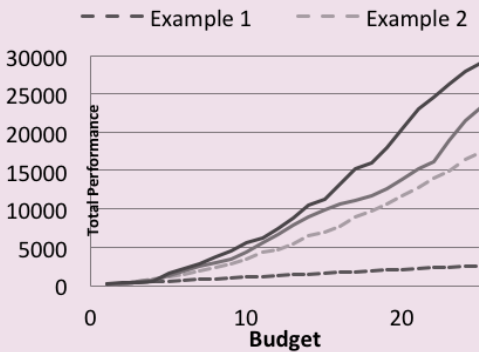


FIGURE 5. Performance vs. budget in simple and integrated systems

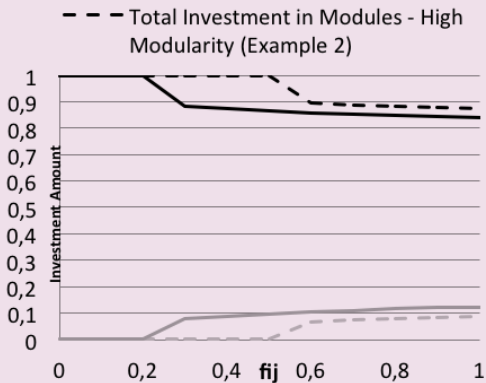


FIGURE 6. Investment profiles for modules and design rules

Baldwin, C., Clark, K. (2000). Design Rules: The Power of Modularity. Cambridge, MA, USA: MIT Press.

Cutherell, D. (1996). Product Architecture. The PDMA Handbook of New Product Development, Rosenau M., Griffin A., Castellion G., and Anschuetz N. (eds), John Wiley & sons, 1996.

Eppinger, S. D. and Browning, T. R. (2012). Design Structure Matrix Methods and Applications, MIT Press, Cambridge, MA.

Ethiraj, S.K. and Posen, H.E. (2013). Do product architectures affect innovation productivity in complex product ecosystems. *Advances in Strategic Management*, 30, 127-166.

Ethiraj, Sendil K. (2007). Allocation of Inventive Effort in Complex Product Systems. *Strategic Management Journal*, 28(6):563-584.

Fine, C.H. (1998). Clockspeed—Winning Industry Control in the Age of Temporary Advantage. Reading, MA: Perseus.

Fixson, S. K. and Park, J.-K. (2008). The Power of Integrality: Linkages between Product Architecture, Innovation, and Industry Structure. *Research Policy*, 37(8), 1296-1316

Hölttä, K., Suh, E., de Weck, O. (2005). Tradeoff between Modularity and Performance for Engineered Systems and Product? *International Conference on Engineering Design*, AUGUST 15 – 18, 2005.

Joglekar, N., Yassine, A., Eppinger, S., Whitney, D. (2001). “Performance of Coupled Product Development

Activities with a Deadline”, *Management Science*, 47(12), 2001.

Mihm, J., Lock, C., Huchzermeier, A. (2003). Problem-Solving Oscillations in Complex Engineering Projects. *Journal of Management Science*, 49 (6), 733-750.

Schilling, M.A. (2000). Towards a general modular systems theory and its application to interfirm product modularity. *Academy of Management Review* 25 (2), 312–334.

Smith, R. P., Eppinger, S.D. (1997). Identifying Controlling Features of Engineering Design Iteration, *Manage. Sci.*, vol. 43, pp. 276-293.

Ulrich, K., (1995). The Role of Product Architecture in the Manufacturing Firm. *Journal of Research Policy*, 24, 419-440.

Ulrich, K., Eppinger, S.D. (2008). Product Design and Development. New York, NY: McGraw-Hill/Irwin.

Wasserman, S., Faust, K. (1994). Social Network Analysis. Cambridge University Press, New York.

Whitney D. E. (2004). Physical Limits to Modularity. Working paper, ESD-WP-2003-01.03-ESD, Massachusetts Institute of Technology, Engineering Systems Division.

Yassine A., Joglekar N., Braha, D., Eppinger S., and Whitney D. (2003). “Information Hiding in Product Development: The Design Churn Effect.” *Research in Engineering Design*. 14 (3), 131-144.

author

Ali Yassine is a Professor and Chair of the Industrial Engineering & Management Department at the American University of Beirut. Previously, he was an Assistant Prof. at the Department of Industrial & Systems Engineering at the University of Illinois at Urbana-Champaign (UIUC). Prior to joining UIUC, Professor Yassine was a research scientist at MIT Center for technology, Policy and Industrial Development (CTPID) and a project manager at Chrysler International Corporation. Professor Ali Yassine is an expert in using Operations Research (OR) techniques for managing the development and

operations of complex engineering systems. Professor Yassine has authored over seventy articles, in refereed academic journals and international conferences, covering Product Design & Development, Project & Portfolio Management, Systems & Entrepreneurial Engineering, and Management of Technology & Technological Innovations. Dr. Yassine received the B.E. degree in Mechanical Engineering in 1988 from the American University of Beirut. He received the M.S. and Ph.D. degrees in 1989 and 1994 in Industrial and Manufacturing Engineering from Wayne State University in Detroit, Michigan. He is a member of INFORMS, ASME and PDMA.

