Proceedings of IDETC/CIE 2009
ASME 2009 International Design Engineering Technical Conference
& Computers and Information in Engineering Conference
August 30-September 2, San Diego, California, USA

DETC2009-87280

A STRATEGIC PLATFORM DESIGN METHOD FOR DEVELOPING CUSTOMIZED FAMILIES OF SERVICES

Seung Ki Moon
Department of Mechanical Engineering
Texas A&M University
College Station, TX 77840 USA

Jun Shu
Department of Supply Chain & Information system
The Pennsylvania State University
University Park, PA 16802 USA

Timothy W. Simpson[*] and **Soundar R. T. Kumara**
The Harold & Inge Marcus Department of Industrial & Manufacturing Engineering
The Pennsylvania State University
University Park, PA 16802 USA

ABSTRACT

Products are often paired with additional services to satisfy customers' needs, differentiate product offerings, and remain competitive in today's market. This research is motivated by the need to provide guidelines and methods to support the design of such services, addressing the lack of knowledge on customized service design as well as methods for designing and evaluating services for mass customization. We extend concepts from module-based product family design to create a method for designing families of services. In particular, we introduce a strategic platform design method for developing customized families of services using game theory to model situations involving dynamic market environments. A module-based service model is proposed to facilitate customized service design and represent the relationships between functions and processes that constitute a service offering. A module selection problem for platform design is considered as a strategic module sharing problem under collaboration, and we use a coalitional game to model module sharing and decide which modules provide more benefit when in the platform based on marginal contribution of each module. To demonstrate implementation of the proposed method, we use a case study involving a family of banking services.

1. INTRODUCTION

Products are often paired with additional services to satisfy customers' needs, differentiate product offerings, and remain competitive in today's market. Customized services are an important source of revenue for many companies, particularly those working within a mass customization environment where customer satisfaction is of paramount importance. Innovative companies that generate a variety of products and services for satisfying customers' specific needs are invoking and increasing research on mass-customized services, but the majority of their efforts are still focused on products and manufacturing operations [1]. This research is motivated by the need to provide a basis of service design guidelines and methods, primarily because of a lack of knowledge on customized service design as well as methods for designing and evaluating services for mass customization.

Service science research seeks to improve the productivity and the quality of service offerings by creating new innovations, facilitating business management, and applying practical applications [2]. Recently, theories and methodologies for mass-customized products are being applied to service development [3, 4], and the concept of product family design, in particular, provides good solutions to various customized service industries [3, 5, 6]. Product family design is a cost-effective way to achieve mass customization by allowing highly differentiated products to be developed from a common platform while targeting individual products to distinct market segments [7]. By sharing and reusing assets such as components, processes, information, and knowledge across a family of products and services, companies can efficiently develop a set of differentiated economic offerings by improving flexibility and responsiveness of their product and service development processes [8].

The value of services depends on market segmentation strategies that are identified by information derived from the relationship between customer needs and service providers [9]. In dynamic and uncertain market environments, however, we often have incomplete or uncertain information regarding

---

[*] Please address all correspondences to this author: tws8@psu.edu

market trends, customer's preferences, production costs, and a company's strategies for service development. To facilitate customized service design in dynamic market environments, we investigate strategic module sharing among services for designing a service platform using a game theoretic approach. Game theoretic approaches provide a rigorous framework for managing and evaluating different strategies to achieve players' goals using their complete or incomplete information and knowledge [10].

The objective in this research is to develop a strategic platform design method for developing customized families of services using game theory to model situations involving dynamic market environments. We extend concepts from module-based product families to create a method for service design. A module-based service model is proposed to facilitate customized service design and represent the relationships between functions and processes in a service. Object-oriented concepts and a function-process matrix are used to analyze service processes and identify the relationships between the service functions and the service processes that are offered as part of a service. In this paper, we utilize module-based platform design by introducing common modules, variant modules, and unique modules for service family design. A module selection problem for platform design is considered as a strategic module sharing problem under collaboration situation. We employ a coalitional game to model module sharing and decide strategic solutions for selecting modules in the service family being designed.

The remainder of this paper is organized as follows. Section 2 reviews related literature and background for product and service family design as well as game theory applications. Section 3 describes the proposed method to design customized families of services using a module-based service model and a coalitional game. Section 4 gives a case study using a family of banking services. Closing remarks and future work are presented in Section 5.

2. BACKGROUND AND LITERATURE REVIEW

2.1 Extending Product Family Design Methods to Service Family Design

A product family is a group of related products based on a product platform, facilitating mass customization by providing a variety of products for different market segments cost-effectively [11]. A product platform is the set of features, components or subsystems that remain constant from product to product, within a given product family. A successful product family depends on how well the trade-off between the economic benefits and performance losses incurred from having a shared platform are managed [12]. Based on theories and methodologies for mass customized product design, families of services and service platforms have been developed and applied to provide solutions in various customized service industries [3, 5]. For example, a bank can develop a number of checking account service families by combining common services and various options for offering different services. The common services are considered as a service platform for the service families. A typical approach to create a variety of services is to provide customers with various options and choices related to individual customer needs, which often warrant additional charges as they add value to the initial offering [13]. Meyer and DeTore [5] proposed a platform-based approach to develop new services using methods and processes for applying product family and product platform design and applied the approach to define a new service platform in an international insurance company. Jiao *et al.* [3] discussed how design theories and methodologies for products and manufacturing systems can be applied to the design of service delivery systems for mass customization. They considered a service delivery system as a product system instead of an operational system. Perters and Saidin [6] investigated key factors for the implementation of mass customization in a services context and used service modules to represent the levels of modularization of the scope of work and process in designing mass customization processes. Li [14] introduced some concepts and assumptions for service package and service product module level in service innovation and service product development. Moon *et al.* [4] proposed a method to identify a service platform along with variant and unique models for service family design using a service process model and a fuzzy c-mean clustering method based on functional similarities.

2.2 Game Theory Applications

A game is a description of a strategic interaction that includes constrains based on players' actions. Game theory provides reasonable solutions for various games and evaluates their properties [10, 15]. According to the constraints and situations of the games, game theoretic models can be partitioned into three categories: (1) cooperative and non-cooperative games, (2) strategic and extensive games, and (3) games with complete and incomplete information. In engineering design, game theoretic approaches have been applied to model strategic relationships between designers for sharing design knowledge and solving design problems. Xiao *et al.* [16] applied game theoretic approaches and design capability indices to model the relationships between engineering teams that were described as cooperative, non-cooperative, and leader/follower protocols, and facilitate collaborative decision-making during a product realization process. Fernandez *et al.* [17] proposed a framework for establishing and managing collaborative design spaces by combining elements of cooperative and non-cooperative behavior, and formulating strategic and extensive games with utility theory. Kopin and Wilbur [18] introduced a Bayesian game to model cost sharing in uncertain and incomplete information that were related to producer and consumer attributes such as nature, production costs, players and information, and preferences. Correia [19] investigated the representation of incomplete and asymmetric information to model a strategic Bayesian game that was represented by the constraints of a transmission system and player's strategic reactions to estimate uncertainties. Lewis and Mistree [20] presented mathematical constructs for modeling a multidisciplinary optimization problem using game theoretic principles and the compromised Decision Support Problem (DSP) in a collaborative, sequential, and isolated design environment. Lariviere and Van Mieghem [21] used a timing game to determine customers' arriving behaviors based on a delay cost function in a service facility and proposed appropriate strategies to select arrival times for the various volumes of service capacities. Finally, Huang *et al.* [22] described a multi-stage non-cooperative configuration game between platform products and supply chains to determine the optimal configuration decisions of a manufacturer and suppliers for mass customization, and demonstrated the applications of the proposed game and solution procedure using a series of simulation experiments and a numerical example. Module-based service platform design and the proposed coalitional game for service family design are discussed next.

3. MODULE-BASED SERVICE FAMILY DESIGN

To develop customized services, we use the following definitions for service family design [4]:

- A *service family* is a set of services based on a service platform, facilitating mass customization by promoting customer value and providing a variety of services for different market segments cost-effectively.
- A *service platform* is a common basis that consists of processes, activities, objects, and/or features that are shared and remain constant from service to service, within a given service family.
- A *service module* is a set of service components for performing a service.
- A *service component* is regarded as an activity to satisfy certain services, which are defined by a set of processes, operations, people, objects, and/or features.

These definitions provide a foundation for modeling customized families of services. Based on these definitions, we extend concepts from platform-based product family design to develop a module-based service family. A service platform consists of common service modules that are defined as service components representing functions and processes. Based on the service platform, we can create a variety of services and families of services for satisfying various market segments depending on service-related design factors such as location, facility design and layout for effective customer and work flow, procedures and job definitions for service providers, measures to ensure quality, extent of customer involvement, equipment selection, and adequate service capacity [23]. Figure 1 illustrates a checking account service family that consists of a variety of services for market segments based on customers' characteristics, like personal/ business size and credit. In this case, the preferred service can be considered a service platform.

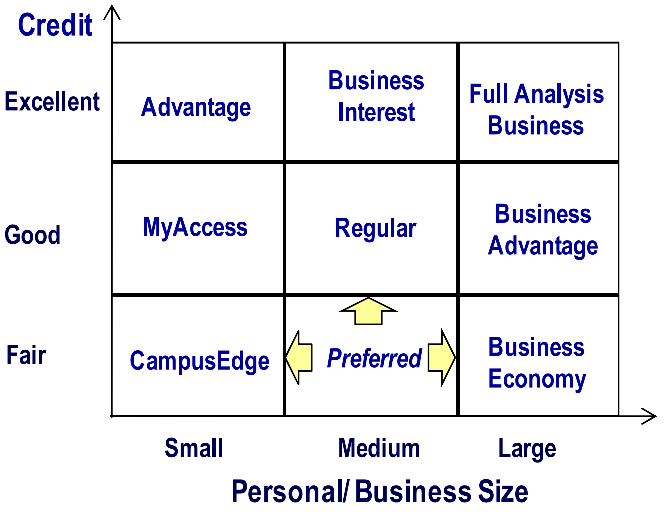

Figure 1: Example of a Family of Checking Account Services

Figure 2 shows the proposed method for developing customized families of services based on the bottom-up and module-based approaches in product family design. The proposed method consists of three phases: (1) service analysis and representation, (2) platform design strategy development, and (3) strategy determination. In the initial phase, service modules and components are identified as service functions and processes through service analysis. Then, platform design strategies are developed by module-based design concepts. After evaluating different platform design strategies using a game theoretic approach, a final platform is determined to provide a service family according design characteristics. A description of each phase follows.

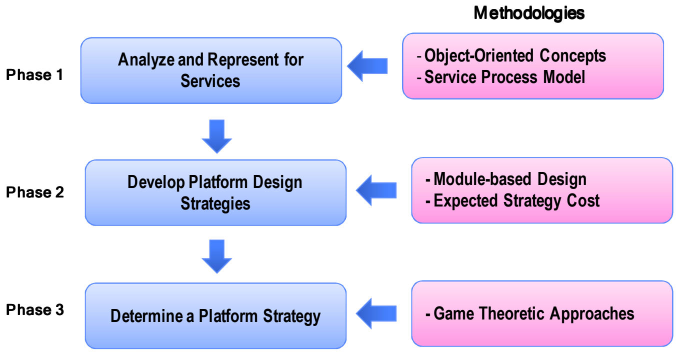

Figure 2: The Proposed Method for Determining a Service Platform

3.1 Phase 1: Service Analysis and Presentation

Object-oriented concepts can be used to support service analysis and representation by combining the module-based service model. Object-oriented design and analysis methodologies are used to develop information systems by modeling a system as a set of objects in the area of software engineering and business [24]. Through service analysis, we can determine service-related design factors that are represented as processes, activities, objects, and /or features, and service functions and processes. Based on service functions and processes, we can use a function-process matrix (FPM) to identify the relationships between functional modules and process modules in a service [4].

Based on the concepts of the product module-based design [25], we assume that a service can be decomposed into modules, which provide specific functions and processes, and processes are achieved by the combination of the modules' attributes. As shown in Figure 3, service modules are categorized into two different levels in a conceptual design phase: (1) a strategic level and (2) an operational level. The strategic level consists of service functional modules for developing service design strategies. The operational level is represented by service processes and provides a designer with cost information related to specific service functional modules and design strategies. To effectively define the relationships between functional hierarchies in a service, an appropriate representation scheme must be adopted for the services.

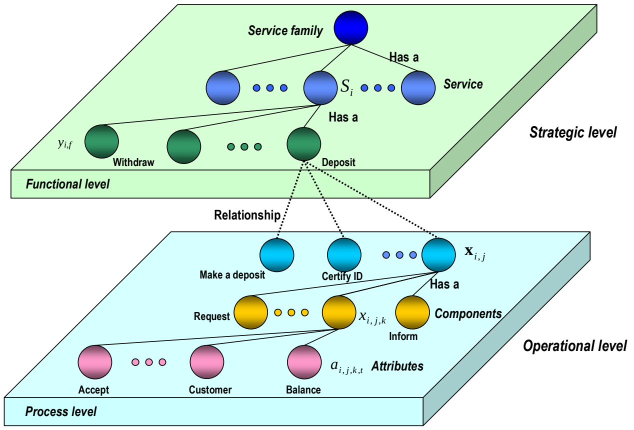

Figure 3: Service Strategic and Operational Levels and Hierarchy

Suppose that a service family consists of l services, $SF = (S_1, S_2, \ldots, S_l)$, and a service consists of f_i functional modules, $S_i = (y_{i,1}, y_{i,2}, \ldots y_{i,f}, \ldots, y_{i,f_i})$, where $y_{i,f}$ denotes service functional module f in service i. For service processes, suppose that a service consists of m_i service process modules, $S_i = (\mathbf{x}_{i,1}, \mathbf{x}_{i,2}, \ldots, \mathbf{x}_{i,j}, \ldots, \mathbf{x}_{i,m_i})$, where $\mathbf{x}_{i,j}$ is process module j in service i. and consists of a vector of length n_m, $\mathbf{x}_{i,j} = (x_{i,j,1}, x_{i,j,2}, \ldots, x_{i,j,k}, \ldots, x_{i,j,n_m})$, and the individual scalar components $x_{i,j,k}$ (k=1, 2,…, n_m) of a process module $\mathbf{x}_{i,j}$ are called *process features*. Each process feature consists of several attributes, $a_{i,j,k,t}$ (t=1, 2, …, t_n) $\in$ A_i, representing the component, $x_{i,j,k} = (a_{i,j,k,1}, a_{i,j,k,2}, \ldots, a_{i,j,k,t}, \ldots, a_{i,j,k,t_n})$, where t_n is the number of attributes defined by service analysis and A_i denotes the set of attributes in service i. Figure 3 shows the corresponding hierarchy for representing a family of checking account services. The identification of the attributes is problem-dependent, but an example is illustrated in the banking services case study in Section 4.

A functional service module consists of service process modules that are represented by service components to perform a service in a service family. Let C_f^F be functional module cost of functional module f, C_{jf}^P process module cost of process module j in function f, C_{jk}^C process component cost of process component k in process module j, C_{jkt}^A attribute cost of attribute t in the process component k of process module j, and C_{fj}^I interface cost of process module j in function f. Interface cost depends on the number of service process modules. The functional module cost for function f is proportional to the number of service process modules as follows:

$$C_f^F \propto \sum_{j \in f} (C_{jf}^P + C_{fj}^I) \tag{1}$$

Process module cost, C_{fj}^P, is depended on the number of process components and can be represented by:

$$C_{fj}^P \propto \sum_{k \in j} C_{jk}^C \tag{2}$$

Process component cost, C_{jk}^C, is proportional to the number of component's attributes and can be obtained by:

$$C_{jk}^C \propto \sum_{t \in k} C_{jkt}^A \tag{3}$$

By introducing coefficients, α, β, and λ, the functional module cost can be formulated at the operational level as follows:

$$C_f^F = \alpha \sum_{j \in f} ((\beta \sum_{k \in j} (\lambda \sum_{t \in k} C_{jkt}^A)) + C_{fj}^I) \tag{4}$$

where α, β, and λ, are mapping cost coefficients related to process modules, process components, and attributes, respectively ($0 < \alpha, \beta, \lambda < \infty$). To set the functional module cost, an industrial case study is necessary to determine the mapping cost coefficients by investigating the relationships between modules in various conditions [26].

3.2 Phase 2: Platform Design Strategy Development

A module-based service family strategy will allow efficient design and transaction of viable families of service offerings. Specifically, modules for functional services can be categorized based on function into: (1) unique, (2) common, and (3) variant modules. Unique modules are based on distinctive functions within a service family so that functions in the modules cannot be replaced by those in different modules to fulfill the same functions. Different functions within the service family can be designed as unique modules to create a variety of services. Common modules are based on common functions within a service family so that functions in the modules can be shared. Variant modules are considered as options and can be used to increase a variety of services the same functions in a given service family. A variant module can be designed as a common module for a platform based on service processes related to fulfill a service. If a variant module is selected for a platform, additional service processes or service components are required to meet functional requirements for all services across a service family design. As such, service transaction costs include the costs caused by increasing the number of services that should be served by the variant module.

A well-defined platform reduces service transaction costs by improving economies of scale and reducing the number of redundant processes that are used. Suppose that a service family consists of unique modules, common modules, and variant modules as illustrated in Figure 4. The platform level is defined as the number of functional modules in the platform and consists of the common modules and the variant modules. An appropriate platform level for a service family can be determined by minimizing the process module costs associated with the service functional modules. For instances, high levels of the platform (i.e., high commonality) decrease interface and component costs while increasing customers' preference loss. On the contrary, low platform levels (i.e., low commonality) decrease customers' preference loss while increasing interface and component costs.

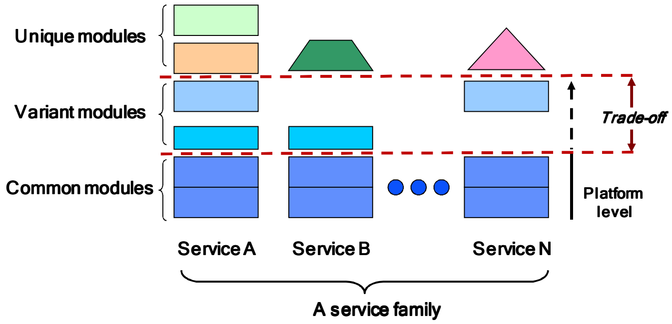

Figure 4: Trade-off in Platform Level Selection

Based on functional service modules, a platform designer determines a feasible set of strategies for the service platform based on his/her design knowledge. The strategies are represented as alternative designs for a service platform and can be constructed by combining process modules based on common functions in a service family. Let *G* be a set of strategies, *Y* a set of functions, *S* a set of services in a service family, and *Q* a set of module quantities. *G*, *Y*, *S*, and *Q* are finite sets.

The expected strategy cost, $c(g_i)$, for designer's strategy g_i ($i = 1,..,N$) is estimated by an expected cost function: $f^i : G \times Y \times S \times Q \mapsto \Re$. Hence, the real number of $f^i(g,y,s,q)$ represents the cost of function *y* at quantity *q* for performing strategy *g* for a platform in *s* services. For example, the expected strategy cost for *s* can be determined based on functional costs and the benefit of family design as [12]:

$$f^i(g,y,s,q) = \eta \times \frac{\sum_{i \in g_i} C_f^F}{l \times q} \tag{5}$$

where η is a factor for overhead, and l is a strategy weighting function as follows:

$$l = \begin{cases} 1, & \text{if module is unique} \\ \chi, & \text{otherwise} \end{cases} \tag{6}$$

where χ is the number of services including g_i in a service family, and q is a volume penalty factor related to product sales quantity. For a given set of services, $c(g_i)$ varies depending on choosing a strategy for platform design. The next section discusses a coalitional game model for determining a platform design strategy.

3.3 Phase 3: Platform Strategy Determination

Various game theoretic approaches can be applied to determine a design strategy according to information related to design environments as mentioned in Section 2.2. In this paper, we use a coalitional game that is designed to model situations wherein some of players have cooperation for seeking a goal in a game. A coalitional model focuses on the potential benefits of the groups of players rather than individual players [15]. In the coalitional model, the sets of payoff vectors are used to represent *value* or *worth* that each group of individuals can achieve through cooperation.

A module selection problem in platform design can be considered as a strategic module sharing problem under collaboration situation. We employ a coalitional game to model module sharing situations regarding uncertain market environments and solve the module selection problem in given service family design. To determine a platform, we decide which variant modules provide more benefit in the platform based on the marginal contribution of each module.

We assume that each module in a service can be modeled as a player. Then, consider the following module selection problem for platform design in a dynamic market environment. Each player has a strategy for designing a module. Each group of players (coalition) can be represented as a platform design strategy for a service family and be independent on the remaining players. To determine modules for platform design, we consider the set of all possible coalitions and evaluate the benefits of coalitions based on individuals' preferences.

In order to formulate the proposed scenario as a coalitional game, we must first identify the set of all players, *N*, and a function, *v*, that associate with every nonempty subset *G* of *N* (a coalition) [15]. A real number *v(G)* represents the *worth* of *G* and the total payoff that is available for division among the members of *G*. And, *v* satisfies the following two conditions: (1) $v(\varnothing) = 0$, and (2) (superadditivity) If $G, T \subset N$ and $G \cap T = \varnothing$, then $v(G \cup T) \geq v(G) + V(T)$.

Based on the definition of a coalitional game, the proposed game can be defined as:

- N: players who represent (variant) modules
- $v(G)$: the benefit of a coalition, $G \subset N$

where a coalition, *G*, represents a platform design strategy that consists of several variant modules as mentioned in Section 3.2. A utility is used to evaluate players' coalitional benefits for their pay offs and can be calculated by a utility function, $u_i(g)$, that is the difference between the cost of designing unique services, *N*, and the cost of designing family services based on a platform strategy, g_i :

$$u_i(g) = c(N) - c(g_i) \tag{7}$$

where $c(N)$ and $c(g_i)$ are estimated by the expected cost functions as mentioned in Section 3.2.

In this research, we use the Shapley value to analyze the benefits of family design and determine a design strategy for platform design[27]. The Shapley value is a solution concept for coalitional games and is interpreted as the expected marginal contribution of each player in the set of coalitions. Shapley value is defined as follows [27]:

$$\varphi_i(N,v) = \sum_{G \subseteq N, i \in G} \frac{(|G|-1)!(n-|G|)!}{n!}[v(G) - v(G \setminus \{i\})] \tag{8}$$

where $\varphi_i(N,v)$ is the payoff of player *i*, $|G|$ is the players' amount in the coalition *G*, $G \setminus \{i\}$ is the players' amounts deducing that of *i*, *n* is all players' numbers, and $v(G)$ is the payoff of the coalition *G*. Through the Shapley value, we can determine the payoff of each variant module.

In this game, strategies for players represent various platform design methods depending on variant modules in a service family. Therefore, service functions for platform design can be determined by selecting strategies based on service processes in a dynamic market environment. In the next section, the proposed game is applied to determine a variant module for platform design using a case study involving a family of banking services.

4. CASE STUDY

To demonstrate the implementation of the proposed method, a family of banking services consisting of four checking account services is used (see Table 1[1]). The checking account services are designed for four different market segments based on customer's preference, balance, credit, status, and so on. Using the proposed coalitional game, we will determine a platform design strategy for this family of banking services. This case study focuses on a module-based platform for the family of banking services at the conceptual stage of development and investigates which variant modules provide more benefit when in the platform based on marginal contribution of each variant module.

Table 1: Four Checking Account Services in a Banking Service Family

Option	Service A	Service B	Service C	Service D
Deposit	Yes	Yes	Yes	Yes
Withdraw	Yes	Yes	Yes	Yes
Transfer	Yes	Yes	Yes	Yes
Banking Statement	Yes	Yes	Yes	Yes
Online account statement	Yes	Yes	Yes	Yes
Checking writing	Yes	Yes	Yes	Yes
ATM transactions	Yes	Yes	Yes	Yes
Online banking with bill pay	Yes	Yes	Yes	Yes
Telephone banking	Yes	Yes	Yes	Yes
Trade stocks online	Yes	No	Yes	Yes
Optional business economic checking	Yes	No	Yes	No
Maintenance fee	Yes	No	Yes	Yes
Additional checking and saving account	No	No	Yes	No
Loans and lines of credit	No	No	Yes	No
Service for cashier' check, and so on	No	No	Yes	No
Interest	No	No	Yes	No
Preferred rates on Money Market, CDs	No	No	Yes	No

4.1 Phase 1: Service Analysis and Representation

Applying object-oriented concepts to service analysis, we can develop activity diagrams for identifying service processes or basic workflow. Activity diagrams provide a modeling method to represent the business and operational workflows using the detailed logic of a business rule [28]. By analyzing the activity diagrams for a service, we can define attributes and identify information flow among objects for service design. For example, an activity diagram for the deposit process was developed as shown in Figure 5. Processes in the diagram are represented by activities and attributes for performing the service. In this case study, a service process component is described as five attributes: (1) activity, (2) object, (3) input flow, (4) output flow, and (5) state. The activity is an event defining the intended interpretation of a service process between objects and is used as the name of a component. The object represents an object performing activities using input flow in certain services. The flow includes information, data, and materials, which occur in service processes. States are defined as things that change the input flow and the output flow. For example, a component changes the state of its inputs, such as information like a customer's account balance or credit, materials like money, and data in a banking service. Table 2 shows the process attributes of the selected service process modules and components based on the results of the service process analysis for the four checking accounts services.

---

[1] https://www.bankofamerica.com

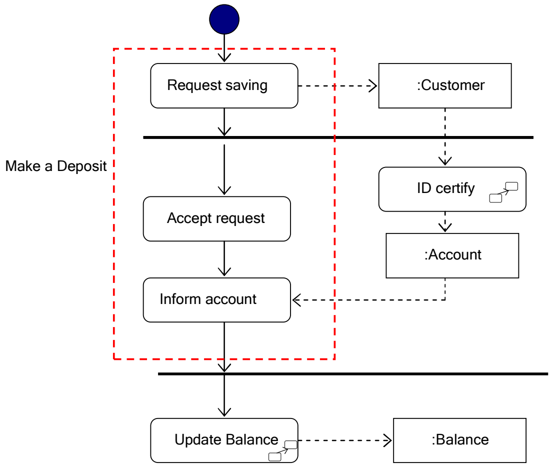

Figure 5: Example Activity Diagram for a Deposit Service

Table 2: Service Process Modules and Components for the Checking Account Services

Module Process	Component	Attributes				
		Activity	Object	Input flow	Output flow	State
Make a Deposit	$X_{1,1}$	Request	Customer	Account No.	Money	-
	$X_{1,2}$	Accept	Employee	Money	Amount	Balance
	$X_{1,3}$	Inform	Employee	Amount	Amount	-
Withdraw	$X_{2,1}$	Request	Customer	Account No.	Amount	-
	$X_{2,2}$	Accept	Employee	Amount	Money	Balance
	$X_{2,3}$	Inform	Employee	Amount	Amount	-
Transfer Money	$X_{3,1}$	Request	Customer	Account No.	Amount	-
	$X_{3,2}$	Query	Employee	Amount	Amount	-
	$X_{3,3}$	Confirm	Database	Amount	Amount	Balance
	$X_{3,4}$	Inform	Employee	Amount	Message	-
Trade Stocks	$X_{4,1}$	Request	Customer	Customer ID	Message	-
	$X_{4,2}$	Query	Employee	Message	Message	-
	$X_{4,3}$	Inform	Trading	Message	Message	-
Check writing	$X_{5,1}$	Request	Customer	Account No.	Amount	-
	$X_{5,2}$	Confirm	Employee	Amount	Amount	Balance
	$X_{5,3}$	Inform	Employee	Amount	Amount	-
Certify ID	$X_{6,1}$	Query	Employee	Customer ID	Customer ID	-
	$X_{6,2}$	Accept	Database	Customer ID	Account No.	-
	$X_{6,3}$	Reject	Database	Customer ID	Message	-
Check Credit	$X_{7,1}$	Query	Employee	Customer ID	Customer ID	-
	$X_{7,2}$	Inform	Database	Customer ID	Credit	-
Check Balance	$X_{8,1}$	Query	Employee	Account No.	Account No.	-
	$X_{8,2}$	Inform	Database	Account No.	Balance	-
Make a Loan	$X_{9,1}$	Proposal	Customer	Customer ID	Amount	-
	$X_{9,2}$	Accept	Employee	Amount	Message	Balance
	$X_{9,3}$	Reject	Employee	Amount	Message	-
Record Transaction	$X_{10,1}$	Confirm	Database	Amount	Amount	Balance
	$X_{10,2}$	Inform	Database	Amount	Balance	-
Open an Account	$X_{11,1}$	Request	Customer	Customer ID	Customer ID	-
	$X_{11,2}$	Accept	Employee	Customer ID	Account No.	-
	$X_{11,3}$	Inform	Employee	Account No.	Message	-

From the results of the service analysis, a function-process matrix (FPM) was developed to identify relationships between the service functions and processes in this set of four services as shown in Table 3. For example, a deposit service function consists of three service process modules: Make a Deposit, Certify ID, and Record Transaction. The Make a Deposit process module is composed of three components: request, accept, and inform (refer to Table 2). Each service process component has five attributes as defined in the service analysis. Based on functional modules and service plans, we categorized the modules into three modules: (1) common, (2) variant, and (3) unique modules.

Table 3: The Function-Process Matrix for Four Checking Account Services

Process module / Functional module		P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	P11	Module
		Make a Deposit	Withdraw	Transfer Money	Trade Stocks	Check writing	Certify ID	Check Credit	Check Balance	Make a Loan	Recode Transaction	Open an Account	
F1	Deposit	1					1				1		Common
F2	Withdraw		1				1		1		1		
F3	Transfer			1			1		1		1		
F4	Banking statement						1				1		
F5	Online account statement						1				1		
F6	Check writing		1			1	1		1		1		
F7	ATM transactions	1	1	1			1		1		1		
F8	Online banking with bill pay			1			1		1		1		
F9	Telephone banking			1			1		1		1		
F10	Trade stocks online				1		1		1		1		Variant
F11	Optional Business Economy Checking		1			1	1	1	1		1		
F12	Maintenance fee						1		1		1		
F13	Additional checking and saving account						1	1	1	1	1	1	Unique
F14	Loans and lines of credit						1	1	1	1	1		
F15	Service for cashier' check, and so on		1			1	1	1	1		1		
F16	Interest						1	1	1	1	1		
F17	Preferred rates on Money Market, CDs						1	1	1	1	1		

4.2 Phase 2: Platform Design Strategy Development

From the information in the FPM in Table 3, we can define seven design strategies based on the combination of three variant functional modules as shown in Table 4. In this paper, we assumed that the cost of a service component's attributes and the interface cost of a process module are 1 unit, respectively. Suppose $\alpha = \beta = \gamma = 1$, and then, the functional cost for strategy can be calculated by Equation (5). We assumed that a factor of overhead is 2 units. For example, strategy G3 has a component cost of 29 (=12+8+9) and an interface cost of 3 units according to its process modules. Therefore, since G3 supports three service plans, the expected design cost for G3 is 21.33, if the value of a volume penalty factor is 1. Table 4 shows seven expected design costs based on process modules when the value of a volume penalty factor is 1.

Table 4: Expected Strategy Cost for Variant Modules

Design	Functional module	Service plans	Process module	Expected design cost
G1	F10	A, C, D	P4, P6, P8, P10	30
G2	F11	A, C	P2, P5, P6, P7, P8, P10	69
G3	F12	A, C, D	P6, P8, P10	21.33
G4	F10, F11	A, C, D	P2, P4, P5, P6, P7, P8, P10	54.7
G5	F10, F12	A, C, D	P4, P6, P8, P10	30
G6	F11, F12	A, C, D	P2, P5, P6, P7, P8, P10	46
G7	F10, F11, F12	A, C, D	P2, P4, P5, P6, P7, P8, P10	21.33

4.3 Phase 3: Platform Strategy Determination

The game between three variant modules for platform design of this service family is defined as the proposed coalitional game that is described in Section 3.3. Table 5 summarizes the coalitional game for determining modules with three players. To determine marginal contributions for each variant module, the coalitional benefit of the design strategies were calculated by Equation (7) as shown in Table 6. The expected unique cost can be calculated by the sum of the expected design costs for individual functional modules in Table 5. Since there is no benefit in a unique design strategy, we defined four collaborations as the combination of three variant modules for design strategies. Therefore, the payoff vector of the game is $\mathbf{v}$(0, 0, 0, 0, 44.3, 21.33, 44.33, 95.66).

Table 5: Proposed Coalitional Game for Service Family Design

Game	Modules in Checking Account Services
Players (*N*)	F10, F11, F12
Coalition (*G*)	G0(ϕ), G1(F10), G2(F11), G3(F12), G4(F10, F11), G5(F10, F12), G6(F11, F12), G7(F10, F11, F12)

Table 6: Benefit of Coalitional Design Strategies

Coalition	Design strategy		Expected strategy cost	Expected unique cost	Benefit
	Common module	Unique module			
G4	F10, F11	F12	76.03	120.33	44.3
G5	F10, F12	F11	99		21.33
G6	F11, F12	F10	76		44.33
G7	F10, F11, F12	-	21.33		95.66

To determine the marginal contribution of each variant module, we used the Shapley value as mentioned in Section 3.3. The Shapley values of the variant modules (F10, F11, F12) are (28.05, 39.55, 28.06). Since F11 provides the largest contribution among the variant modules, F11 can be designed as a common module into a platform. Therefore, the Optional Business Economy Checking option will be included in a new platform to increase common options based on these service processes.

Through the case study, we demonstrated that the proposed coalitional game can be used to determine functional modules by selecting modules which provide more benefit with respect to service processes in service family design. We believe that the coalitional game can facilitate service family design in dynamic market environments.

5. CLOSING REMARKS AND FUTURE WORK

In this paper, we have established a foundation for service family design by providing a new and visual representation of modeling and designing services using modular design concepts and object-oriented concepts. We extended concepts from platform-based *product* family design to develop a method for module-based service family design. A module-based service cost model was introduced to support service design strategies in a service family. A function-process matrix (FPM) was used to identify the relationships between the service functions and the service processes in a family of services. A module selection problem was considered as a strategic module sharing problem under collaboration situation. We employed a coalitional game to model module sharing and decide which modules provide more benefit when in the platform based on marginal contribution of each variant module. We have applied the proposed coalitional game to determine service platform design strategies using a case study involving a family of banking services. Through the case, we demonstrated that the coalitional game can be used to determine the functional service modules that were described as a design strategy for platform design.

This research provides a method for developing a module-based platform for a family of services using functional and process modules in mass customization. To improve the proposed method, we need to develop a method that can identify modules based on the designers' knowledge and customers' requirements for establishing design strategies effectively. Since the proposed service cost function is sensitive to parameters in the mathematical model, the parameters should be determined based on services' characteristics, company's and customers' preferences, and a market environment. For a large-scale service family, an effective search algorithm is needed to generate a set of feasible strategies in a game. Future research efforts will be focused on improving the efficiency of the coalitional game, developing design strategies including unique modules for various service family environments, and expanding its application to develop a negotiation mechanism for web-based service family design. Also, the proposed method will be compared to other decision-making methods for determining a design strategy in a service family design.

REFERENCES

[1] Silveria, G. D., Borenstein, D., and Fogliatto, F. S., "Mass Customization: Literature review and research directions," *International Journal of Production Economics*, vol. 72, no. 1, pp. 1-13, 2001.

[2] Hidaka, K., "Trends in Services Sciences in Japan and Abroad," *Quarterly Review*, vol. 19, no. 2, pp. 35-47, 2006.

[3] Jiao, J., Ma, Q., and Tseng, M. M., "Towards high value-added products and services: mass customization and beyond," *Technovation*, vol. 23, no. 10, pp. 809-831, 2003.

[4] Moon, S. K., Simpson, T. W., Shu, J., and Kumara, S. R. T., "A Method for Platform Identification to Support Service Family Design," *International Journal of Services Operations and Informatics*, vol. 3, nos. 3/4, pp. 294-317, 2008.

[5] Meyer, M. H. and Detore, A., "Perspective: Creating a platform-based approach for developing new services," *The Journal of Product Innovation Management*, vol. 18, no. 3, pp. 188-204, 2001.

[6] Peters, L. and Saidin, H., "IT and the mass customization of services: the challenge of implementation," *International Journal of Information Management*, vol. 20, no. 2, pp. 103-119, 2000.

[7] Shooter, S. B., Simpson, T. W., Kumara, S. R. T., Stone, R. B., and Terpenny, J. P., "Toward an Information Management Infrastructure for Product Family Planning and Platform Customization," *International Journal of Mass Customization*, vol. 1, no. 1, pp. 134-155, 2005.

[8] Simpson, T. W., "Product Platform Design and Customization: Status and Promise," *Artificial Intelligence for Engineering Design, Analysis, and Manufacturing*, vol. 18, no. 1, pp. 3-20, 2004.

[9] Gunes, E. D. and Aksin, O. Z., "Value Creation in Service Delivery: Relating Market Segmentation, Incentives, and Operational Performance," *Manufacturing & Service Operations Management*, vol. 6, no. 4, pp. 338-357, 2004.

[10] Gibbons, R., *Game Theory for Applied Economics*. Princeton, NJ: Princeton University Press, 1992.

[11] Simpson, T. W., Siddique, Z., and Jiao, J., Eds., *Product Platform and Product Family Design: Methods and Applications*. New York, YN: Springer, 2005.

[12] Moon, S. K., Park, J., Simpson, T. W., and Kumara, S. R. T., "A Dynamic Multi-Agent System Based on a Negotiation Mechanism for Product Family Design," *IEEE Transactions on Automation Science and Engineering*, vol. 5, no. 2, pp. 234-244, 2008.

[13] Kratochvil, M. and Carson, C., *Growing Modular: Mass Customization of Complex Products, Services and Software*. Heidelberg, Germany: Springer, 2005.

[14] Li, J. H., "Strategy of Mass Customization-based Services Product Innovation," presented at *IEEE International Engineering Management Conference*, Singapore, 2004.

[15] Osborne, M. J. and Rubinstein, A., *A Course in Game Theory*. Massachusetts, MA: MIT, 2002.

[16] Xiao, A., Zeng, S., Allen, J. K., Rosen, D. W., and Mistree, F., "Collaborating Multidisciplinary Decision Making using Game Theory and Design Capability Indices," presented at *9th AIAA/ISSMO Symposium on Multidisciplinary Analysis and Optimization*, 4-6, September, Atlanta, Georgia, 2002.

[17] Fernandez, M. G., Panchal, J. H., Allen, J. K., and Mistree, F., "Concise Interactions and Effective Management of Shared Design Spaces - Moving beyond Strategic Collaboration towards Co-design," *ASME International Design Engineering Technical Conference & Computers and Information in Engineering Conference*, Long Beach, CA, ASME Paper No. DETC2005-85381, 2005.

[18] Kopin, V. and Wilbur, D., "Bayesian Serial Cost Sharing," *Mathematical Social Sciences*, vol. 49, no. 2, pp. 201-220, 2005.

[19] Correia, P. F., "Games With Incomplete and Asymmetric Information in Poolco Markets," *IEEE Transactions on Power Systems*, vol. 20, no. 1, pp. 83-89, 2005.

[20] Lewis, K. and Mistree, F., "Collaborative, Sequential, and Isolated Decisions in Design," *Journal of Mechanical Engineering*, vol. 120, no. 4, pp. 643-652, 1998.

[21] Lariviere, M. A. and Van Mieghem, J. A., "Strategically Seeking Service: How Competition Can Generate Poisson Arrivals," *Manufacturing & Service Operations Management*, vol. 6, no. 1, pp. 23-40, 2004.

[22] Huang, G. Q., Zhang, X. Y., and Lo, V. H. Y., "Integrated Configuration of Platform Products and Supply Chains for Mass Customization: A Game-Theoretic Approach," *IEEE Transactions on Engineering Management*, vol. 54, no. 1, pp. 156-171, 2007.

[23] Fitzsimmons, J. A. and Fitzsimmons, M. J., *Service Management: Operations Strategy, and Information Technology*. New York, NY: McGraw-Hill, 2004.

[24] Schach, S. R., *An Introduction to Object-Oriented Analysis and Design with UML and the unified Process*. Boston, MA: McGraw-Hill/Irwin, 2004.

[25] Kamrani, A. K. and Salhieh, S. M., *Product Design for Modularity*. Boston, MA.: Kluwer Academic Publishers, 2000.

[26] da Cunha, C., Agard, B., and Kusiak, A., "Design for Cost: Module-Based Mass Customization," *IEEE Transactions on Automation Science and Engineering*, vol. 4, no. 3, pp. 350-359, 2007.

[27] Shapley, L. S., "Cores of Convex Games," *International Journal of Game Theory*, vol. 1, no. 1, pp. 111-129, 1971.

[28] Arlow, J. and Neustadt, I., *UML and the Unified Process: Practical Object-Oriented Analysis and Design*. London, UK: Addison-Wesley, 2002.