

## A user centric service-oriented modeling approach

Ding-Yuan Cheng · Kuo-Ming Chao · Chi-Chun Lo ·  
Chen-Fang Tsai

Received: 30 June 2010 / Revised: 9 January 2011  
Accepted: 24 January 2011 / Published online: 2 May 2011  
© Springer Science+Business Media, LLC 2011

**Abstract** With rapid development of service-oriented architecture and cloud computing, web services have been widely employed on the Internet. Quality of Service (QoS) is a very important criterion for service consumers to measure and select services. The selection of web services with respect to non-functional QoS criteria can be considered as a Multiple Criteria Decision Making (MCDM) problem when multiple consumers need to share a number of services. This paper describes a new user centric service-oriented modeling approach which is featured by integrating fuzzy Technique for Order Preference by Similarity to an Ideal Solution (TOPSIS) and Service Component Architecture (SCA) to facilitate web service selection and composition and to effectively satisfy a group of service consumers' subjective requirements and preferences in the dynamic environment. The main contribution of this method is able to translate a group of users' fuzzy requirements to services as well as model different levels of hardware and software as services to meet the requirements. We also design a simulated environment that includes 8\*8 LED matrix on a circuit board that corresponds to an office with different appliances to demonstrate the dynamic service selection and binding. The simulation is used to assess the computational efficiency of the fuzzy TOPSIS method and the effectiveness of the proposed system.

---

D.-Y. Cheng (✉) · C.-C. Lo  
Institute of Information Management, National Chiao-Tung University, Hsin-Chu 300, Taiwan  
e-mail: kewas@iim.nctu.edu.tw

C.-C. Lo  
e-mail: cclo@faculty.nctu.edu.tw

K.-M. Chao  
DSM Research Group, Department Computing and the Digital Environment, Coventry University,  
Coventry CV1 5FB, UK  
e-mail: k.chao@coventry.ac.uk

C.-F. Tsai  
Department of Industrial Management and Enterprise Information, Aletheia University, New Taipei City,  
Taiwan  
e-mail: tasi@email.au.edu.tw

**Keyword** SCA · fuzzy TOPSIS · service selection · multiple criteria decision making (MCDM)

## 1 Introduction

With rapid advancement of service-oriented architecture (SOA), the use of web services has made rapid growth. The advantages of Web services are highly interoperable and loosely coupled software components which can be published, located, and invoked on the web. The growing number of Web services available within an organization and on the Web raises the new challenge in search problems which involve service discovery [7, 27, 59], service selection [19, 40, 70, 74], and service composition [11, 12, 43, 50, 71, 77].

Therefore, service discovery allows service providers to publish service profiles and descriptions in UDDI repositories that store information about businesses, services and other related details. Efforts in this area focus on providing rich and machine-readable representation of service properties, capabilities, and behavior as well as reasoning mechanisms to support automated discovery. Service composition is a process to compose selected services in a logical way to meet the user requirements which can take place at the design time or runtime. A number of services can be composed as a composite service to provide required functions.

Smart home or environment often involves numerous hardware devices and software components to sense and control the environment. The integration of these devices and components is an important task. The communications and interactions among sensors, objects, and human participants, however, are inherently complex, as they often involve different protocols and languages. It can be viewed as configuration problem that requires various components to work together in cooperative and coordinated manner in order to produce an optimized environment to meet the system and user requirements. Service-Oriented Architecture (SOA), which offers a way to view sensors, objects, participants as services, can provide an effective approach to improve their communication and coordination, so the human participants can effectively interact with the system and offer their opinions and preferences through a coherent technology. SOA could be a potential approach to maximize users' satisfaction degree, and minimize the cost of software design, implementation, application installation, and system maintenance.

A web service, a supporting technology for SOA, can be invoked and executed by a number of consumers simultaneously. Most service discovery and recommending systems recommend a list of services according to the functional and/or non-functional requirements given by service consumers. Thus, the service consumer can choose the best one. This kind of cases is based on an assumption that the cardinality between an instance of service and a service consumer is built upon one to one relationship. However, in some cases such as smart environment, an instance of a service needs to serve a number of consumers. In other words, an instance of a service needs to meet various requirements specified by different service consumers. It, however, can be very difficult to design a service when conflicting requirements occur. For example, the staffs in one office use a collection of the same devices or functions provided by the system. Each function is intended as a service such as light service, air condition service, heating service, and ventilation service, etc. Service consumers could have different preferences or requirements on the services such as room temperature. Some users may subjectively feel hot for the current room temperature, but some may feel cool or comfortable. How to set the temperature for air condition/heating service according to these consumers' preferences can be a challenging issue. In addition, once users have reached a consensus on the temperature and other conditions, how to

configure and trigger these appliances or devices automatically and consistently can be another issue.

In this research, we present a new user centric service-oriented modeling method that can derive group consensus requirements from a group of individuals with inconsistent and fuzzy opinions and preferences, and translate them into required services. The method is able to systematically model software and hardware into services to meet the changing requirements.

It adopts an evaluation method based on the Technique for Order Preference by Similarity to an Ideal Solution (TOPSIS) to help service consumers to select web services with fuzzy user requirements involved. The linguistic variables, parameterized by triangular fuzzy numbers, are used for evaluating the weight of various criteria and the ratings of each alternative web service. Then the fuzzy TOPSIS method is employed to transform the linguistic terms into crisp numbers and obtain the preference order of available alternative web services. It also uses Service Component Architecture (SCA) to offer a coherent systematic process and approach to model heterogeneous hardware and software components as services in order to increase flexibility for system configuration, organization and execution in a dynamic environment. The whole process can be applied iteratively in order to mitigate the gaps in the requirements, if there is any. We designed 8\*8 LED matrix on a circuit board to simulate an office with different appliances to demonstrate the dynamic service selection, composition and binding.

The rest of this paper is organized as follows. Section 2 reviews the background and related works on existing service selection, decision making, and service composition approaches. Section 3 introduces the fuzzy TOPSIS method. Section 4 gives the detailed descriptions on the proposed user centric service-oriented modeling approach. In Section 5, a simulation for a smart environment, which involves a LED matrix device, is used to demonstrate the system based on the proposed approach. The evaluation of the proposed system and the discussions on the experimental results are given in Section 6. Final remarks and future work are reported in Section 7.

## 2 Backgrounds and related works

In this section, a review on the existing approaches in relation to web service selection and an analysis on their pros and cons will be reported. Then, we report a number of group decision making methods that can assist users in reaching consensus. We introduce the Service Component Architecture for service modeling and composition as well as Enterprise Service Bus for service management and service transportation.

### 2.1 Web service selection

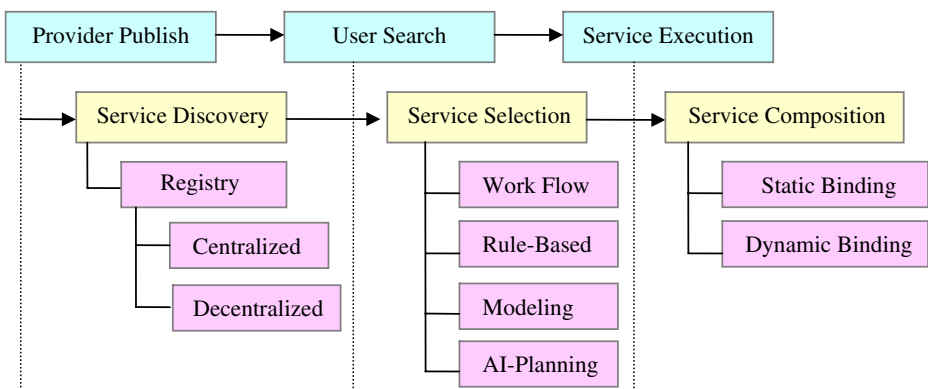
Service contains a collection of functions. Services can be classified into atomic or composite services. An atomic service is a basic unit which cannot be decomposed further. An atomic service on its own normally cannot satisfy the full requirements given by the users, so they need to be composed. A composite service is made of a collection of existing services which can be atomic or composite. Service selection is an important step in service composition. Service providers can publish their services on UDDI repositories, so a consumer could search her/his required services according to some keywords or other criteria. However, the consumer may discover a collection of services providing same function, so a service selection process based on non-functional attributes to refine possible services is required. Once the appropriate services have been identified and selected, the composition process can take place [36, 51, 53, 60, 64].

The following diagram shows their relationship (see Figure 1). The rest of this section will give more detailed reviews on these topics.

Many researches focus on facilitating service-oriented system development by proposing different service discovery protocols [20, 23, 30, 61, 67] to increase accuracy in service discovery and selection. One of the most widely used protocols for publishing service is Universal Description, Discovery and Integration (UDDI) specification [5, 42] that enables service specification to be advertised in a service registry. In our previous studies on the UDDI specification [41], we used the tModel to represent the QoS for web services discovery, selection, and composition. Several researchers have designed the semantic query mechanism into UDDI registry to map RDFS upper concepts with semantics to the UDDI data model in order to increase the precision in service selection [2, 3, 39].

Catalogues and P2P solutions are the well-known techniques in Web Service Discovery [38]. In Catalogues, the services are stored in a central node which provides a standardized specification framework as meta-schema to describe services for service discovery and advertisement such as UDDI. When a number of UDDI nodes exist in a network and can be linked together, so they become a decentralized architecture and can be regarded as a virtual node, this is called UDDI cloud or federation [56]. P2P-based Web service Discovery (PWS) presented in [58, 65] is built upon Peer-to-Peer (P2P) technologies which provides a distributed environment and infrastructure for service routing and service locating. So each node in the network provides routing and data location service and acts as a server providing service access. An agent-based solution proposed in [49] called DASD (DAML Agents for Service Discovery) enhances effectiveness in service discovery and selection with an intelligent mechanism. It provides a matchmaking service that allows web service consumers and providers to discover each other more efficient.

A number of researchers have introduced different approaches such as workflow-based approach, Artificial Intelligent (AI) planning, and other modeling methodologies to the service composition process. Semantic Web Service composition methods are based on Model-Driven Architecture (MDA) [6, 57, 77], and UML [4]. These composite services are specified by using standard UML model to generate system specifications and using MDA approaches to produce applications. Composing services to form an application is a complex and error-prone task, so researchers in AI have proposed different planning techniques in an attempt to automate the composition process.



**Figure 1** Research issue and evolution on web service.

Most existing service discovery or selection techniques [53, 54, 78], however, do not take into account service customers' preferences and opinions. Huang et al. [25] applied semi-order preference model and content-based service discovery concept to improve effectiveness in service discovery and selection. Their work attempts to assist service providers and consumers in discovering appropriate services by considering their opinions and preferences. Wang et al. [70] introduced a new QoS-aware service selection approach by including a fuzzy model to prioritize the service consumers' and providers' QoS preferences as they often cannot be expressed precisely and they are normally associated with uncertainty. Zaki et al. [44] not only add a weight to reduce malicious ratings but also enable rating of services automatically.

## 2.2 Multiple criteria decision making

Decision making can be one of the most complicated administrative processes in management. In the decision-making process, decision makers may have to make choices based on multiple criteria, so Multiple Criteria Decision Making (MCDM) is one of the most well known branches in decision making. MCDM can be divided into two categories: MODM and MADM. A further discussion about MODM and MADM can refer to Hwang and Yoon [29]. Thus, MADM has been widely used by decision makers in management processes to evaluate and rank possible alternatives.

A decision making process could involve a group of decision makers called Group Decision Making (GDM), so members' opinions or preferences have to be considered. Most of the GDM problems are of strategic dimensions and can be complicated due to their multi-criteria framework involving many subjective and qualitative factors. Optimal utilization of the time and resources is a key element sought by many GDM methods. Various researchers have focused their attention on increasing the ability of the group to make the quality decisions [6, 9, 10, 22, 24, 29, 31, 32, 63, 73].

An effective web service discovery mechanism should be able to search and assess services based on their QoS and service contents as well as users' functional requirements. The service assessment or selection often involves multi-criteria decision-making process [34, 75]. So, the GDM is applicable to service selection when the service consumers have inconsistent or conflicting requirements. It is a reasoning process to reach group consensus on the requirements for service selection.

TOPSIS method has become a popular multiple criteria decision technique due to (1) its theoretical rigorousness [59], (2) a sound logic that represents the human rationale in selection [62], and (3) the fact that it has been proved in [44] as one of the best methods in solving rank reversal. Recently, some researchers have focused on developing fuzzy TOPSIS methods to deal with imprecise information [21, 72]. Sun et al. [66] applied fuzzy TOPSIS to evaluate the competitive advantages of shopping websites. Chamodrakas et al. [13, 76] employed fuzzy TOPSIS to help suppliers evaluate customers within the purchase order acceptance process so that the resource allocation and the priority of orders could be optimally identified. Chu et al. [17] designed a fuzzy TOPSIS model based on interval arithmetic of fuzzy numbers. Kahraman et al. [33] proposed an interactive group decision making methodology based on fuzzy TOPSIS to select information system providers under multiple criteria. Chen et al. [14] extended the TOPSIS method based on interval-valued fuzzy sets in decision analysis. Abo-Sinna et al. [1] extended the TOPSIS approach to solve multi-objective large-scale non-linear programming problems with block angular structure. Lin et al. [39] applied fuzzy TOPSIS for order selection and pricing for make-to-order products when orders exceed production capacity. Li [37] developed a Compromise Ratio

(CR) methodology for fuzzy multi-attribute group decision making (FMAGDM), which is an important part of decision support system. Wang and Chang [68, 70] utilized fuzzy TOPSIS to help the Air Force Academy in Taiwan select optimal initial training aircraft under fuzzy environment. Wang and Lee [69] generalized TOPSIS to fuzzy multiple-criteria group decision-making by proposing two operators, Up and Low, which satisfy the partial ordering relation on fuzzy numbers, to find positive and negative ideal solutions.

### 2.3 Service composition

To achieve dynamically service composition, in this research it entails two important technologies namely Service Component Architecture and Service Execution Engine. We explain their characteristics in the following subsections.

#### 2.3.1 Service component architecture

Service Component Architecture (SCA) [52] is a set of specifications which describe a model for building applications and systems by composing and deploying new and existing service components. SCA builds upon an open standard such as Web services and complements to the existing approaches in service implementation. It is based on the idea that business function is supported by a series of services, which can be assembled together to create solutions and to serve a particular business need. The composite applications can contain both new services and business functions from existing systems and applications [15, 18, 80].

Furthermore, SCA is aimed to encompass a wide range of technologies to support service component accessibility and web service connectivity. For components, this includes not only different programming languages, but also framework and environments commonly used with those languages. These binding technologies include Web services, Message systems and CORBA IIOP. Bindings are handled declaratively and independent of the implementation.

The SCA specifications are divided into a number of documents such as assembly model, policy framework, transaction policy, Web Service binding, JMS binding, and etc. and each document is responsible to one aspect of SCA. SCA assembly model deals with the aggregation of components and their linkages. The Assembly Model is independent of implementation language which deals with the aggregation of components and the linking of components through wiring using composites. In SCA, a composite is a collection, or assembly, of components or services. A composite considered a “service” when it’s wrapped for external consumption. Furthermore, a composite can contain properties and reference to other components or services. A component is not necessarily intended to be exposed for external consumption. A service in SCA is a self-contained unit of functionality. The benefits of building software based on the reusable building blocks of components and composites have been espoused for several decades. Moreover, SCA does not only extend and complement prior approaches to implementing services, but also provide a programming model for building applications and systems based on a SOA.

#### 2.3.2 Enterprise service bus

The heritage of ESBs is derived from enterprise application integration (EAI). Enterprise Service Bus (ESB) provides a run time environment for service binding and execution environment. One important advantage is that an ESB allows architects to exploit the value of messaging without writing code. In an enterprise application system making use of an ESB, an

application will communicate via the bus, which acts as a message broker between applications. When the ESB receives a message, it routes the message to the appropriate application.

ESB supports flexible data transformation and intelligent routing which can provide content-based routing based on message properties or inline statements in XML via XPath and XSLT. Furthermore, the custom API enables developers to add customized adapters or components in the ESB. Individual applications or services can be attached to the bus via adapters. The architecture of ESB is shown as follows (see Figure 2).

### 3 A fuzzy topsis method for web service selection

The Technique for Order Preference by Similarity to Ideal Solution (TOPSIS) method, which is initially proposed by Hwang et al. [28], is one of the best-known multiple criteria decision making (MCDM) methods. The TOPSIS method introduces the shortest distance from the positive ideal solution (PIS) and the farthest distance from the negative ideal solution (NIS) to determine the best alternative. While the PIS is to maximize benefit criteria and minimize cost criteria, the NIS is to maximize cost criteria and minimize benefit criteria.

TOPSIS uses triangular fuzzy numbers representing linguistic variables of the weights of criteria and of the ratings of web services which can be transformed into crisp numbers [35, 45, 46, 55, 79]. The transformation is performed by the graded mean integration representation method proposed by Chou [16]. The canonical representation of the addition and the multiplication operations on triangular fuzzy numbers is then used to obtain the PIS and the NIS. To avoid the problem with doubling weightings on each alternative, the distance of each alternative web service from the PIS and the NIS is measured by Minkowski distance function [47, 48]. As a result, the preference order of alternatives can be identified accordingly. Based on the graded mean integration representation of triangular fuzzy numbers and the canonical representation of addition and multiplication on triangular fuzzy numbers, the procedure of fuzzy TOPSIS method can be performed in an efficient way. This property can reduce the computational complexity in the decision making process.

#### 3.1 Fuzzy number

In this section, some basic and related definitions of fuzzy sets will be briefly reviewed.

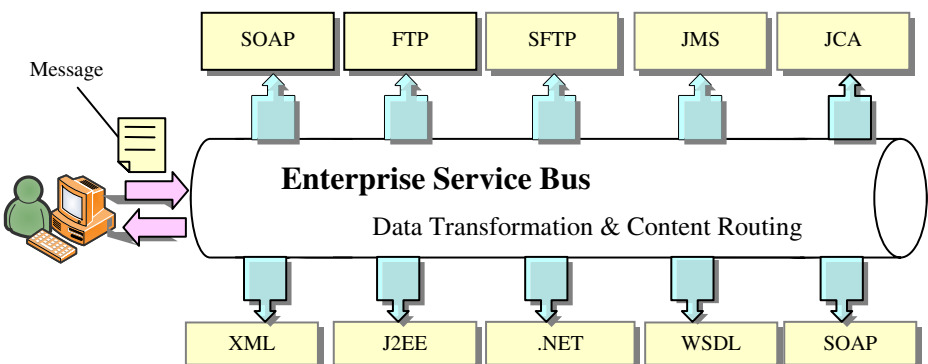


Figure 2 Architecture of enterprise service bus.

**Definition 1** *Fuzzy set.* Let  $X$  be a universe of discourse. Where  $\tilde{A}$  is a fuzzy subset of  $X$ ; and for all  $x \in X$ , there is a number  $\mu_{\tilde{A}}(x) \in [0, 1]$  which is assigned to represent the membership of  $x$  in  $\tilde{A}$ , and is called the membership of  $\tilde{A}$ .

**Definition 2** *Normality.* A fuzzy subset  $\tilde{A}$  is normal if and only if

$$\exists x \in \mathbb{R}, \forall \mu_{\tilde{A}}(x) = 1. \tag{1}$$

**Definition 3** *Convexity.* A fuzzy subset  $\tilde{A}$  is convex if and only if

$$\mu_{\tilde{A}}(\lambda x_1 + (1 - \lambda)x_2) \geq \mu_{\tilde{A}}(x_1) \wedge \mu_{\tilde{A}}(x_2), \forall x_1, x_2 \in X, \forall \lambda \in [0, 1]. \tag{2}$$

**Definition 4** *Fuzzy number.* A fuzzy number  $\tilde{A}$  is a normal and convex fuzzy subset of  $X$ .

**Definition 5** *Triangular fuzzy number.* Let  $\tilde{A}$  as a fuzzy set and its values will be located between 0 and 1. It is a triangular fuzzy number  $\tilde{A}$  can be defined by a triplet  $(a, b, c)$ , as shown in Figure 3. Its membership function is defined as (Equation 3)

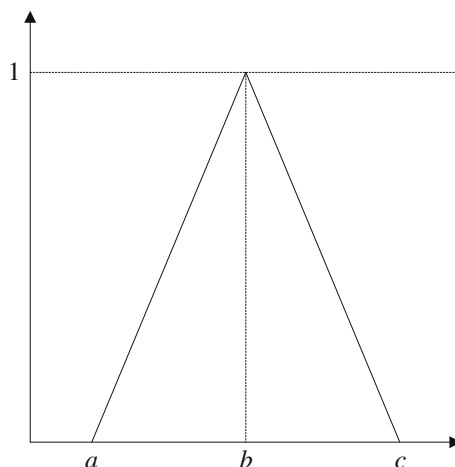
$$\mu_{\tilde{A}} = \begin{cases} 0, & \text{if } x < a, \\ \frac{x-a}{b-a}, & \text{if } a \leq x \leq b, \\ \frac{c-x}{c-b}, & \text{if } b \leq x \leq c, \\ 0, & \text{if } x > c. \end{cases} \tag{3}$$

**Definition 6** *The graded mean integration representation on triangular fuzzy numbers.* Let  $\tilde{A} = (a, b, c)$  be a triangular fuzzy number. The graded mean integration representation of  $\tilde{A}$  is defined as

$$P(\tilde{A}) = \frac{1}{6}(a + 4b + c).$$

**Definition 7** *The canonical representation of addition and multiplication operations on triangular fuzzy numbers.* Let  $\tilde{A} = (a_1, b_1, c_1)$ , and  $\tilde{B} = (a_2, b_2, c_2)$  be two triangular fuzzy

**Figure 3** Triangle membership function.





numbers. Then the canonical representation of addition and multiplication operations on triangular fuzzy numbers can be defined as follows.

*Addition operation*  $\oplus$ :

$$\begin{aligned}
 P(\tilde{A} \oplus \tilde{B}) &= P(\tilde{A}) + P(\tilde{B}) \\
 &= \frac{1}{6}(a_1 + 4b_1 + c_1 + a_2 + 4b_2 + c_2)
 \end{aligned}
 \tag{4}$$

*Multiplication operation*  $\otimes$ :

$$\begin{aligned}
 P(\tilde{A} \otimes \tilde{B}) &= P(\tilde{A}) \times P(\tilde{B}) \\
 &= \frac{1}{6}(a_1 + 4b_1 + c_1) \times \frac{1}{6}(a_2 + 4b_2 + c_2)
 \end{aligned}
 \tag{5}$$

Two important operators the addition and the multiplication operations on triangular fuzzy numbers can be used to obtain the PIS and the NIS. The use of graded mean integration representation method is to transform a triangular fuzzy number into a crisp number which can reduce the required computation under fuzzy environment. Similarly, it can also provide the ability of transformation trapezoidal fuzzy number into a crisp number. For more detailed information, it can be found in [15].

In evaluation processes, human users intend to use natural languages to express their opinions rather than exact numbers. Linguistic variables can be defined to represent evaluations [8]. Then each linguistic variable can be parameterized into a fuzzy set. For example, the importance weights of various criteria and the ratings of the criteria can be expressed as linguistic variables which can be modeled as triangular fuzzy numbers as shown in Tables 1, 2 and 3, respectively.

Once the users express their opinions based on the above tables, Eqs. 2, and 3 can be applied to obtain the graded mean integration representation of the importance weight of each criterion and ratings shown in Tables 4 and 5 to form the importance weight of each criterion and each alternative rating.

### 3.2 A fuzzy TOPSIS method for web service selection

Assume that a group of  $k$  users ( $D_1, D_2, \dots, D_k$ ) is formed for ranking  $m$  alternatives of web service ( $A_1, A_2, \dots, A_m$ ) with respect to  $n$  criteria ( $C_1, C_2, \dots, C_n$ ). Then the decision matrix,  $R_t$ , given by decision maker,  $d_t, t=1, 2, \dots, k$ , is as follows.

$$R_t = \begin{matrix} & \begin{matrix} C_1 & C_2 & \cdots & C_n \end{matrix} \\ \begin{matrix} A_1 \\ A_2 \\ \vdots \\ A_m \end{matrix} & \begin{bmatrix} r_{11t} & r_{12t} & \cdots & r_{1nt} \\ r_{21t} & r_{22t} & \cdots & r_{2nt} \\ \vdots & \vdots & \ddots & \vdots \\ r_{m1t} & r_{m2t} & \cdots & r_{mnt} \end{bmatrix} \end{matrix}
 \tag{6}$$

**Table 1** Linguistic variables for the importance weight of each criterion.

Linguistic variables	Importance weight of each criterion
Very low (VL)	(0.0, 0.1, 0.3)
Low (L)	(0.1, 0.3, 0.5)
Medium (M)	(0.3, 0.5, 0.7)
High (H)	(0.5, 0.7, 0.9)
Very high (VH)	(0.7, 0.9, 1.0)

**Table 2** Linguistic variables for the rating of each alternative.

Linguistic variables	Importance weight of each alternative
Very poor (VP)	(1, 1, 3)
Poor (P)	(1, 3, 5)
Fair (F)	(3, 5, 7)
Good (G)	(5, 7, 9)
Very good (VG)	(7, 9, 9)

We use  $r_{ijt} = (o_{ijt}, p_{ijt}, q_{ijt})$ ,  $r_{ijt} \in \mathfrak{R}^+$ ,  $i=1, 2, \dots, m; j=1, 2, \dots, n; t=1, 2, \dots, k$ ; to denote the rating of alternative  $a_i$  with respect to criterion  $C_j$  given by the user  $d_t$ .

The procedure of the fuzzy TOPSIS method is stated as follows.

Step 1 aggregate the importance weights. Let  $w_{jt} = (a_{jt}, b_{jt}, c_{jt}), j=1, 2, \dots, n; t=1, 2, \dots, k$ ; be the importance weight of criterion  $C_j$  given by the user  $d_t$ . Then we can calculate the aggregated crisp weight  $W_j$  of criterion  $C_j$  by the following formula:

$$W_j = \frac{\sum_{t=1}^k w'_{jt}}{k}, \tag{7}$$

where  $w'_{jt}$  is the weight derived from the graded mean integration representation of fuzzy numbers, as illustrated in Eq. 2.

Step 2 aggregate rating of alternatives. The following formula is used to obtain the aggregated crisp rating of alternatives  $R_{ij}$ .

$$R_{ij} = \frac{\sum_{t=1}^k r'_{ijt}}{k}, \tag{8}$$

where  $r'_{ijt}$  is obtained by the graded mean integration representation of fuzzy numbers, as illustrated in Eq. 2.

Step 3 construct the normalized and weighted decision matrix. Let  $S = [s_{ij}]_{m \times n}$  be the normalized decision matrix. We can calculate the normalized value  $s_{ij}$  via the following formula.

$$s_{ij} = \frac{R_{ij}}{\sqrt{\sum_{i=1}^m (R_{ij})^2}}. \tag{9}$$

**Table 3** Graded mean integration representation for the importance weight of each criterion.

Linguistic variables	Importance weight of each criterion
Very low (VL)	0.1167
Low (L)	0.3000
Medium (M)	0.5000
High (H)	0.7000
Very high (VH)	0.8833

**Table 4** Linguistic variables for the rating of each alternative.

Linguistic variables	Importance weight of each criterion
Very poor (VP)	1.3333
Poor (P)	3.0000
Fair (F)	5.0000
Good (G)	7.0000
Very good (VG)	8.6667

Let  $V = [v_{ij}]_{m \times n}$  be the weighted decision matrix. The weighted value  $v_{ij}$  is derived from the product of elements in the normalized decision matrix and crisp weights.

$$v_{ij} = W_j \cdot s_{ij} \tag{10}$$

Step 4 determine the Positive Ideal Solution (PIS) and the Negative Ideal Solution (NIS). Let  $I$  and  $J$  be the index sets associated with the alternative set and the criterion set, respectively. We can gain the PIS,  $A^+$ , and the NIS,  $A^-$ , from the following methods.

$$A^+ = \{v_1^+, v_2^+, \dots, v_n^+\} = \{\max_{i \in I} v_{ij} | j \in J\} \tag{11}$$

$$A^- = \{v_1^-, v_2^-, \dots, v_n^-\} = \left\{ \min_{i \in I} v_{ij} | j \in J \right\} \tag{12}$$

Step 5 measure the distance of each alternative from the PIS and the NIS respectively. Traditionally, the Euclidean distance is used to measure the distance of each alternative from  $A^+$  and  $A^-$  as follows.

$$d_i^+ = \sqrt{\sum_{j=1}^n (v_{ij} - v_j^+)^2}, i = 1, 2, \dots, m, \tag{13}$$

**Table 5** Ratings by users with respect to criteria.

Criteria	Candidates	Users				
		$D_1$	$D_2$	$D_3$	...	$D_{10}$
$C_1$	$A_1$	F	VG	G	...	G
	$A_2$	G	F	VG	...	VG
	...	...	...	...	...	VG
	$A_{10}$	G	G	F	...	F
$C_2$	$A_1$	G	F	G	...	G
	...	...	...	...	...	G
	$A_{10}$	P	G	F	...	F
$C_3$	$A_1$	F	G	G	...	G
	...	...	...	...	...	...
	$A_{10}$	P	F	G	...	G

$$d_i^- = \sqrt{\sum_{j=1}^n (v_{ij} - v_j^-)^2}, i = 1, 2, \dots, m. \tag{14}$$

However, the use of the Euclidean distance may have the problem associated with weight having been calculated twice. This problem can be resolved by introducing Eq. 13 or Eq. 14 as follows.

From Eq. 14, we can easily observe that the decision results overly controlled by weighting.

$$d_i^+ = \sqrt{\sum_{j=1}^n (v_{ij} - v_j^+)^2} = \sqrt{\sum_{j=1}^n (W_j s_{ij} - W_j s_j^+)^2} = \sqrt{\sum_{j=1}^n W_j^2 (s_{ij} - s_j^+)^2}. \tag{15}$$

Therefore, this problem can be overcome by means of Minkowski distance [47, 48],  $L_p^w$ , as follows.

$$L_p^w(x, y) = \left[ \sum_{j=1}^n w_j |x_j - y_j|^p \right]^{1/p}, \tag{16}$$

where  $w_j$  is the weight of importance with respect to the  $j$ -th criterion and  $p \geq 1$ .

Note that  $L_p^w$  with  $p=2$  is known as the weighted Euclidean distance.

Based on the weighted Euclidean distance,  $A^+$  and  $A^-$  can be redefined as follows. Recall that  $S = [s_{ij}]$  is the normalized decision matrix. Define

$$A^+ = \{s_1^+, s_2^+, \dots, s_n^+\} = \{(\max_{i \in I} s_{ij} | j \in J)\} \tag{17}$$

$$A^- = \{s_1^-, s_2^-, \dots, s_n^-\} = \{(\min_{i \in I} s_{ij} | j \in J)\}, \tag{18}$$

and then the distance of each alternative from  $A^+$  and  $A^-$  based on the weighted Euclidean distance is computed as

$$d_i^+ = \sqrt{\sum_{j=1}^n W_j |s_{ij} - s_j^+|^2}, i = 1, 2, \dots, m, \tag{19}$$

$$d_i^- = \sqrt{\sum_{j=1}^n W_j |s_{ij} - s_j^-|^2}, i = 1, 2, \dots, m \tag{20}$$

Step 6 Calculate the relative closeness coefficient and rank the preference order. The relative closeness coefficient of the  $i$ -th alternative,  $RCC_i$ , can be computed by

$$RCC_i = \frac{d_i^-}{d_i^+ + d_i^-} \tag{21}$$

Consequently, the alternatives can be ranked according to  $RCC_i$ .

In this research, we employ the fuzzy TOPSIS method to evaluate web services for selection. It helps this research to identify an idea solution across a set of service

QoS criteria by maximizing their acceptable preferences, while satisfying the QoS constraints given by a group of consumers.

#### 4 The proposed user centric service-oriented modeling approach

This research is to propose a user-centric service-oriented modeling approach including the aforementioned SCA principles and TOPSIS method to form an architecture that enables system developers to model hardware and software components as services and to work together seamlessly. The approach is able to efficiently satisfy a group of participating service consumers' subjective requirements and preferences in a dynamic environment and it can also effectively facilitate service development. The proposed system architecture shown in Figure 4 comprises of a number components, including Enterprise Server Bus (ESB), TOPSIS evaluation service, data adapter service, collection service, SCA application server (Tuscany server), Web Service registry, group preference database. Furthermore, the overall architecture can be classified as Service Providers (the right side) and Service Consumers (the left side), and a middleware. The process flow of the overall architecture is depicted in Figure 4.

We use a sequence diagram to help readers understand how services are activated and adjusted to meet users' requirements. The main processes for the proposed architecture are divided into 7 steps which are illustrated as follows (Figure 5):

- Step 1 Users prescribe their functional requirements and their QoS preferences. Users use linguistic variables to rate all possible alternatives and give a weight to each criterion based on their importance.
- Step 2 Collection service gathers the users' preferences in fuzzy terms, transforms them into RDFs and sends them to a data adapter service.
- Step 3 The Data adapter service stores users' preferences send by the collection service into database. Then, the data adapter service is responsible for maintaining the existing users' preferences consistently as well as new ones. In addition, the adapter service makes sure that the data can be accessible coherently regardless

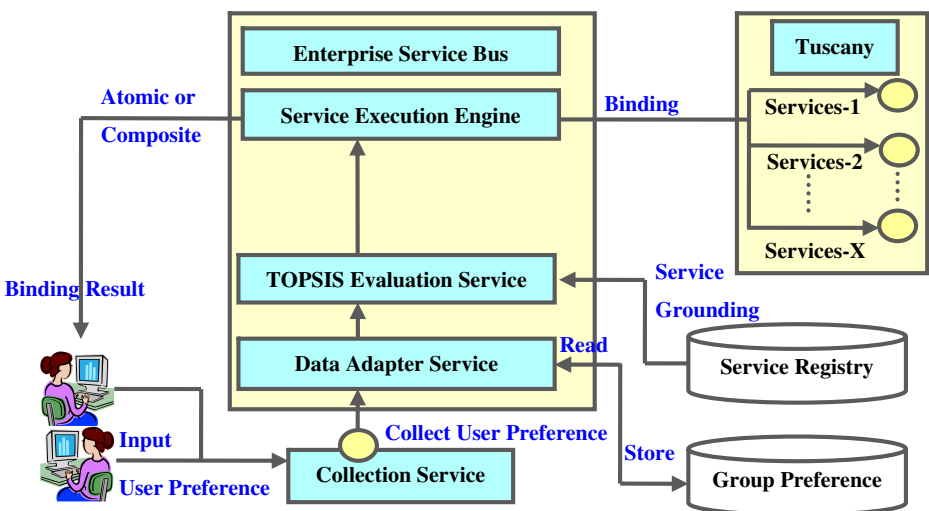
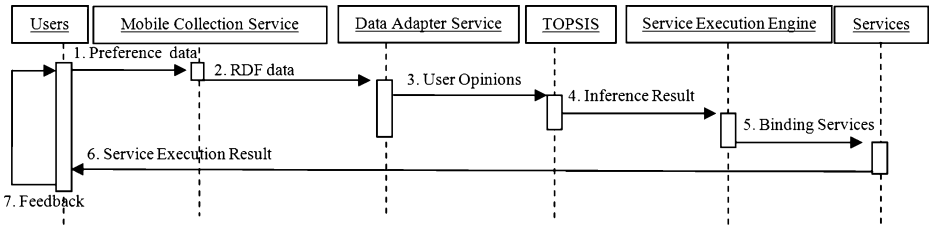


Figure 4 Prototype system architecture.



**Figure 5** Sequence diagram of service support the consensus-adjusting between ESB and users.

their locations and physical database structures. In the end, the data adapter service makes the data available to the TOPSIS evaluation service to reason.

- Step 4 The weight corresponding to importance of a criterion is represented by linguistic variables which are defined as triangle fuzzy numbers. Then, we transform a triangular fuzzy number into a crisp number through a graded mean integration method [16] which is defined in Eqs. 4 and 5. The TOPSIS is able to reason over the values relating to QoS criteria and alternatives to identify the possible group consensus by recommending appropriate services. The recommended services must be able to satisfy most users’ requirements. If not, it means that the group cannot reach a consensus.
- Step 5 In the previous step, the recommended services are abstract services. Enterprise Service Bus (ESB) can bind these abstract services to the physical ones. In ESB, the Service Execution Engine (SEE) locates the designated services which can be a collection of atomic and composite services.
- Step 6 The service is located, invoked, and shows the result to users. The service normally runs on a web container which provides essential library for a runtime environment.
- Step 7 Once users received the service, they can assess the results according to functional and nonfunctional requirements. If the services cannot satisfy the requirements, the user can give their opinions and preferences again. These feedbacks will be included in the system as inputs for the next round of group consensus. The above steps will be carried out again in order to reflect new group requirements.

In our development, we adopted an open source project which is Synapse to support the required enterprise service bus. We also used Tuscany to realize SCA development environment. With these enabling technologies and middleware along with other modules, the services can be advertised, discovered, selected, composed, and executed according to the requirements. The following subsections will offer more details on these components.

### 4.1 Module design

There are several modules in our proposed architecture that depicts in the above figure. The system functionality and key modules are described as follows.

#### 4.1.1 Service execution engine

We use an open source system, Synapse, which is an open source based on Apache web server to be our Enterprise Service Bus (ESB). ESB is not only a web server container but also can forward the message to other web services accordingly which are deployed on the other server.

Synapse provides a simple, lightweight and fully open source SOA infrastructure to assemble and manage composite applications as well as route message. Synapse supports HTTP, SOAP, SMTP, JMS, FTP and file system transport for message exchange using XSLT, XPath and XQuery to bind the web services and to relate each other. Figure 6 shows the ESB architecture.

#### 4.1.2 Data adapter service

The purpose of data adapter service module is to output Service Data Object (SDO) in XML format so that web service can manipulate data between database and application. Service data objects (SDO) that represent XSLT data output include all data content (e.g., attribute values). Those objects later serve as input parameters for the TOPSIS evaluation service. Moreover, user preference received from collection service can be stored into group preference database. The SDO is a technology which is a language-independent representation of a data entity that can be passed between services. Within the SCA, the SDO provides common and simple APIs which manipulate data from heterogeneous sources including relational databases, XML data, or web services, etc.

SDO is the input/output of data adapter service. Here, we use Service Data Objects (SDO) to be our implementation foundation. There are three key elements inside the SDO which are data object, data graph, and data access service (DAS). The source of data graph can be XML file, web service, or Enterprise Java Bean et al. (Figure 7).

The SDO mediator reads data from difference sources (database, flat file or proprietary data APIs) for conversion into data graph. Data graph that represent SDO data adapter input include all data content (e.g., attribute values). Those objects later serve as input parameter for the SDO query engine and XML generator. The service router can forward and send the query string to one or another web service. Moreover, the result of invoked web service could also be another data source, and then, the difference web service can invoked synchronous at the same time for one purpose or asynchronous for service composition. The operation concept is shown as Figure 8.

#### 4.1.3 Collection service

Since users need to express their preferences over their required services, there is a GUI for them to input their opinions. A collection service would gather these inputs and send them to a data adapter service.

#### 4.1.4 Service registry

The aim of service registry is to store service profiles based on the functionalities provided by the web services. These services are registered in a service repository which is UDDI.

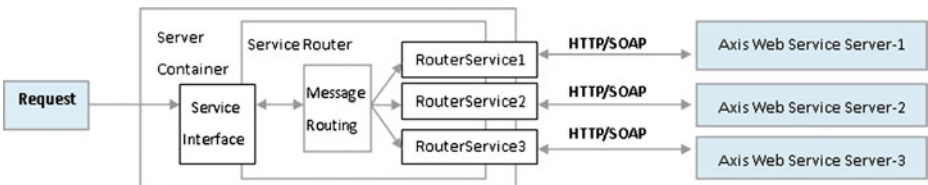


Figure 6 Architecture of enterprise service bus architecture.

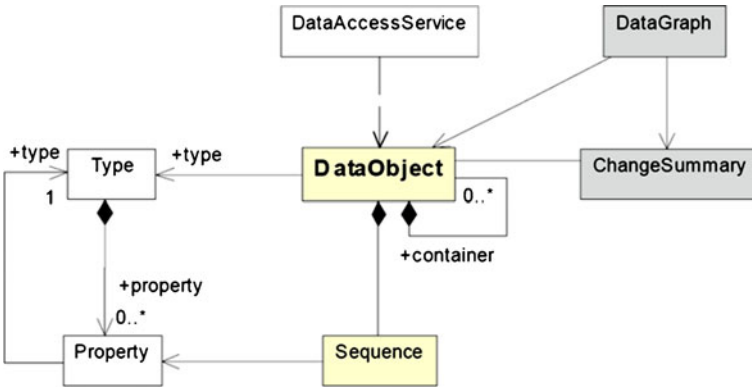


Figure 7 SDO API meta-diagram.

JUDDI, an open source tool, is adopted to support UDDI. It allows the services to represent their business details by providing a number of facilities such as Business Entities, Service Entities, Binding Templates, and tModels. Services registered with in JUDDI can be searched by name, location, business, bindings or tModels. However, the JUDDI or UDDI specification does not offer any facility to characterize service QoS profile. Therefore, we enhanced JUDDI by introducing a QoS tModel to model service QoS profile in details. So, the services can be searched according to QoS requirement.

4.1.5 Group preference database

The group preference database stores the users’ preferences (e.g., Very Cold, Normal, Warm, Hot, Very Hot) and the corresponding user profile in a relational database. The data can only be accessible to the data adapter service to ensure the data consistency. When a new user opinion occurs and needs to be included in the system, all the data in the group preference database will be retrieved by the DAS and be conveyed to the evaluation service to carry out the reasoning process in order to derive a new group consensus, if there is any.

4.1.6 Consensus evaluation service-fuzzy TOSIS method

We design a consensus evaluation service which is based on the Fuzzy TOPSIS method to reason over a group of users’ preferences to identify their potential agreements. These preferences can be very subjective and inconsistent and they could be represented in different ways. The preferences can be associated with uncertainty, fuzziness, and

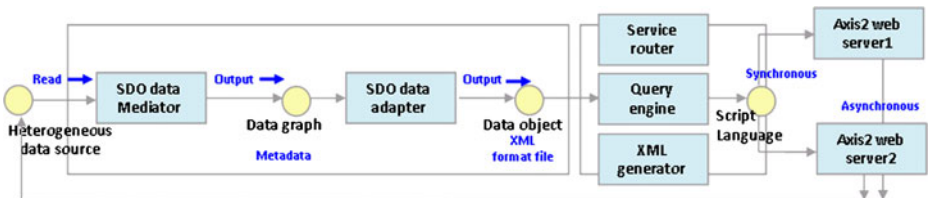


Figure 8 Data adapter service operation flow.



incompleteness. The consensus evaluation service, which obtains the data from the DAS, is able to identify common requests from the majority of users and made recommendations to the users or other services for them to dispose.

## 5 Cast study

In this section, we offer an example to demonstrate how the proposed user centric service-oriented modeling approach functions. It includes a TOPSIS method in a consensus evaluation service to resolve potential conflicting opinions from the users. Then, we describe how the proposed approach can model physical devices (components) as services to be composed dynamically in order to meet the consumers' requirements.

Section 3.2 shows how the TOPSIS process is applied to assist in reaching the group consensus on a number of alternatives. This section presents the overall architecture and system flows to illustrate how the proposed approach is able to model and compose services according to users' requirements. In this case study, 30 services have been designed to control different devices and the UDDI Registry has these service profiles for search and selection. There are 26 services corresponding to different lighting devices in an office, two services control two air-condition appliances and two dehumidifier services control the humidity level in the office. In this experiment, we only allow these services to be composed in 10 different composite services. In other words, one of these alternatives can be selected to provide services. There are 10 staffs in the office and they are service consumers who can express their preferences and opinions in order to control settings on these devices. Each alternative in our case refers to a composite service, as each device is independent and is regarded as a service. The required components and the overall simulation process are shown in Figure 9.

We designed a prototype system to demonstrate and evaluate the proposed user-centric service-oriented modeling approach. In the system, these 10 composite services (alternatives) can be classified into three different types which are LightingComposite, AirconditionComposite, and DehumidifierComposite services. The LightingComposite service is designed for controlling the brightness. Here, we view one LED as an atomic service, and one LED receives the command from the corresponding web service to switch it on/off. Similarly, AirconditionComposite service is associated with Air Condition device (component) and is designed to control it for setting the office temperature. DehumidifierComposite service works with the dehumidifier to manage the humid level in a space. The communication protocol, which allows the interactions between services and devices to take place, is represented in XML. We model these composite services by using SCA diagram which is shown in Figure 10.

Once all of the components have been transformed to services, the system is ready to get the service consumers involved. The system starts with the collection service by gathering the consumers' preferences, ranking alternatives accordingly, composing services, and executing them. The process involves these steps which are carried out in a linear style. The whole process iterates when the new inputs (opinions) emerge. The rest of this section illustrates the whole process in the context of the case study (Figure 11).

**Step 2** The Collection service transforms the users' preferences in fuzzy terms and their profiles into RDFs. For example, we can store the user name and the linguistic ratings on the alternatives for each criterion. The following RDF sample shows a user giving two preference values (good and very good) to two different criteria (brightness and temperature) (Figure 12).

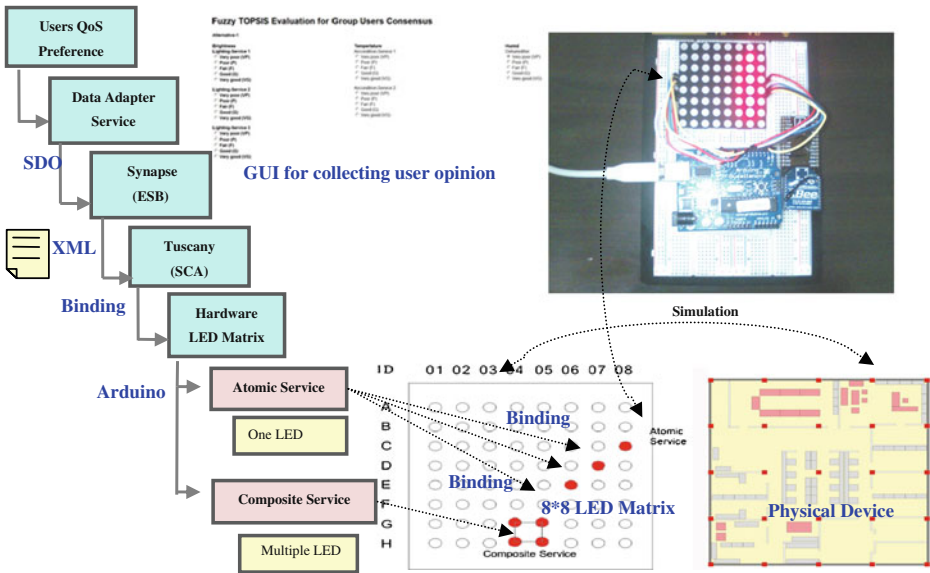


Figure 9 LED matrix and simulation device.

Step 3 Once the collection service has completed the transformation, the RDF will pass on to the data adapter service. The key task for the data service adapter is to store the information into the preference database consistently. It adopts SDO objects to alleviate the barriers that could be caused by heterogeneous database systems. It represents the information into graphs. Figure 13 shows the code to allow the SDO to retrieve user preferences from a relational database via a SQL statement and store them in a graph for the consensus evaluation service.

Step 4 We design a consensus evaluation service which is based on the Fuzzy TOPSIS method to reason over a group of users' preferences to identify their potential agreements. The general calculation process is described as follows. Assume that ten alternatives of web services  $A_i, i = 1, 2, 3, \dots, 10$  are chosen for evaluation. A group of ten users,  $D_k, k = 1, 2, 3, \dots, 10$  have been formed to conduct the assessment

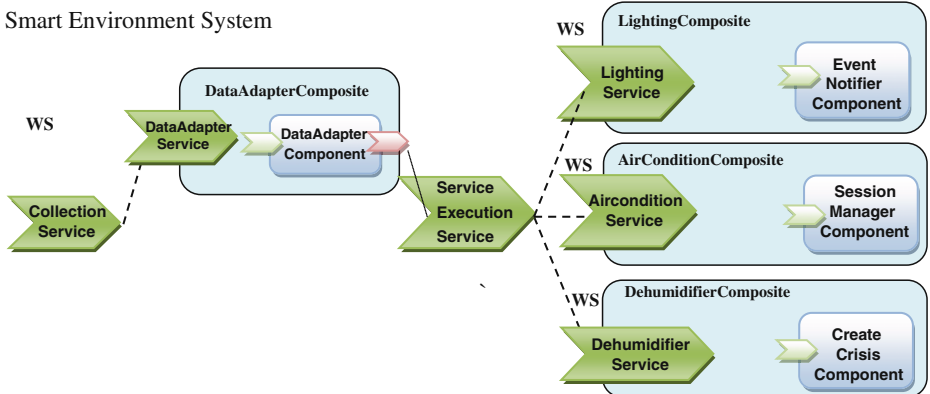


Figure 10 Smart environment system based on SCA designed-principle.



### Fuzzy TOPSIS Evaluation for Group Users Consensus

<p><b>Brightness</b></p> <p><b>Lighting-Service 1</b></p> <p><input type="radio"/> Very poor (VP)</p> <p><input type="radio"/> Poor (P)</p> <p><input type="radio"/> Fair (F)</p> <p><input type="radio"/> Good (G)</p> <p><input type="radio"/> Very good (VG)</p> <p><b>Lighting-Service 2</b></p> <p><input type="radio"/> Very poor (VP)</p> <p><input type="radio"/> Poor (P)</p> <p><input type="radio"/> Fair (F)</p> <p><input type="radio"/> Good (G)</p> <p><input type="radio"/> Very good (VG)</p> <p><b>Lighting-Service 3</b></p> <p><input type="radio"/> Very poor (VP)</p> <p><input type="radio"/> Poor (P)</p> <p><input type="radio"/> Fair (F)</p> <p><input type="radio"/> Good (G)</p> <p><input type="radio"/> Very good (VG)</p> <p><b>Lighting-Service 4</b></p> <p><input type="radio"/> Very poor (VP)</p>	<p><b>Temperature</b></p> <p><b>Aircondition-Service 1</b></p> <p><input type="radio"/> Very poor (VP)</p> <p><input type="radio"/> Poor (P)</p> <p><input type="radio"/> Fair (F)</p> <p><input type="radio"/> Good (G)</p> <p><input type="radio"/> Very good (VG)</p> <p><b>Aircondition-Service 2</b></p> <p><input type="radio"/> Very poor (VP)</p> <p><input type="radio"/> Poor (P)</p> <p><input type="radio"/> Fair (F)</p> <p><input type="radio"/> Good (G)</p> <p><input type="radio"/> Very good (VG)</p>	<p><b>Humid</b></p> <p><b>Dehumidifier-Service</b></p> <p><input type="radio"/> Very poor (VP)</p> <p><input type="radio"/> Poor (P)</p> <p><input type="radio"/> Fair (F)</p> <p><input type="radio"/> Good (G)</p> <p><input type="radio"/> Very good (VG)</p>
--	---	--

Figure 11 GUI for user’s preference collection.

based on three QoS criteria, denoted by  $C_l, l= 1, 2,3$ . The hierarchical structure of these criteria is shown in Figure 14.

The details for these criteria are listed as follows: (1) **Brightness Related QoS** ( $C_1$ ). (2) **Temperature QoS** ( $C_2$ ). (3) **Humidity QoS**( $C_3$ ).

The fuzzy TOPSIS method is then applied to solve this problem according to the following procedures step 4-1 to step 4-7:

Step 4-1 The users use the linguistic variables to evaluate the importance of each criterion. Without using the linguistic variables, we applied the entropy method to obtain fuzzy weights of criteria [70] as follows.

$$w_1 = (0.090, 0.200, 0.208); w_2 = (0.129, 0.206, 0.216);$$

$$w_3 = (0.041, 0.141, 0.477);$$

Figure 12 RDF sample.

```
<RDF
  xmlns="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:n="http://www.nist.gov/units/">
  <rdf:Description about="userPreference">
    <rdf:Description aboutEach="#C01"
      pics:by="Jane Doe"
      pics:rating="G"
      pics:alternative="1"/>

    <rdf:Description aboutEach="#C02"
      pics:by="Jane Doe"
      pics:rating="VG"
      pics:alternative="1"/>
    ...
  </Description>
</RDF>
```

**Figure 13** SDO object sample.

```

DataGraph graph = db.executeQuery( select * from user_preference);
DataObject root = graph.getRootObject();
// use xpath to get the name of the first customer
String userName = root.getString(user_preference [0]/name );
// iterate through all customers
Iterator iter = root.getList( customer ).iterator();
while (iter.hasNext()) {
    DataObject dataObject = (DataObject) iter.next();
    String userName = dataObject.getString( name );
    Double userRating = dataObject.getString( Rating );
    ...
}
    
```

In this step, we use the above fuzzy weights of criteria and apply Eq. 2 to calculate the weights of criteria as follows.

$$w_1 = 0.183; w_2 = 0.1948; w_3 = 0.1803;$$

Step 4-2 For each web service, the users use the linguistic variables to produce fuzzy or crisp performance ratings against each criterion.

The graded mean integration representation of each linguistic variable is listed in Table 6.

Step 4-3 By applying Eq. 9, the aggregated ratings of web services with respect to the three criteria can be computed.

Step 4-4 Construct the normalized decision matrix. The normalized decision matrix can be calculated by applying Eq. 10 as follows.

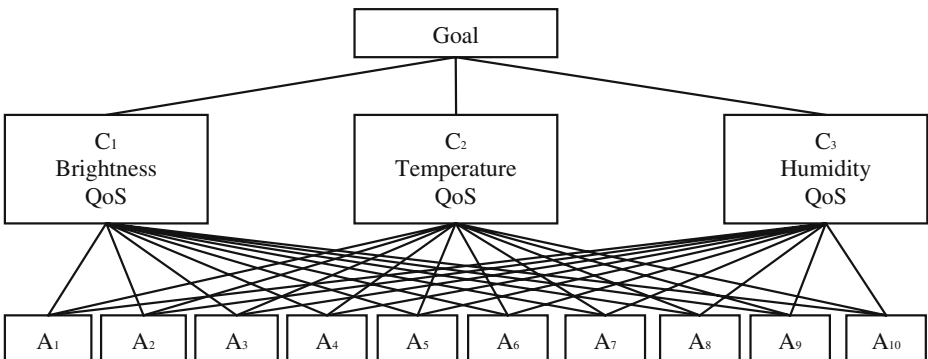
Step 4-5 Determine the positive ideal solution,  $A^+$ , and the negative ideal solution,  $A^-$  as follows.

$$A^+ = (0.4101, 0.4213, 0.4333) \quad A^- = (0.2839, 0.2917, 0.2500)$$

Step 4-6 Calculate the weighted Euclidean distance of each web service from  $A^+$  and  $A^-$  as follows (Table 7).

Step 4-7 Obtain the relative closeness coefficient and rank the order of web services.

$$\begin{aligned}
 &RCC_1 = 0.3000, RCC_2 = 0.4070, RCC_3 = 0.2694, RCC_4 = 0.2436, RCC_5 = 0.1286, \\
 &RCC_6 = 0.2755, RCC_7 = 0.3800, RCC_8 = 0.5830, RCC_9 = 0.6774, RCC_{10} = 1 \\
 &RCC_{10} \succ RCC_9 \succ RCC_8 \succ RCC_6 \succ RCC_7 \succ RCC_1 \succ RCC_6 \succ RCC_3 \succ RCC_5 \succ RCC_4
 \end{aligned}$$



**Figure 14** A hierarchical structure for the web service selection problem.

**Table 6** The normalized decision matrix.

Candidates	Criteria		
	$C_1$	$C_2$	$C_3$
$A_1$	0.3273	0.2917	0.2500
$A_2$	0.3470	0.2999	0.2749
...	...	...	...
$A_{10}$	0.4101	0.4213	0.4333

According to the above relative closeness coefficient, the ranking order of the ten alternative web services is  $A_{10}, A_9, A_8, A_6, \dots$ , and  $A_4$ . From Eqs. 20 and 21, our method is capable of revealing the positive and negative preference degree associated with Users’ alternative and assisting the users to make a decision based on the group consensus. We can use the alternative  $A_{10}$  to bind the web services that simulates the other scenario on a LED matrix on the Figure 15.

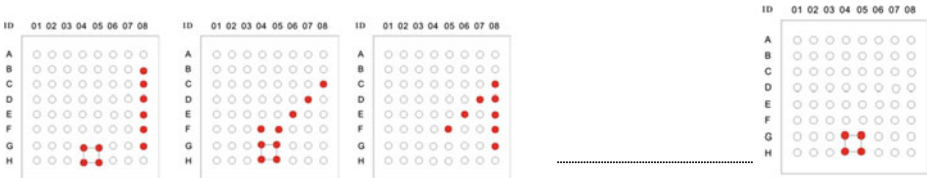
Step 5 The previous step determine which services will be selected. In this step, Synapse is introduced to bind the physical service. Synapse is a vehicle to enable services to be located and invoked. Synapse is driven by a set of XML statement in the configuration, so it is easy to deploy and invoke services. Part of XML for configuring light services is illustrated as follows (Figure 16).

Step 6 The corresponding web service would be executed after the previous step. The following XML code shows that LightingService SOAP message. Later, the SOAP message will be sent to our simulated circuit board. There are three main elements “Type”, “Duration”, and “Blank”. The first element “Type” means the output signal. The signal could be digital or analog. The second element “Duration” stands for the execution time. The third element “Blink” stands for the lighting twinkle (Figure 17).

We implement a XML parser to receive SOAP message and convert the XML to the command which can be carried out on Arduino. The simulation device can receive data from wired or wireless network. The LED matrix on the circuit board stands for different services, that is, we simulate these services in our system. We design and implement the hardware plugged with multiple input and output channels. It can also extend the storage by adding extra memory card such as SD-Card. A circuit board, which runs on 2.7–5.5 V and comprises of an ATmega168V microcontroller, a reset switch, an indicator LED. It also includes a 5 V power supply and an RGB LED Matrix. The hardware architecture of simulation device is shown in Figure 18.

**Table 7** The distance measure.

Candidates	The distance measure	
	$d^+$	$d^-$
$A_1$	0.089119	0.038192
$A_2$	0.076057	0.052212
...	...	...
$A_{10}$	0	0.164882



**Figure 15** Alternatives presents on a LED matrix.

Step 7 Users can revise their preference after they receive the binding results. On the other hand, if a new staff enters the office, he/she can provide his/her opinion to the system. Similarly, these feedbacks will be included in the system as inputs for the next round of calculating group consensus. If the alternative changes from one to another, the service binding would change accordingly (see Figure 19). So, the system configuration is dynamic based on the group’s preferences.

### 6 Discussion

In this previous section, the proposed user centric service-oriented modeling approach has been applied to a smart environment which includes a number of appliance services and 10 staffs in an open office. Since we designed a simulated environment using LED lights instead of control system to instruct appliances in order to demonstrate the system feasibility, the evaluation on service consumers’ subjective opinions and preferences on these composite services presents a difficulty. The adjustments on the appliances, however, were carried out manually according to predefined configurations and questionnaire was used to collect their feedbacks. In addition, we have evaluated the system effectiveness by examining whether the LED lights have behaved consistently according to the user requirements. In other words, composite services should be selected in line with the group consensus. We have varied the values in the preference database, so different composite services are selected in response to the changes. The experimental results given by the previous sections shows composite service 10 having been selected initially and LED lights have been switched on accordingly. The composite service 9 was chosen later, as the users have changed their preferences. This demonstrates the system coherence and SCA can be used to model software and hardware components as services.

The system performance and complexity are important criteria for evaluating the system. The system cannot be scaled up, if it required huge computational resources when the number of services and QoS criteria increases. We have evaluated TOPSIS method performance and complexity, as it is a critical task in the system. We also compare it with

**Figure 16** Sample XML of synapse configuration.

```

<definitions xmlns="http://ws.apache.org/ns/synapse">
  <!-- filtering of messages with XPath and regex matches -->
  <filter source="get-property('To') regex=".*\/LightingService.*" />
    <send>
      <endpoint>
        <address uri="http://localhost:9000/soap/LightingService" />
      </endpoint>
    </send>
    <drop />
  </filter>
</send />
</definitions>

```

```
<?xml version='1.0' encoding='UTF-8'?>
  <soapenv:Envelope xmlns:xenc="http://www.w3.org/2001/04/xmldcnc#"
  xmlns:wsa="http://www.w3.org/2005/08/addressing".>
    <soapenv:Header>
    </soapenv:Header>
    <soapenv:Body xmlns:wssu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.
    xsd">
      <Device ID="01">
        <Command Type="Digital">On</Command>
        <Command Duration="Loop">Yes</Command>
        <Command Blink="NO">0</Command>
      </Device>
    </soapenv:Body>
  </soapenv:Envelope>
```

Figure 17 Sample lightingservice of SOAP message.

other existing approaches to express the distinct features of the proposed approach. Another experiment also has been designed to examine system usability and service performance.

### 6.1 Evaluation

Opinion similarity measure is the most computational resource demanding step in the proposed approach. The opinions collected from the users are fuzzy terms. The traditional methods [26] would calculate the maximum and minimum intersection area of two membership functions given by the users. For example, the triangle membership function is composed of four piecewise linear segments. The first segment function given by the user 1 would check the point of intersection in respect to the four piecewise linear segments from the user 2. After that, the second segment function would check others until all the four segments have been done. Computing similarity measures and constructing an agreement matrix would be done in time  $O(m^n)$ . The crisp values can be derived from the graded mean integration representation

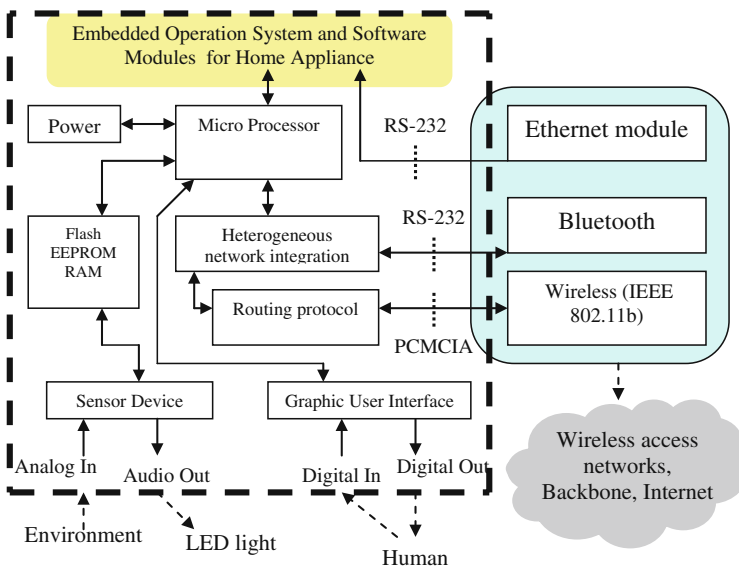
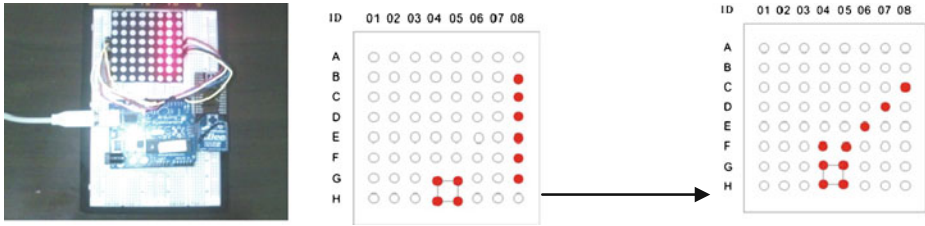


Figure 18 Implementation of microcontroller hardware.



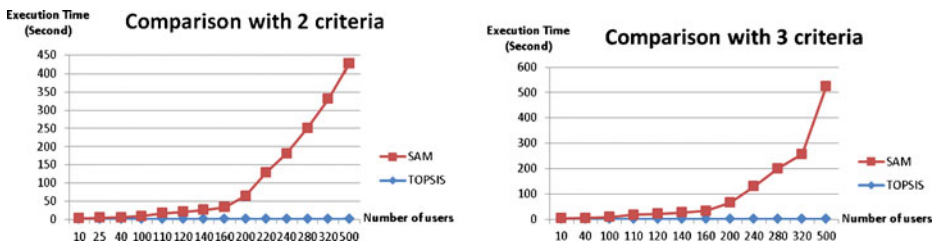
**Figure 19** Group users change alternative from A10 to A9.

method. So, the required computation on the crisp number significantly reduces the complexity compared with fuzzy values. It can be complete in time  $O(m \cdot n)$ .

According to the complexity of computing users' evaluation, we also used the TOPSIS method to eliminate the problem associated with the duplicated calculation on weightings by introducing the Minkowski distance function. This can increase the accuracy in measurement. Moreover, our approach can significantly reduce computation complexity in similarity measure, so the proposed approach can be scaled up. Figure 20 (left) shows the system performance of our proposed TOPSIS method against Huang's work-SAM with 2 criteria. Figure 20(right) shows the system performance with 3 criteria. The computational time of SAM would increase exponentially as the number of user opinions and criteria grow. On contrast, the computational time of TOPSIS method only increases slightly when the number of criteria rises to 16. The computational time does not exceed 10 s (see Figure 21). Overall, the TOPSIS is very efficient in the cases where the large number of users involves. Hence, the approach is suitable for on-line applications which often involve large amount of data.

From the above case study, we can get the ranking order of the three alternative web services is  $A_{10}, A_9, A_8, A_6, \dots$ , and  $A_4$ . The resulting order of these preferences is derived from 10 users. While the number of users increases significantly, the proposed approach still outperforms Huang's work [28].

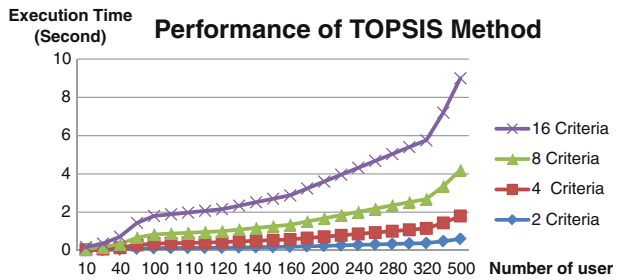
We setup the system within a research lab which is an open office in order to evaluate system usability and service performance. We designed an experiment in which 10 users had to rank the alternatives and choose them according to the predefined criteria. Three different types of services such as air-condition, dehumidifiers and lighting are included for service provision, but with 10 different combinations (10 composite services). Before the system is introduced to the users, the average satisfaction rate from 10 users was just under 43.75%. After the system reasoned over their opinions and preferences, a new service composition was selected to reflect most users' requirements by changing office appliance



**Figure 20** Performance analysis under difference users.



**Figure 21** Performance analysis of different criteria.



configuration. As a result, the average satisfaction rate increases to 68.75%. This evidences usability of the system.

The performance of TOPSIS is analyzed by varying numbers of criteria and users, so we can conclude that the method is very efficient when the number of users is under 320 and the number of criteria does not exceed 16 (see Figure 21). The number of services that can be supported or executed depends on the capacity of web container. We deploy all the services to one server in this experiment, as we only have 10 composite services. In this case, we have run the tests on an Apache Axis2 server with various numbers of services. The total volume of data relating to users’ preferences and opinions which has been used for reasoning is around 40K. According to the results in Table 8 we conclude that the system is scalable. When the number of services increases in two folds, it did not require twice execution time. This is due to Apache Axis2 server managing concurrent processes with great efficiency. In other words, the services only require linear resources, as the number grows. The total execution time of the whole platform is around 6000ms in the above experiment without taking into account of time for manual operations on appliances.

### 7 Conclusion

In this paper, we proposed a new user centric service-oriented modeling approach which is able to reason over a set of inconsistent fuzzy requirements into required services and provide an approach to model software and hardware as services to meet the dynamic requirements.

The proposed method is featured by integrating fuzzy TOPSIS method and SCA to facilitate web service development and to effectively satisfy a group of service consumers’ subjective requirements and preferences. The overall architecture including a number of modules and procedures provides a systematic approach for service developers to model different levels of hardware and software components as services, so they can be composed on the fly to meet changeable user requirements. In this study, the proposed model is generic and systematic, so system architecture and service abstract level APIs can be

**Table 8** Service execution time on axis2.

Number of services	Execution time
2 services	5187 ms
4 services	5357 ms
10 services	5398 ms
16 services	5448 ms

extended to accommodate new appliances and comfort criteria. New opinions and users can be included in the process of reaching a new group consensus by re-calculating them.

We carried out a number of experiments on a simulated environment which includes a 8\*8 LED matrix representing 30 different services to form 10 composite services for selection. These services can be invoked according to the group preference and the users' feedbacks. The experimental results show that the system can stratify users' requirements. It also shows the system has a reasonable performance, as it did not require exponential resources when the number of services increases. As a result, we can conclude that the proposed approach is effective and efficient.

The future work of this research will include different types of sensors such as environmental and biological sensors, so room temperature, humidity, and brightness etc can be collected automatically along with sensors to measure body temperature, heart beat and blood pressure etc in order to draw a complete picture of users' conform levels in relation to their environment. So, the system can intelligently and automatically interact with users and environment accordingly. The manual inputs required from users can be reduced.

**Acknowledgment** The research is supported by the National Science Council of Taiwan under the grant No. NSC 96-2416-H-009-008-MY3. This research is also partially supported by EU FP7 program under grant agreement no 224609. This research is also partially supported by CHEN-YUNG MEMORIAL FOUNDATION.

## References

1. Abo-Sinna, M.A., Amer, A.H., Ibrahim, A.S.: Extensions of TOPSIS for large scale multi-objective non-linear programming problems with block angular structure. *Appl. Math. Model.* **32**, 292–302 (2008)
2. Anton, N., Sergiy, N., Vagan, T., Jari, V.: Using UDDI for publishing metadata of the semantic web. *J. Syst. Softw.* **81**(3), 368–38 (2008)
3. Asuman, D., Ibrahim, C., Gokce, L., Yildiray, K.: Improving the functionality of UDDI registries through web service semantics. *Lect. Notes Comput. Sci.* **2444**, 9–18 (2002)
4. Bauer, B., Muller, J.P.: MDA applied: From sequence diagrams to web service choreography. In *ICWE'04 (LNCS 3140)*, pp. 132–136, (2004)
5. Bellwood, T.: IBM. Luc Clement, Microsoft, Claw von Riegen, SAP, UDDI Version 3.0.2, UDDI Spec Technical Committee Specification, Oct 19, (2004)
6. Belouadha, F.Z., Omrana, H., Roudiès O.: A model-driven approach for composing SAWSDL semantic Web services. *Int. J. Comput. Sci. Issues* **7**(2), No 1 (2010).
7. Bianchini, D., De Antonellis, V., Melchiori, M.: Flexible semantic-based service matchmaking and discovery. *World Wide Web* **11**(2), 227–251 (2008)
8. Bordogna, G., Fedrizzi, M., Pasi, G.: A linguistic modeling of consensus in group decision-making based on OWA operators. *IEEE Trans. Syst. Man Cybern., A: Syst. Humans* **27**, 126–132 (1997)
9. Bryson, N.: Group decision-making and the analytic hierarchy process: exploring the consensus-relevant information content. *Comput. Oper. Res.* **23**, 27–35 (1996)
10. Carlsson, C., Ehrenburg, D., Eklund, P., Fedrizzi, M., Gustafsson, P., Lindholm, P., Merkuryeva, G., Riissanen, T., Ventre, A.G.S.: Consensus in distributed soft environments. *Eur. J. Oper. Res.* **61**, 165–185 (1992)
11. Casati, F., Ilnicki, S., Jin L.: Adaptive and dynamic service composition in EFlow. In *Proceedings of 12th International Conference on Advanced Information Systems Engineering(CAiSE)*. Springer Verlag, Stockholm, Sweden, June (2000)
12. Casati, F., Sayal, M., Shan, M.C.: Developing e-services for composing eservices. In *Proceedings of 13th International Conference on Advanced Information Systems Engineering(CAiSE)*. Springer Verlag, Interlaken, Switzerland, June (2001)
13. Chamodrakas, I., Alexopoulou, N., Martakos, D.: Customer evaluation for order acceptance using a novel class of fuzzy methods based on TOPSIS. *Expert Syst. Appl.* **36**, 7409–7415 (2009)
14. Chen, T.C., Tsao, C.Y.: The interval-valued fuzzy TOPSIS method and experimental analysis. *Fuzzy Sets Syst.* **159**(11), 1410–1428 (2008)

15. Chen, W., Nguyen, S.T., Coops, R., Bambang Octomo, S., Feijs, L.M.G.: Wireless transmission design for health monitoring at neonatal intensive care units, submitted to the 2nd international symposium on applied sciences in biomedical and communication technologies (ISABEL 2009), Bratislava, Slovak Republic, Nov. (2009)
16. Chou, C.C.: The canonical representation of multiplication operation on triangular fuzzy numbers. *Comput. Math. Appl.* **45**, 1601–1610 (2003)
17. Chu, T.C., Lin, Y.C.: An interval arithmetic based fuzzy TOPSIS model. *Expert Syst. Appl.* **36**, 10870–10876 (2009)
18. Chu, Q., Shen, Y., Jiang Z., A Transaction Middleware Model for SCA Programming, First International Workshop on Education Technology and Computer Science, 2009.
19. Cibrán, M.A., Verheecke, B., Vanderperren, W., Suvée, D., Jonckers, V.: Aspect-oriented programming for dynamic web service selection, integration and management. *World Wide Web* **10**(3), 211–242 (2007)
20. Davis, D., Parashar, M.: Latency performance of SOAP implementations. In Proceedings of the 2nd IEEE/ACM International Symposium on Cluster Computing and the Grid, pp. 407–412, (2002)
21. Deng, H., Yeh, C.H., Willis, R.J.: Inter-company comparison using modified TOPSIS with objective weights. *Comput. Oper. Res.* **27**, 963–973 (2000)
22. Efstathiadsea, A., Tassoub, S., Antoniouc, A.: Strategic planning, transfer and implementation of advanced manufacturing technologies (AMT). *Technovation* **22**(4), 210–212 (2002)
23. Govindaraju, M., Slominski, A., Choppella, V., Bramley, R., Gannon, D.: Requirements for and evaluation of RMI protocols for scientific computing, ACM/IEEE conference on Supercomputing, (2000)
24. Herrera, F., Herrera-Viedma, E., Verdegray, J.L.: A model of consensus in group decision-making under linguistic assessments. *Fuzzy Sets Syst.* **78**, 73–87 (1996)
25. Holleis, P.: Programming interactive physical prototypes. Proc. 1st Int'l Workshop on Design and Integration Principles for Smart Objects (DIPSO 07), (2007)
26. Hsu, H.M., Chen, C.T.: Aggregation of fuzzy opinions under group decision making. *Fuzzy Sets Syst.* **79** (3), 279–285 (1996)
27. Huang, C.L., Lo, C.C., Wang, P., Chao, K.M., Li, Y.S.: Applying semi-order preference model in content-based service discovery. *Int. J. E Bus. Manage.* **5**(1), 48–58 (2007)
28. Hwang, C., Yoon, K.: Multiple attribute decision making methods and application. Springer, New York (1981)
29. Hwang, C.L., Yoon, K.: Multiple attribute decision making: methods and applications. Springer-verlag, New York (1981)
30. Jini Technology Core Platform Specification, v. 2.0, Sun Microsystems, June (2003)
31. Kacprzyk, J., Fedrizzi, M.: Multiperson decision-making models using fuzzy sets and possibility theory. Kluwer, Dordrecht (1993)
32. Kacprzyk, J., Fedrizzi, M., Nurmi, H.: Consensus under fuzziness. Kluwer, Norwell (1997)
33. Kahraman, C., Engin, O., Kabak, Ö., Kaya, İ.: Information systems outsourcing decisions using a group decision-making approach. *Eng. Appl. Artif. Intell.* **22**, 832–841 (2009)
34. Karsak, E.E.: Distance-based fuzzy MCDM approach for evaluating flexible manufacturing system alternatives. *Int. J. Prod. Res.* **40**(13), 3167–3181 (2002)
35. Kauffman, A., Gupta, M.M.: Introduction of fuzzy arithmetic: Theory and applications. Van Nostrand, New York (1985)
36. Konito, J.: A case study in applying a systematic method for COTS selection. In: Proc., 18th Int. Conf. on Soft. Eng. (ICSE). IEEE Computer Society Press, pp. 201–209, (1996)
37. Li, D.F.: Compromise ratio method for fuzzy multi-attribute group decision making, *Applied. Soft Comput.* **7**(3), 807–817 (2007)
38. Li, Y., Zou, F., Wu, Z., Ma, F.: PWSD: A scalable web service discovery architecture based on peer-to-peer overlay network. In Proc. APWeb04, LNCS 3007, pp. 291–300, (2004)
39. Lin, H.T., Chang, W.L.: Order selection and pricing methods using flexible quantity and fuzzy approach for buyer evaluation. *Eur. J. Oper. Res.* **187**, 415–428 (2008)
40. Lin, K.J., Zhang, J., Zhai, Y., Xu, B.: The design and implementation of service process reconfiguration with end-to-end QoS constraints in SOA. *Serv. Oriented Comput. Appl.* **4**(3), 157–168 (2010)
41. Lo, C.C., Cheng, D.Y., Lin, P.C., Chao, K.M.: A study on representation of QoS in UDDI for web services composition: Proceedings of the 2008 International Conference on Complex, Intelligent and Software Intensive Systems, pp. 423–428, (2008)
42. Luo, J., Montrose, B., Kang, M.: An approach for semantic query processing with UDDI. *Lect. Notes Comput. Sci.* **3762**, 89–98 (2005)
43. Luo, J.Z., Zhou, J.Y., Wu, Z.A.: An adaptive algorithm for QoS-aware service composition in grid environments. *Serv. Oriented Comput. Appl.* **3**(3), 217–226 (2010)

44. Malik, Z., Bouguettaya, A.: Rater credibility assessment in web services interactions. *World Wide Web* **12**(1), 3–25 (2009)
45. Medjahed, B., Bouguettaya, A., Elmagarmid, K.: Composing web services on the Semantic Web. *VLDB J.* **12**(4) (2003)
46. Mich, L., Gaio L., Fedrizzi M.: On fuzzy logic-based consensus in group decision, *Proceedings IFSA*, pp. 698–700. (1993)
47. Minkowski, H.: *Geometrie der Zahlen*. Teubner, Leipzig (1896)
48. Minkowski, H.: *Gesammelte Abhandlungen*. Teubner, Leipzig (1911)
49. Montebello, M., Abela, C.: DAML enabled web services and agents in the semantic web. *Revised Papers from the NODe 2002 Web and Database-Related Workshops on Web, Web-Services and Database Systems. Lecture Note on Computer Science*, pp. 46–58. (2003)
50. Moussa, H., Gao, T., Yen, I.L., Bastani, F., Jeng, J.J.: Toward effective service composition for real-time SOA-based systems. *Serv. Oriented Comput. Appl.* **4**(1), 17–31 (2010)
51. Narayanan, S., McIlraith, S.: Simulation, verification and automated composition of web service. In *Proceedings of the 11th International World Wide Web Conference, Honolulu, Hawaii, USA, May (2002)*
52. OASA, a set of Final V1.0 Specifications, Open SOA (OSOA), March (2007)
53. Ortiz, G., Hernández, J.: Aspect-oriented techniques for web services—a model-driven approach. *Int. J. Bus. Process Integration Manage.* **2**(2), 141–152 (2007)
54. Papazoglou, M.P., Georgakopoulos, D.: Service-oriented computing. *Communications ACM* **46**(10), 25–65 (2003)
55. Ribeiro, R.A.: Fuzzy multiple attribute decision making: a review and new preference elicitation techniques. *Fuzzy Sets Syst.* **78**, 155–181 (1996)
56. Rompothong, P., Senivongse, T.: A query federation of UDDI registries, ISICT, (2003)
57. Roy, G., Jaeger, M.C.: Model-driven semantic web service composition. *Proceedings of the 12th Asia-Pacific Software Engineering Conference table of contents*, pp.79–86. (2005)
58. Schlosser, M., Sintek, M., Decker, S., Nejdl, W.: A scalable and ontology-based P2P infrastructure for semantic web services. *Proc. 2nd Int. Conf. P2P'02, Sweden*, pp. 104–111. (2002)
59. Schmidt, C., Parashar, M.: A peer-to-peer approach to web service discovery. *World Wide Web* **7**(2), 211–229 (2004)
60. Schuster, H., Georgakopoulos, D., Cichocki, A., Baker, D.: Modeling and composing service-based and reference process-based multi-enterprise processes. In *Proceeding of 12th International Conference on Advanced Information Systems Engineering (CAiSE), Stockholm, Sweden. Springer Verlag, June (2000)*
61. Service Location Protocol, v. 2, IETF RFC 2608, June (1999)
62. Shih, H.S., Shyur, H.J., Lee, E.S.: An extension of TOPSIS for group decision making. *Math. Comput. Modell.* **45**, 801–813 (2007)
63. Simona, M., Lazăr, I., Pârv, B., Czibula, I.: An agile MDA approach for service-oriented components. *Electron. Notes Theor. Comput. Sci.* **253**(1), 95–110 (2009)
64. Sirin, E., Hendler, J., Parsia, B.: Semi-automatic composition of Web services using semantic descriptions. In *Proceedings of Web Services: Modeling, Architecture and Infrastructure workshop in conjunction with ICEIS2003, (2002)*
65. Sivashanmugam, K., Verma, K., Mulye, R., Zhong, Z., Sheth, A.: Speed-R: Semantic P2P environment for diverse web service registries, 2004. available at: <http://webster.cs.uga.edu/~mulye/SemEnt/Speed-R.html>
66. Sun, C.C., Lin, G.T.R.: Using fuzzy TOPSIS method for evaluating the competitive advantages of shopping websites. *Expert Syst. Appl.* **36**, 11764–11771 (2009)
67. UPnP Device Architecture 1.0, UPnP Forum, Dec. (2003)
68. Wang, T.C., Chang, T.H.: Application of TOPSIS in evaluating initial training aircraft under a fuzzy environment. *Expert Syst. Appl.* **33**, 870–880 (2007)
69. Wang, Y.J., Lee, H.S.: Generalizing TOPSIS for fuzzy multiple-criteria group decision-making. *Comput. Math. Appl.* **53**(11), 1762–1772 (2007)
70. Wang, P., Chao, K.M., Lo, C.C., Huang, C.L., Li, Y.: A fuzzy model for selection of QoS-aware web services. In *Proc. of IEEE ICEBE (2006)*
71. Wang, P., Chao, K.M., Lo, C.C.: On optimal decision for QoS-aware composite service selection. *International Journal of Expert Systems with Applications* **48**(3), Elsevier Science, (2010)
72. Yong, D.: Plant location selection based on fuzzy TOPSIS. *Int. J. Adv. Manuf. Technol.* **28**, 839–844 (2006)
73. Yoon, K., Hwang, C.L.: *Multiple Attribute Decision Making: An Introduction*. Sage, Thousand Oaks (1985)
74. Yu, Q., Rege, M., Bouguettaya, A., Medjahed, B., Ouzzani, M.: A two-phase framework for quality-aware web service selection. *Serv. Oriented Comput. Appl.* **4**(2), 63–79 (2010)

75. Zanakis, S.H., Solomon, A., Wishart, N., Dublisch, S.: Multi-attribute decision making: a simulation comparison of select methods. *Eur. J. Operat. Res.* **107**, 507–529 (1998)
76. Zhang, M., Cheng, Z., Zhaol, Y., Huang, J.Z., Li, Y., Zag, B.: ADDI: An agent-based extension to UDDI for supply chain management. *The 9th International Conference on Computer Supported Cooperative Work in Design Proceedings* **1**, 405–410, May 24–26, (2005)
77. Zhao, H., Doshi, P.: A hierarchical framework for logical composition of web services. *Serv. Oriented Comput. Appl.* **3**(4), 285–306 (2009)
78. Zhu, Z., Chen, Y., Lan, R., Li, Z.: Study of MDA based semantic web service composition. *Inf. Technol. J.* **8**(6) (2009).
79. Zimmermann, H.J.: *Fuzzy set theory and its applications*. Kluwer, Boston (1991)
80. Zou, Z., Duan, Z.: Building business processes or assembling service components: Reuse services with BPEL4WS and SCA. *Proceedings of the European Conference on Web Services (ECOWS'06)*, Zurich, Switzerland. Washington, DC: IEEE Computer Society, pp 138–147, Dec 4–6, (2006)