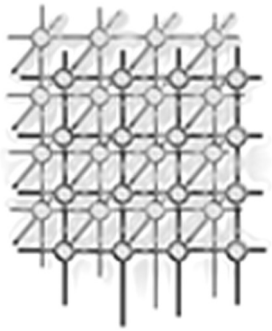
CONCURRENCY AND COMPUTATION: PRACTICE AND EXPERIENCE
Concurrency Computat.: Pract. Exper. 2007; **19**:1219–1236
Published online 3 April 2007 in Wiley InterScience (www.interscience.wiley.com). DOI: 10.1002/cpe.1157

QoS-aware mobile service transactions in a wireless environment



Muhammad Younas[1,*,†], Kuo-Ming Chao[2,3], Ping Wang[4] and Chun-Lung Huang[5]

[1]*Department of Computing, Oxford Brookes University, Oxford OX33 1HX, U.K.*
[2]*Software School, Fudan University, China*
[3]*Department of Computer and Network Systems, Coventry University, Coventry CV1 5FB, U.K.*
[4]*Department of MIS, Kun Shan University of Technology, Taiwan, ROC*
[5]*Institute of Information Management, National Chiao Tung University, Taiwan, ROC*

SUMMARY

Mobile computing greatly facilitates the provision of mobile services using portable and handheld devices which are connected through wireless networks. A pre-requisite for the provision of an effective service is to ensure quality of service (QoS) during the concurrent execution of requests and system or network failures. This paper presents a new transaction management approach for the provision of mobile services which is based on advanced transaction models and fuzzy selection criteria. It employs a novel QoS consensus moderation approach in order to select preferred mobile services from among various alternatives, thus providing multiple pathways to the completion of mobile service transactions. Unlike traditional transaction management schemes, the proposed approach takes into account QoS attributes in the commit process of mobile service transactions. The proposed approach is implemented as a prototype tool and evaluated through a number of experiments. Experimental results show that the proposed approach effectively selects mobile services and significantly increases the throughput (i.e. commit rate) of mobile service transactions. Copyright © 2007 John Wiley & Sons, Ltd.

Received 30 October 2006; Accepted 21 November 2006

KEY WORDS: mobile services; mobile transactions; Web services; quality of service; fuzzy logic

1. INTRODUCTION

Currently, an increasing number of people are using mobile devices (such as mobile phones, PDAs, etc.) in order to access information services in a timely and flexible manner.

*Correspondence to: M. Younas, Department of Computing, Oxford Brookes University, Oxford OX33 1HX, U.K.
‡E-mail: m.younas@brookes.ac.uk

Contract/grant sponsor: National Science Council of the Republic of China; contract/grant number: NSC 95-2416-H-168-007

Examples include obtaining stocks quotes, weather information, and news, and conducting financial transactions. The utilization of such information services is further facilitated through the Web services technology [1], which enables software applications to seamlessly interact and integrate using standard technologies such as XML, SOAP, WSDL, and UDDI. In the following discussion we use the term mobile services (as in [2,3]) to refer to Web services in a wireless environment. That is, Web services which are accessed through mobile devices are termed as mobile services. For instance, a hotel booking service (exposed as a Web service) can be accessed through a mobile device in order to make a hotel booking. Furthermore, such services can be composed with other services (such as flight booking) which can be collectively used by mobile users in order to acquire multiple services with a single request.

Although mobile computing greatly facilitates the provision of services it is associated with various QoS issues such as performance, throughput, reliability, consistency, security, and usability issues. In this paper we investigate the transaction management of mobile services, as this is crucial to maintaining the correct execution of services and the consistency of their underlying data. Transactions have been employed in various applications, such as e-business and mobile commerce applications [4,5]. The classical definition of a transaction is that it comprises a sequence of actions such that either all of them are completed or none. Classical transactions are based on the atomicity, consistency, isolation, durability (ACID) properties [4]. *Atomicity* enforces an 'all or nothing' policy such that a transaction must either appear to execute in its entirety or not at all. *Consistency* demands that a transaction must transform data from one consistent state to another consistent state. *Isolation* requires that an active transaction cannot expose its intermediate results to other transactions before its commitment. *Durability* means that the effects of the committed transactions must not be erased and should be made permanent in persistent storage for recovery purposes. In order to relax the ACID properties, various advanced (or extended) transaction models have been proposed in the literature, including open-nested transactions, flexible transactions, conversational transactions, and so on [4,6]. These models mainly relax atomicity and isolation properties. For example, atomicity is relaxed by allowing the partial commitment of a transaction and isolation is relaxed (or violated) by allowing transactions to see the intermediate results of other transactions.

Appropriate transaction management is important, specifically from the mobile user's perspective, as it should guarantee that services obtained are consistent with what users believe they have requested and what the system has delivered. From the service provider's perspective, it must ensure that transactions are correctly executed and that the information held in service provider data sources is consistently maintained. Existing approaches are limited to classical QoS such as response time, system and network efficiency, and the reliability of mobile transactions. They do not consider QoS attributes specific to mobile services such as execution price, usability, and so on [7]. In this paper we propose a new transaction management approach that takes into account classical as well as mobile service-specific QoS attributes.

Our approach is based on advanced transaction models and the QoS consensus moderation approach (QCMA) [7]. As described above, advanced transaction models relax ACID properties. Such models allow for the partial commitment of mobile transactions in contrast to ACID-based 'all or nothing' commitment. Thus they are more appropriate for mobile service transactions (MSTs) owing to the nature of wireless computing environments wherein long delays, failures, and disconnections are frequent. Traditional ACID properties may result in long delays and in useless consumption of resources (e.g. locking data). Furthermore, advanced transaction models allow for the

DOI: 10.1002/cpe

execution of alternative transactions. That is, if a component transaction of a MST fails to acquire a requested service, then an alternative will be executed in order to acquire an alternative service (if it exists). The use of alternative transactions is useful for two reasons: first, it provides users with a choice of various services; second, it increases the transaction throughput (i.e. commit rate) by preventing transaction failures. However, the selection of appropriate alternative services (for alternative transactions) is complicated by the availability of a large number of alternative services. For instance, a hotel booking service can have many alternatives provided by different service providers. Various elements contribute to the complexity of the selection of appropriate services: (i) the large number of alternative services provided by different service providers; (ii) the various QoS attributes such as execution price, efficiency, usability, security, and availability; (iii) the differences in the preferences and opinions of users as well as service providers. The proposed QCMA deals with these complexities and provides a method for selecting appropriate services. It assigns different QoS weights to services and ranks them accordingly. A MST can then use appropriate services in order to fulfill a user's requests. That is, it uses the highly ranked service first to perform the user request. If this service fails, then it uses the next highly ranked service, and so on. The aim of QCMA-based selection is twofold: first, it ensures the provision of a QoS-based service; second, it facilitates the selection of alternative services that are used to increase the throughout of a MST.

The proposed approach is based on our previous work of mobile transactions and Web services [7–9]. To the best of our knowledge, the existing research does not consider QoS aspects in mobile (service) transactions.

The remainder of this paper is organized as follows. Section 2 reviews current work and identifies related issues and Section 3 describes the basic concepts of fuzzy set theory. Section 4 presents the proposed approach and Section 5 presents the evaluation of the proposed approach. Section 6 concludes the paper.

2. RELATED WORK

MST management is complicated mainly due to the characteristics of mobile devices and wireless networks [8,10]. For instance, the mobility of processing units (e.g. laptop or PDA) means MSTs and data stores may physically re-locate during execution. A MST may originate at one site and terminate at another, and may require frequent re-connections in between. For example, a MST issued at cell 2 (C2) may terminate at cell 1 (C1) in a wireless cellular network as its originating device moves (see Figure 1). Furthermore, the bandwidth of wireless networks is currently low in comparison with wired networks. Thus, a wireless network may be overloaded by a high volume of information exchange. In addition, mobile devices are currently less reliable and have fewer resources than the conventional 'stationary' units. For example, the limited power supplies of mobile devices and the relatively limited resources affect throughput of mobile services, because they increase the probability of transaction failure (e.g. owing to a flat battery or running out of disk space).

In order to address issues that arise as a consequence of the characteristics of mobile devices and wireless networks, various approaches have been proposed in the literature. Alvarado *et al.* [11] present a self-adaptive component-based approach for mobile transactions. This approach does not depend on a single commit protocol. Instead, it implements different protocols (such as two-phase commit (2PC), presumed abort (PA), and presumed commit (PC)) in order to choose an appropriate protocol that suits

DOI: 10.1002/cpe

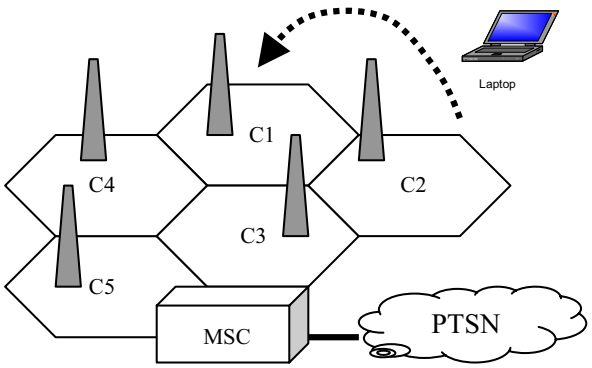


Figure 1. Generalized architecture of mobile computing.

the environment or context of a mobile transaction. For example, the 2PC and PA protocols can be used if the abort rate of mobile transactions is higher. Performance experiments show that the self-adaptive approach performs better than a (pre-defined) single protocol. Varshney and Ravikumar [12] propose a transaction and network-based approach for group-oriented mobile services. This approach identifies transaction requirements for such services and accordingly develops network protocols. It presents various simulation experiments in order to evaluate different types of transactions. The proposed protocols are shown to provide transactions with higher throughput (i.e. higher completion probability).

Kumar *et al.* [10] define a protocol, called TCOT, for mobile transactions. TCOT, a variant of the classical 2PC protocol, is based on a timeout mechanism and is claimed to have improved performance and throughput over the 2PC protocol in managing mobile transactions (the 2PC protocol enforces classical ACID criteria). However, 2PC and ACID criteria are inappropriate for MSTs. Lee and Helal [13] introduce a mobile transaction model, called high commit mobile transactions (HiCoMo), in order to improve the commitment rate of mobile transactions. Using HiCoMo transactions, aggregated data can be updated in situations where mobile units are disconnected.

The Kangaroo Transaction (KT) model [14] takes into account the movement behavior of mobile transactions so that they can hop from one base station to another as their mobile device moves. Such transaction hopping is modeled through the use of a split transaction model. In the KT model, a transaction splits only when it hops from one base station to another. Splitting is not allowed if transaction remains in the same cell. The KT model aims to reduce message overhead in the following way. When a mobile transaction moves to a new cell, the control of the transaction also moves to the new cell. However, if it were to remain at the originating cell messages would have to be sent from the originating cell to the current base station whenever the mobile unit requests information. To avoid this message overhead, the transaction management function should move with the mobile unit. However, if the originating device of a transaction moves back to the previous base station (or cell), then the information related to the transaction management needs to be sent back to the previous station. Thus, in certain situations, it may not help in reducing communication overhead.

DOI: 10.1002/cpe

A pre-serialization technique for managing transactions in a mobile multidatabase environment is proposed in [15]. This approach aims to maintain the isolation property of mobile transactions. The benefits claimed include supporting disconnection and maintaining transaction isolation. However, the main limitation of this approach is that these benefits are gained at the cost of processing overhead. The approach presented in [16] proposes a pre-write operation technique. The main objective of this approach is to increase data availability and reduce transaction processing on mobile devices. In this technique, a transaction is executed on the mobile device using operations such as pre-writes and pre-commit. The pre-write operation is performed on the local data items locked by the mobile device, and pre-commit is used to locally commit the transaction at the mobile device. After local commit, pre-writes of the data items are sent to the database, which makes them permanent and commits the mobile transaction.

Dunham and Kumar [17] study the impact of mobility on transaction management (IMTM). The IMTM study analyzes the performance of mobile transactions according to the mobility of different transaction management schemes. That is, the transaction management can: (i) be fixed on the mobile unit home site; (ii) be fixed on the anchor site of mobile unit; and (iii) move with the movement of mobile unit. This study shows that none of these management schemes always achieves a better performance. Younas *et al.* [8] proposes a pre-emptive resume scheduling mechanism that exploits the underlying network capabilities in order to improve the performance of mobile transactions.

The above approaches do not consider the provision of alternative services nor do they give attention to QoS-based service selection in mobile transaction management. Menasce [18] studies the QoS of component Web services in terms of cost and execution time. This study employs probability techniques to measure the cost and execution time of component Web services by considering different execution scenarios such as parallel, sequential, fastest-predecessor-triggered, and so on. This study helps in selecting appropriate component services for Web service composition. However, this research does not focus on MSTs.

3. PRELIMINARIES

This section illustrates the fundamental aspects of fuzzy set theory on which the proposed QCMA [7,9] is based.

Ordinary or crisp set theory enforces an all or nothing membership policy. Fuzzy set theory relaxes this policy and allows the inclusion of fuzzy (or vague) terms. These terms are associated with membership values which range between 0 and 1 (see [19]). Fuzzy set theory allows for partial membership wherein 0 represents non-membership and 1 represents complete membership of a term.

In many situations the classical set theory is inappropriate as it precludes the modeling of (alternative) services with marginal values. For instance, a service with a 20 ms response time can be considered as an efficient service (i.e. membership value = 1). However, a service with a slightly lower response time, say 19 ms, cannot be considered as efficient (i.e. membership value = 0). Thus, the latter cannot be considered as an alternative to the former. Fuzzy theory can be usefully applied to such a situation. A service with 19 ms response time can be represented as a nearly efficient service (i.e. membership value = 0.9), which can be considered as an alternative service in order to fulfill user requirements.

A fuzzy set can be defined in two ways: (i) by calculating membership values of those members in the set separately; or (ii) by defining membership function mathematically. In general, the former

DOI: 10.1002/cpe

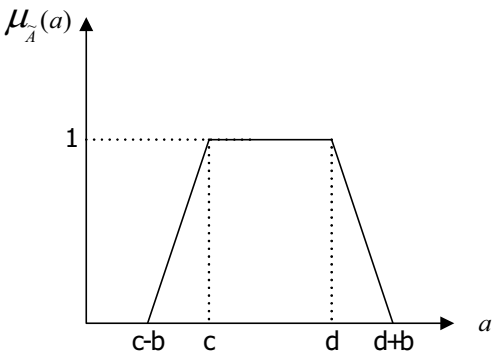


Figure 2. Trapezoidal membership function [6].

is used when the set is composed of discrete members and the latter is used when the domain is a continuous variable. For example, a fuzzy set $(\tilde{A})$ can be defined through enumeration using the following expression [19].

$$\tilde{A} = \sum \mu_A(x_i)/x_i = \mu_A(x_1)/x_1 + \mu_A(x_2)/x_2 + \mu_A(x_3)/x_3 + \mu_A(x_{...})/x_{...}$$

where the summation and addition operators refer to the union operation and the notation $\mu_A(x_i)/x_i$ refers to an element x_i with a membership degree $\mu_A(x_i)$. Those elements x_i whose membership value is zero are not represented.

A fuzzy set for continuous members is shown as follows (four common types for the membership function are trapezoidal, Gaussian, and S- and Z-membership functions):

$$\tilde{A} = \int_x \mu_A(x)/x$$

Using fuzzy set theory, a mobile user's opinion can be represented in a trapezoidal membership function that can be specified by four parameters:

$$\mu_{\tilde{A}}(a) = \begin{cases} \dfrac{x-(c-b)}{b}, & c-b \le a \le c \\ 1 & c \le a \le d \\ \dfrac{-x+(d+b)}{b}, & d \le a \le d+b \\ 0, & a > d+b, a < c-b \end{cases}$$

For example, Figure 2 shows a trapezoidal membership function for a fuzzy set $(\tilde{A})$ with parameters $(c-b, c, d, d+b)$.

DOI: 10.1002/cpe

4. THE PROPOSED APPROACH

This section presents the proposed approach which comprises three parts. Section 4.1 describes the transaction model of mobile services. Section 4.2 presents the QCMA method for selecting appropriate mobile services. Section 4.3 illustrates the execution protocol of MSTs.

4.1. MST model

The model comprises a set of mobile service providers that provide various services (e.g. hotel booking, stock trading, weather forecast, and banking). Each of the services is associated with underlying data representing information such as the type and price of a hotel room. A mobile service user may request a number of such services using a mobile device.

A MST is a sequence of different actions, where each action represents a component mobile transaction (CMT), which is executed in order to perform a requested task (or acquire a requested service). The sequence is terminated with commit or abort, where commit marks the successful execution and abort marks the unsuccessful execution of a MST. Each CMT has a type and comprises a sequence of operations, which concludes with either abort or commit. A CMT is compensatable if its effects can be semantically undone by executing a compensating service transaction. It is non-compensatable if its effects cannot be undone by executing a compensating service transaction. A CMT is replaceable if there is an associated alternative service transaction. A CMT is non-replaceable if there is no alternative service transaction associated with it. A CMT is vital if its abort results in the abort of a MST. It is non-vital if its abort does not result in the abort of MST. For example, a CMT of hotel booking can be considered as vital, while a CMT of car rental can be considered as non-vital if the user specifies the hotel booking as mandatory and car rental as optional services.

In the proposed approach, a MST is characterized by the following properties that are based on advanced transaction models [4,6,20].

Relaxed atomicity

This is weaker (more relaxed) than the classical atomicity. Whereas classical atomicity requires a MST to complete successfully or not at all, relaxed atomicity allows the partial commitment of such transaction. That is, a MST can still be committed if some of its CMTs are committed. This type of atomicity also allows the unilateral commit of CMTs irrespective of the commitment of their sibling CMT, with the constraint that information sources must remain consistent after the execution of such a transaction. Such unilateral commitment is useful for mobile services owing to their complex nature and longer execution times. Relaxed atomicity also allows for executing alternative CMTs. As described earlier, there exist various alternative services that can be executed in case of failures. For instance, if a room cannot be booked in one hotel then it can be booked in an alterative hotel provided that the alternative room is acceptable to the user.

Consistency

By enforcing relaxed atomicity intermediate states of CMTs become accessible. Thus, *consistency* can be enforced only at the component service level. For instance, if a CMT books a hotel then it will

DOI: 10.1002/cpe

release the resources (e.g. locks) of the hotel data source so that other transactions can make flight bookings. The traditional notion of consistency and isolation of ACID criteria [4] cannot be enforced in mobile services as they either result in the useless consumption of resources or resource blocking. That is, they require all of the CMTs to wait for the completion of each other in the prepare-to-commit state.

Durability

As in the ACID criteria, this requires that effects of a committed MST must be made permanent in the respective data sources in order to ensure failure recovery.

4.2. Mobile service selection

The process of selecting appropriate mobile services for a MST is a non-trivial task. As described earlier, it involves various factors including the availability of a large number of alternative services, the choice of QoS attributes, and the distinct preferences and opinions of users. For instance, a service with a 20 ms response time and 100 pence execution price can be considered as efficient and cheap by some users while others may consider this service as inefficient and expensive. It is therefore important to have a classification of alternative services for selection according to their overall quality rating. A ranked list of alternative services could facilitate users to effectively select alternatives services. The overall quality rating (or QoS) of a specific service is a view collected from a group of mobile users who use the service. In order to deal with the differences in the opinions and preferences of mobile users, we employ QCMA [7,9,21] which enables a group of users to reach a consensus on the QoS rating of a specific service. Such group consensus on the QoS rating is a basis to determine service ranking in the selection process.

Recall that there exist various QoS attributes for mobile services such as price, reliability, performance, integrity, adaptability, and bandwidth. It is therefore difficult to achieve a consistent view of a service QoS among different mobile users. Moreover, it is inappropriate to set a strict threshold to classify alternative services. Thus, in the proposed approach, the opinions of mobile users are represented via fuzzy logic. In the following we describe the service ranking process using QCMA (see [7,9] for further details on QCMA). QCMA, based on the fuzzy set theory [22,23], takes into account mobile users' opinions on a set of QoS attributes in order to achieve an overall quality rating for a specific service.

QCMA comprises two sub-processes: the similarity aggregation method (SAM) and the resolution method for group decision problems (RMGDP). The group consensus on a specific QoS attribute is gathered by SAM sub-process and the weightings over different QoS attributes are conducted by RMGDP sub-process (see [9,24,25] for details on SAM and RMGDP sub-processes). SAM aggregates different mobile users' fuzzy opinions to form a group's fuzzy consensus opinion on a specific QoS attribute. Each user represents their subjective fuzzy preference on one specific QoS attribute with a positive trapezoidal fuzzy number. SAM employs a similarity measure to calculate the differences between individuals within the group in order to obtain an index of consensus. The indexes of consensus for all pairs of individuals can be used to form an agreed group fuzzy opinion. SAM ensures the consistency of the definitions of fuzzy terms for providers and consumers [19].

DOI: 10.1002/cpe

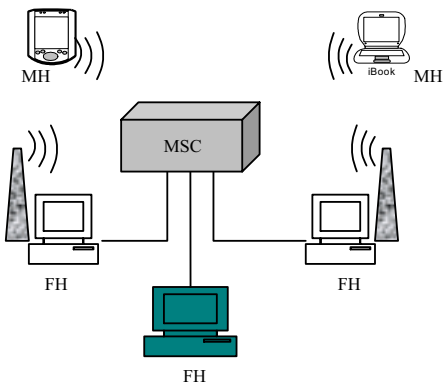


Figure 3. Underlying architecture of the proposed system.

The objective of RMGDP is to resolve group differences and to reach a group consensus on the weightings over different QoS attributes. This sub-process can be divided into three phases: (1) the transformation phase—transform the individuals' opinions on different QoS attributes into preference values; (2) the aggregation phase—aggregate the individual preference values for obtaining the group preference using an ordered weighted averaging (OWA) operator [26]; and (3) the exploitation phase—compute the weightings of QoS attributes by group preference [19].

The group consensus on QoS attributes are updated periodically by the QoS fuzzy moderator in line with the interactions taking place among mobile users. The QoS and their moderated opinions are represented in OWL [27] as the ontology. After the QCMA process, mobile users' opinions on a set of QoS attributes will be aggregated into a group consensus of opinion, which is represented in the form of a fuzzy trapezoidal membership function and is used to evaluate the overall quality rating for each alternative service. That is, each alternative service will be rated by a fuzzy value. According to the fuzzy values, alternative services can be ranked and listed in ascending order for the selection of alternative services. An alternative service with higher rank indicates a higher possibility for substitution which also improves the throughput in the case of transaction failures.

4.3. The execution protocol

The protocol is implemented as one *coordinator* and several *component service coordinators* (CSCs). The coordinator is deployed at a fixed host (FH) where CSCs can be deployed at fixed as well as mobile hosts (MHs) (as shown Figure 3). The selection process is deployed at the FH. The coordinator and each CSC maintain log files in order to record the required information (such as start, commit, and abort) about the execution of a MST and its component MST.

The first step of the protocol is to discover and select the required mobile services (using the selection process). When the required component services are identified, the system starts executing the MST. Each MST is associated with one coordinator and several CSCs that coordinate the execution of the CMTs. The coordinator and CSCs communicate with each other during the execution of the MST.

DOI: 10.1002/cpe

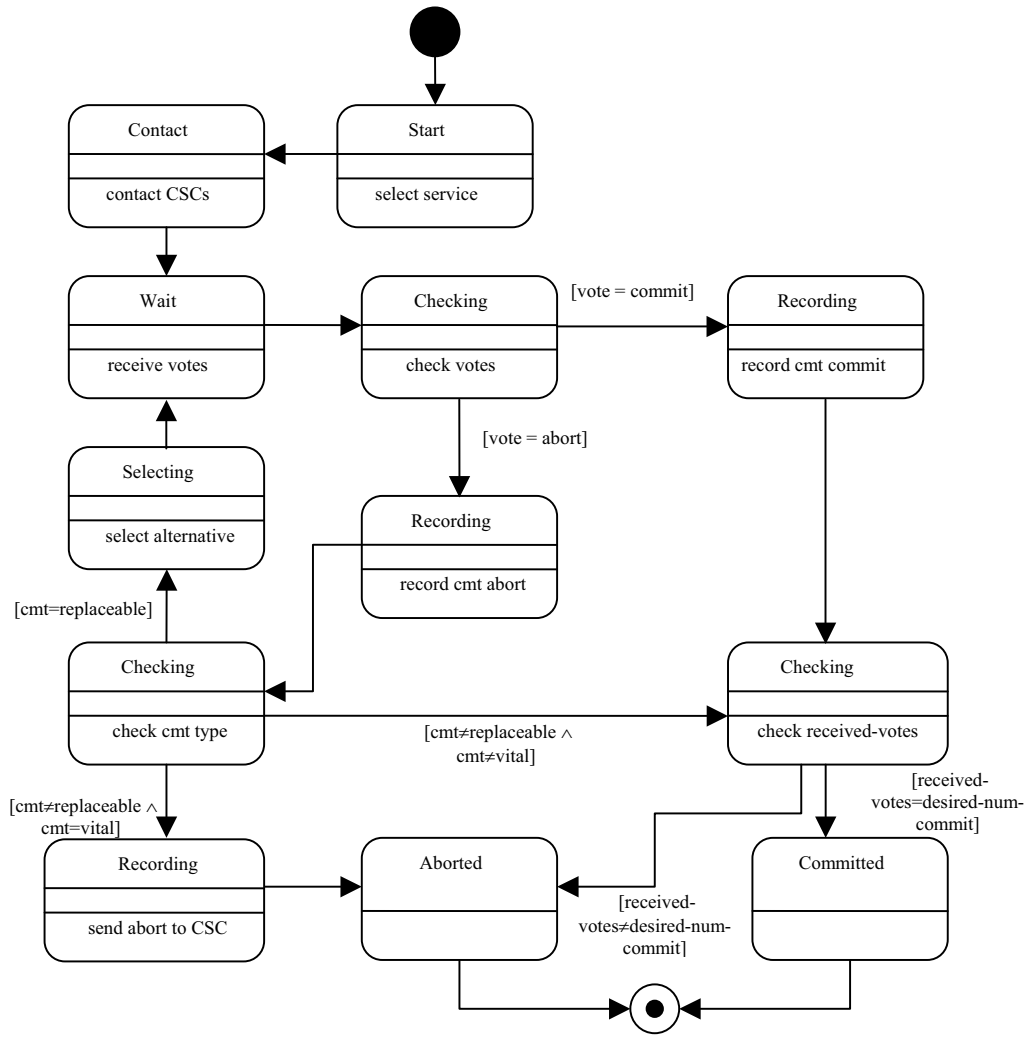


Figure 4. Coordinator state diagram.

We will describe two algorithms that illustrate the processing of our protocol. For the sake of simplicity, the coordinator is shown to communicate with one CSC. Algorithms 1 and 2 respectively describe the working mechanisms of the coordinator and a CSC. These algorithms are diagrammatically represented in Figures 4 and 5, respectively, using UML state diagrams.

DOI: 10.1002/cpe

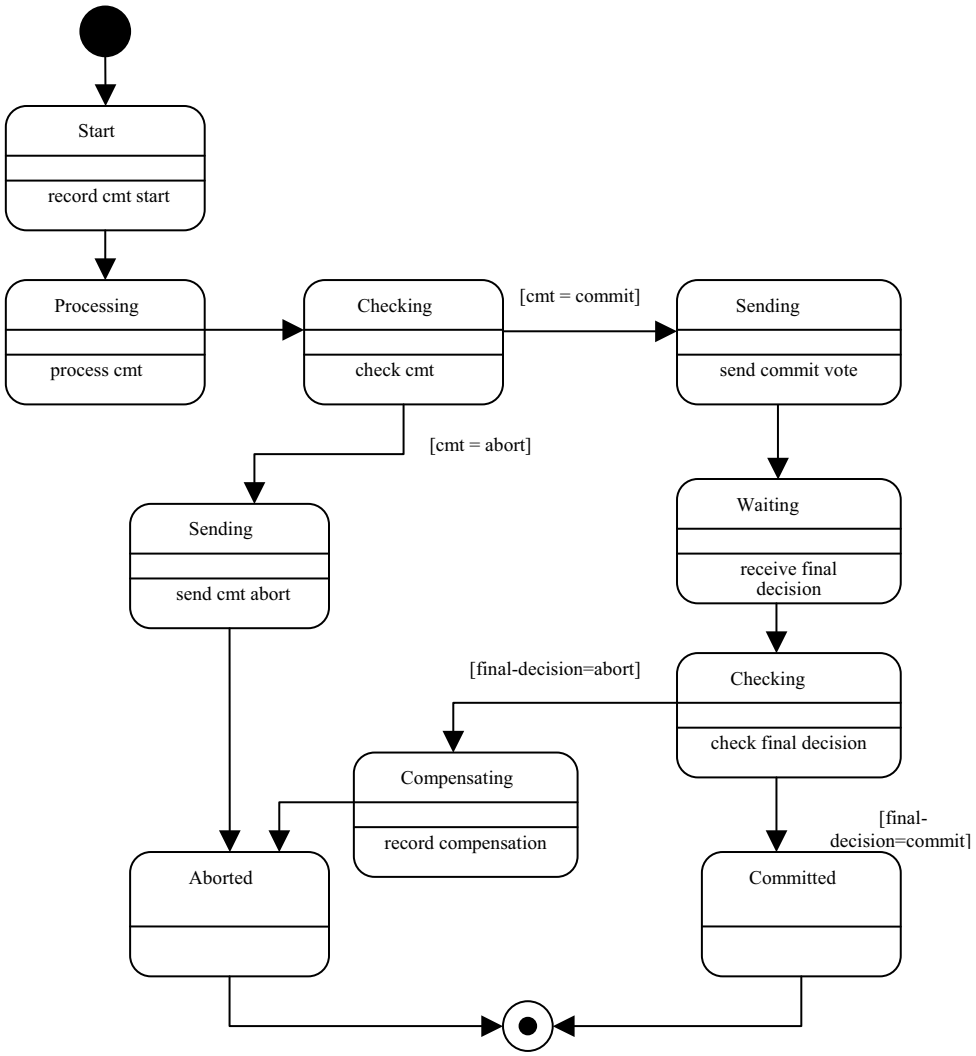


Figure 5. State diagram for component service coordinator.

DOI: 10.1002/cpe

Algorithm 1: Coordinator

1. receive request for a new MST
2. record start(MST) in a log-file
3. select appropriate service //using selection process
4. contact CSC_i regarding $CMT_i \in$ MST
5. wait to receive votes from all CSCs
6. receive $vote_i$ from CSC_i
7. check the status of votes received:
   - if $vote_i$ = commit
     - record commit(CMT_i) in log file
   - else if $vote_i$ = abort {
     - record abort(CMT_i) in log file
     - if CMT_i = replaceable {
       - check set of alternative CMT
       - repeat steps 3–7
     - }
     - else if $CMT_i \neq$ replaceable $\wedge$ $CMT_i \neq$ vital
       - check received-votes to decide on MST abort
       - else if $CMT_i \neq$ replaceable $\wedge$ CMT_i = vital {
         - record abort(MST) in log file
         - send abort to all CSCs
     - }
   - }
   - if received-votes = desired-num-commit{
     - record commit(MST) in log file
     - send final-commit to all CSCs
   - else if received-votes $\neq$ desired-num-commit{
     - record abort (MST) in log file
     - send final-abort to all CSCs
   - }
8. terminate MST

Algorithm 1. A new MST is assigned to a coordinator (step 1). It records the start of the MST in a log file (step 2), selects appropriate (highly ranked) service (step 3), and contacts the CSC to initiate CMTs (step 4). The coordinator then enters into a *wait* state, awaiting messages from CSCs concerning completion of the CMTs (step 5). The coordinator receives votes from the CSCs (step 6) and checks whether these votes represent a commit or an abort of the CMTs (step 7). If a vote received is a commit, it records this in a log file and proceeds to the next. If a vote received is an abort then it checks whether there is alternative service (i.e. replaceable) associated with the aborted CMT. If there is an alternative, then it selects the appropriate alternative service among the ranked services (using the selection process). The coordinator then communicates with the respective CSC regarding the commit or abort of the alternative CMT (repeating steps 4–7). If the aborted CMT does not have an alternative, then it takes one of two actions. If the CMT is defined as vital (by the user), then the coordinator records the abort of the MST and sends abort messages to the CSCs (if they have voted to commit). However, if the aborted CMT does not have an alternative transaction and is non-vital, then the coordinator ignores such aborts as the MST can still be committed if the desired number of other CMTs are committed.

DOI: 10.1002/cpe

Algorithm 2: Component Service Coordinator

1. receive request for a new $CMT_i \in MST$
2. record start(CMT_i) in a log-file
3. process CMT_i
4. check commit/abort of CMT_i
   if CMT_i = commit {
      record commit (CMT_i)
      send commit-vote$_i$ to coordinator
      wait for final-decision (from coordinator)
   }
   else if CMT_i = abort {
      record abort (CMT_i)
      send abort-vote$_i$ to coordinator
   }
5. receive final-decision
   if final-decision = commit
      record final-commit (CMT_i)
      record termination of CMT_i
   else if final-decision = abort
      compensate CMT_i
      record compensation of CMT_i
6. record termination of CMT_i

Algorithm 2. The CSC receives a request for a new CMT (step 1). It records the start of the CMT in a log file (step 2). After processing the CMT (step 3) the CSC checks whether a CMT is committed or aborted (step 4). If the CMT is committed, the CSC records the commit of the CMT and sends a commit vote to the coordinator. It then waits for the final decision regarding the commit of the CMT from the coordinator, i.e. whether the CMT will be committed or compensated. This ultimately depends on the commit of other CMTs. However, if the CMT is aborted, then the CSC records the abort of the CMT and sends an abort vote to the coordinator. It then marks the termination of the CMT. When the CSC receives the final decision from the coordinator (step 5) it acts as follows. If the final decision is to commit, then it records the commit of the CMT and marks the termination of the CMT. However, if the final decision is to abort, then it executes the compensating transaction to compensate for the effects of the committed CMT in order to ensure the data consistency of the underlying data source. After compensation, it marks the termination of the CMT (step 6).

5. EVALUATION

This section describes the evaluation of the proposed approach. The evaluation is performed in two stages. First, we evaluate the proposed selection process for selecting mobile services. Based on the selection process, we then evaluate the throughput (i.e. commit rate) of the proposed approach.

5.1. Mobile service selection

We evaluate the selection process using a case study of a hotel booking service. We use three hotel services, which are represented as HS1, HS2, and HS3. The selection process takes into account five

DOI: 10.1002/cpe

QoS attributes (or criteria) in order to evaluate the hotel services. Note that the QCMA is not limited to these attributes. It is capable of accommodating other QoS attributes.

These attributes include service reliability (a_1), usability (a_2), price/cost (a_3), availability (a_4), and efficiency (a_5). A service is considered to be reliable if it does not encounter failures for a specific time interval. Usability is the ease or simplicity of use of a service as experienced by a mobile user. The service price is the cost users have to pay to use the service per transaction. The availability of a service is defined as its availability for a specific time interval. The efficiency is defined as the response time of a service.

The procedure for evaluating service QoS is described as follows.

Each mobile user ($User_k$) has to evaluate the above list of attributes, $S_{\text{QoS}} = \{a_1, a_2, a_3, a_4, a_5\}$, and then assign an ordering preference to the alternatives HS1, HS2, and HS3. User preferences are represented by a 'preference relation' [25] which can be obtained directly or can be transformed from a 'preference ordering' [28]. For example, p_{ij}^k characterizes the preference degree between criteria a_i and a_j expressed by $User_k$. The value of p_{ij}^{c} is the collective preference which is an aggregation of k users' preferences, $\{p_{ij}^1, \ldots, p_{ij}^k\}$, by means of a fuzzy majority [29]. The final collective preference, p_{ij}^{c}, of all user preferences in this experiment is given by the following matrix:

$$p_{ij}^{\text{c}} = \begin{bmatrix} 0.5000 & 0.6500 & 0.3500 & 0.3625 & 0.5125 \\ 0.3500 & 0.5000 & 0.2000 & 0.2125 & 0.3625 \\ 0.6500 & 0.8000 & 0.5000 & 0.5125 & 0.6625 \\ 0.6375 & 0.7875 & 0.4875 & 0.5000 & 0.6500 \\ 0.4875 & 0.6375 & 0.3375 & 0.3500 & 0.5000 \end{bmatrix}$$

As described above, two fuzzy ranking methods are used to identify the priorities of attributes from collective preference, p_{ij}^{c}: the quantifier guided non-dominance degree (QGNDD) and the quantifier guided dominance degree (QGDD) [30]. QGDD can quantify the ordering preference dominance that a_i has over all of the other criteria using the fuzzy majority concept and QGNDD specifies the degree to which a_i is not dominated by a fuzzy majority of the remaining criteria [28]. The aim of the procedures is to identify the importance of these criteria and to calculate their weightings according to the users' preferences.

Figure 6 shows the result of the importance of QoS attributes obtained from QGDD and QGNDD. The value of wa_i $(i = 1 \ldots 5)$ represents the weighting of attribute a_i. The importance of the criteria can be concluded as

$$a_3 \text{ (service price)} > a_4 \text{ (availability)} > a_1 \text{ (reliability)} > a_5 \text{ (efficiency)} > a_2 \text{ (usability)}$$

The equation for evaluating QoS attributes with different weightings can be represented as

$$\text{QoS} = 0.2625a_3 + 0.2562a_4 + 0.1875a_1 + 0.1813a_5 + 0.1125a_2$$

After the derived QoS formula has been obtained, the evaluation process of the hotel services has to be carried out again by inviting a group of users to assess these weightings. The average score of a number of users' feedback is shown in Table I.

The consensus weights for attributes from the QGDD are then multiplied by the average score to obtain the final score for three alternatives:

$$\text{QoS score} = [0.2226 \text{ (HS1)} \quad 0.2281 \text{ (HS2)} \quad 0.2004 \text{ (HS3)}]$$

Thus, the complete preference order of the alternative services is HS2 > HS1 > HS3. The MST first uses HS2 to process user's requests. If HS2 fails it will then use HS1 and if HS1 fails it then uses HS3.

DOI: 10.1002/cpe

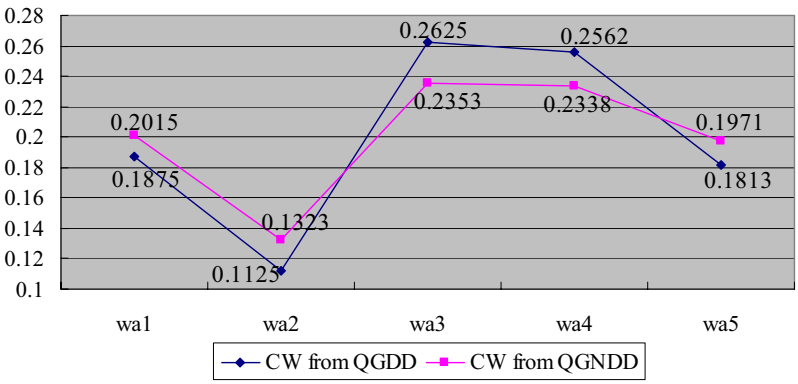


Figure 6. The QoS attributes obtained from the consensus weights from QGDD and QGNDD.

Table I. Weightings of services according to QoS attributes.

	QoS attributes				
Hotel service	a_3	a_4	a_1	a_5	a_2
HS1	0.4578	0.2648	0.1278	0.0412	0.0284
HS2	0.3830	0.2642	0.1648	0.1140	0.0740
HS3	0.1428	0.1800	0.4010	0.1536	0.1226

5.2. MST throughput

Throughput is defined as the number of MSTs that are committed (successfully completed). We simulate simple MSTs that book hotel rooms and accordingly make payment for the rooms, i.e. a MST comprises one CMT for hotel booking (and two alternatives in the case of failures) and another CMT for making payment for the hotel room. The MST first uses the most preferred mobile service, HS2, for hotel booking. However, if it fails then its alternative service is executed to book a room at the next preferred hotel (i.e. HS1). If this alternative also fails then the next alternative is executed to book a room at the third hotel (i.e. HS3). In the experiments, we assume that the money transfer CMT does not encounter failures. However, should such a failure occur then the transaction will be aborted.

The commit of a MST is simulated as a commit probability, which represents a probability that a MST will successfully complete. Failure is simulated as an abort probability. We conducted a range of experiments with varying degrees of failure, and these failures were simulated using different probabilities with probability values ranging from 0 (no failures) to 1 (complete failures). The outputs of these experiments are shown in Figures 7 and 8. Figure 7 represents the commit scenario of a MST. It shows that our approach increases the throughput (commit) of a MST when compared with

DOI: 10.1002/cpe

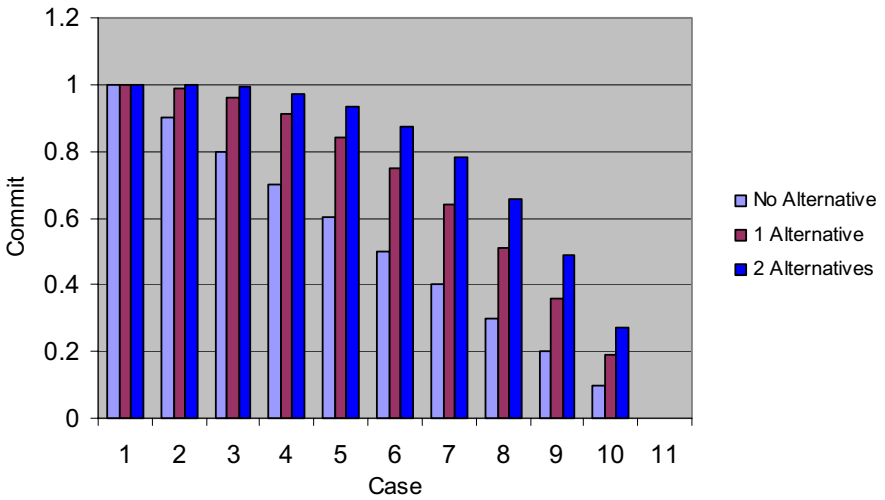


Figure 7. Commit scenario of a MST.

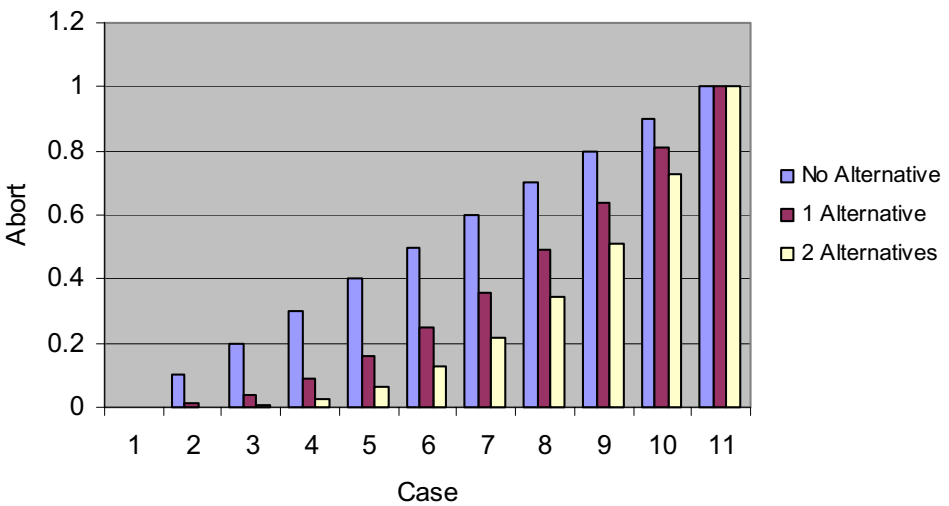


Figure 8. Abort scenario of a MST.

DOI: 10.1002/cpe

other approaches that do not consider alternative transactions. Figure 8 represents the abort scenario of a MST. Again this figure demonstrates that our approach significantly reduces the abort chances of a MST.

6. CONCLUSION

In this paper we have presented a new approach for MSTs. Unlike existing approaches, we have considered classical as well as mobile service-specific QoS attributes. Considering a limited number of classical QoS attributes is inappropriate given the availability of a large number of alternative mobile services. Such QoS attributes are important, specifically from the mobile user's perspective, as they demand services that fulfill their QoS needs. The development of a group consensus approach reduced subjective opinions given by mobile users in the selection of alternatives.

Our approach provides users with services that meet their QoS demands and also increases the throughput (i.e. the number of committed transactions) of their transactions in the case of failures. The experiments show the average successful rate with one and two alternatives is higher than the case with no alternative in the commit scenario of a MST. Accordingly, the experiments show that the abort rate is also minimized. However, the use of alternative service transactions increases the throughput as they affect the overall performance of a MST. However, evaluation of the overall performance is out of the scope of this paper and will be the subject of future research work.

ACKNOWLEDGEMENT

This research is partially supported by grant No. NSC 95-2416-H-168-007 from the National Science Council of the Republic of China.

REFERENCES

1. Pilioura T, Tsalgatidou A, Hadjiefthimiades S. Scenarios of using Web services in m-commerce. *ACM SIGecom Exchanges* 2003; **3**(4):28–36.
2. Maamar Z, Sheng Q, Benatallah B. Selection of Web services for composition using location of provider hosts criterion. *Proceedings of the 2003 Ubiquitous Mobile Information and Collaboration Systems Workshop (UMICS 2003)*, Austria, June 2003.
3. Maamar Z, Sheng QZ, Benatallah B. On composite Web services provisioning in an environment of fixed and mobile computing resources. *Information Technology and Management* 2004; **5**(3–4):251–270.
4. Elmagarmid AK. *Database Transaction Models for Advanced Applications*. Morgan Kaufman: San Francisco, CA, 1992.
5. Kuramitsu K, Sakamura K. Towards ubiquitous database in mobile commerce. *Proceedings of the 2nd ACM International Workshop on Data Engineering for Wireless and Mobile Access*, CA, May 2001. ACM Press: New York, 2001.
6. Younas M, Eaglestone B, Chao K. A low latency resilient protocol for e-business transactions. *International Journal of Web Engineering and Technology* 2004; **1**(3):278–296.
7. Wang P, Chao K-M, Lo C-C, Huang C-L, Younas M. A fuzzy outranking approach in risk analysis of Web service security. *Cluster Computing: The Journal of Networks, Software Tools and Applications*, to appear.
8. Younas M, Awan I, Chao K. Network-centric strategy for mobile transactions. *Journal of Interconnection Networks* 2004; **5**(3):329–350.
9. Huang C, Lo C, Chao K, Younas M. A moderated fuzzy Web services discovery method. *Journal of Information and Software Technology* 2006; **48**(6):410–423.
10. Kumar V, Prabhu N, Dunham M, Seydim YA. TCOT—a timeout-based mobile transaction commitment protocol. *IEEE Transactions on Computers* 2002; **5**(1):1212–1218.

DOI: 10.1002/cpe

11. Alvarado P, Rouvoy R, Merle P. Self adaptive component based transaction commit management. *Proceedings of the 4th International Workshop on Adaptive and Reflective Middleware (ARM)*, Grenoble, France, November 2005. ACM Press: New York, 2005.
12. Varshney U, Ravikumar K. Transaction and networking support for group oriented mobile commerce services. *Proceedings of the 1st Conference on Broadband Wireless Networks (BroadWISE)*, San Jose, CA, October 2004.
13. Lee M, Helal S. HiCoMo: High commit mobile transactions. *Distributed and Parallel Databases* 2002; **11**(1):73–92.
14. Dunham MH, Helal A, Balakrishnan S. A mobile transaction model that captures both the data and movement behavior. *ACM/Baltzer Journal on Special Topics in Mobile Networks and Applications* 1997; **2**(2):149–162.
15. Dirckze RA, Gruenwald L. A pre-serialization transaction management technique for mobile multidatabases. *Mobile Networks and Applications* 2000; **5**(4):111–321.
16. Madria S, Bhargava B. A transaction model to improve data availability in mobile computing. *Distributed and Parallel Databases* 2001; **10**(2):127–160.
17. Dunham M, Kumar V. Impact of mobility on transaction management. *Proceedings of the ACM International Workshop on Data Engineering for Wireless and Mobile Access*, Seattle, WA, 1999. ACM Press: New York, 1999.
18. Menasce D. Composing Web services: A QoS view. *IEEE Internet Computing* 2004; **8**(6):88–90.
19. Huang C-L. An approach to consensus-based service discovery. *PhD Dissertation*, National Chiao Tung University, Taiwan, September 2006.
20. Younas M, Eaglestone M, Holton R. A formal treatment of a SACReD protocol for multidatabase Web transactions. *Proceedings of the 11th International Conference on Database and Expert Applications*, London, 2000 (*Lecture Notes in Computer Science*, vol. 1873). Springer: Berlin, 2000.
21. Huang C, Chao K, Lo C, Wang P, Chung J. Service discovery through consensus-based preferences. *International Journal of Computer Systems Science and Engineering* 2006; **21**(4):255–262.
22. Zimmermann H. *Fuzzy Set Theory and its Applications*. Kluwer: Dordrecht, 1991.
23. Yen J, Langari R. *Fuzzy Logic: Intelligence, Control, and Information*. Prentice-Hall: Englewood Cliffs, NJ, 1999.
24. Hsu H, Chen C. Aggregation of fuzzy opinions under group decision making. *Fuzzy Sets and Systems* 1996; **79**:279–285.
25. Chiclana F, Herrera F, Herrera-Viedma E. Integrating multiplicative preference relations in a multipurpose decision-making model based on fuzzy preference relations. *Fuzzy Sets and Systems* 2001; **122**(2):277–291.
26. Yager R. On ordered weighted averaging aggregation operators in multi criteria decision making. *IEEE Transactions on Systems, Man, and Cybernetics* 1988; **18**:183–190.
27. Smith M, Welty C, McGuinness D. OWL Web Ontology Language Guide. *W3C Recommendation*, February 2004. Available at: http://www.w3.org/TR/2004/REC-owl-guide-20040210/.
28. Chiclana F, Herrera F, Herrera-Viedma E. Integrating three representation models in fuzzy multipurpose decision making based on fuzzy preference relations. *Fuzzy Sets and Systems* 1998; **97**:33–48.
29. Chiclana F, Herrera F, Herrera-Viedma E. A classification method of alternatives for multiple preference ordering criteria based on fuzzy majority. *Journal of Fuzzy Mathematics* 1996; **34**:224–229.
30. Herrera-Viedma E, Herrera F, Chiclana F. A consensus model for multiperson decision making with different preference structures. *IEEE Transactions on Systems, Man, and Cybernetics* 2002; **32**:394–402.

DOI: 10.1002/cpe