Movement-aware and QoS-driven indoor location and mobile service discovery framework

Mohamed Adel Serhani*

College of Information Technology,
UAE University,
P.O. Box 17551, Al-Ain, UAE
Email: serhanim@uaeu.ac.ae
*Corresponding author

Wassim El-Hajj

Computer Science Department,
American University of Beirut,
Beirut 1107-2020, Lebanon
Email: we07@aub.edu.lb

Abstract: The ever evolving wireless technology and the advances in mobile devices induced a growing interest in indoor location-based services (ILBS). In this paper, we propose an ILBS discovery framework that finds the accurate indoor location of mobile users and provides information about services within the users’ surrounding area keeping in mind the client’s Quality of Service (QoS). It also provides support for movement-aware services where the path the mobile user takes is recorded to provide richer awareness of the client’s context than pure location. An ontology chain is proposed to support service discovery and to capture context information dynamically from different client’s locations, then map it to both single and composite web services. Comprehensive experiments, comparative study with other ILBS approaches, and performance evaluation were conducted. The results demonstrate that our proposed scheme outperforms the state-of-the-art methods presented in the literature and achieves very high localisation and service discovery accuracy.

Keywords: mobile computing; wireless networks; localisation; location-dependent; nomadic computing; service discovery; ontology; QoS; movement-aware services; indoor location.


Biographical notes: Mohamed Adel Serhani is an Associate Professor, CIT, UAE University, Al Ain, UAE. His research interests include web services architecture, management and QoS monitoring; service discovery in MANET; management of communities of web services; web services applications and security; and cloud computing for intensive e-health data, applications, and services; SLA enforcement in cloud data environment. He was the program Co-chair of the IEEE IIT’13, Chair of IEEE Workshop on Web service (IWCMC’13), Chair of IEEE Workshop on Web, Mobile, and Cloud Services (IWCMC’12), and Co-chair of NDT’12. He has published more than 65 refereed conferences, journals and book chapters.

Wassim El-Hajj is currently the Chairman of the Computer Science Department at the American University of Beirut (AUB). He joined the UAE University as an Assistant Professor, the AUB as a Visiting Professor, and then as a full-time faculty. He received many research grants for his research in information security, data mining, and network planning. His research activities include more than 50 publications, some of which were published in prestigious journals and conferences such as IEEE Transactions, IEEE Sensors, BMC Bioinformatics, IEEE Globecom, and IEEE ICC. He has also a valuable industrial experience with Boeing and Ten Strategic Consulting.

1 Introduction

Indoor location-based services solutions have attracted very high momentum in the past few years. This is due to the advances in mobile platforms (specifically smartphone devices), wireless communication technologies, embedded systems and mobile and pervasive computing. The Financial Times cites a JPMorgan prediction that 657 million smartphones will leave
stores in 2012. According to Morgan Stanley Research, these sales will exceed those of PCs in 2012. IDC predicts that smartphone sales will rise to 982 million in 2015. IMS research expects smartphones to reach 1 billion in annual sales in 2016. Hence, all these smartphones are expected to offer some kind of indoor location services. Based on these expectations and given the predicted widespread of smart devices, and the fact that humans spend over 90% of their time in indoor environments (Srivastava et al., 2012), a revolution in indoor location-aware applications and consequently in Indoor Location-Based System (ILBS) has occurred. According to the 2011 report by Indoor LBS, LLC (see www.indoor LBS.com), a company that has been monitoring and evaluating the location-based services industry since 2003, more than 100 companies are heavily investing in ILBS including Apple, Google, Microsoft, Cisco, Skyhook Wireless, MGM Casino, Qualcomm, Verizon, and many more. These companies are interested in ILBS because of the potential revenue it entails. For instance, when a shopper approaches a certain store in a mall, advertisers (most probably store owners) will send advertisements via the ILBS module installed on the shopper smartphone, attempting to persuade the shopper to buy from their products. Advertisers will have to pay the company that developed the ILBS solution for such service. Numerous other applications of ILBS exist including navigation applications within large buildings (for instance retail stores, convention centres, university campuses), along with entertainment and social applications such as social networking, local search, advertising and geo-tagging.

ILBS, however, continues to exhibit various challenges due to the constrained resources of mobile devices, the heterogeneity of mobile platforms and network environments, the user mobility and its varying contexts, preferences and quality expectations. To address these challenges, a comprehensive solution should be designed that enhances the localisation accuracy, considers user movement, manages different user contexts, and maps these contexts to a coordination of services using the concept of service composition.

In this paper, we suggest a solution that addresses these challenges just mentioned and closes the gaps in the work done so far on ILBS while enhancing the existing strategies when possible. Our solution for indoor location and service discovery determines the accurate location/path of mobile clients and provides information about services within the users’ surrounding area keeping in mind the client’s Quality of Service (QoS) requirements and the changing user’s context.

In our work, we take advantage of the fact that most current mobile devices are equipped with a wireless network card that provides users with connectivity to real-time information anywhere a Wireless Local Area Network (WLAN) infrastructure is present. We utilise the WLAN infrastructure to create a localisation module that effectively locates mobile clients and finds the path they are following. Our localisation module locates users with an error of zero metres 75% of the time and an error of less than one metre in the remaining 25%, which is better than the existing localisation approaches which have an average error of 2 m. Moreover, this module is resilient to access point (AP) failure and records the path of the moving user, not just a single instance.

We also take advantage of the web services benefits, mainly web service composition, and introduce a movement-driven service discovery and provisioning based on client’s QoS requirement, context and user/device profile. The suggested service discovery scheme satisfies scalable and changing client preferences (requirements) by aggregating web services together to provide richer services that better capture the user context and preferences. The discovery scheme is also movement-aware and context/QoS driven, in a sense that the context of the user is recorded at different situations/locations. Moreover, client’s requirements in terms of QoS are taken into account in the selection of the available services. A concept of ontology chain is used to support our movement-aware service discovery and capture context information from different user’s locations, then map it not only to a single service but also to an aggregation of services functionalities. These important features provide richer awareness of the client’s intent/context than pure location.

Our suggested indoor location and service discovery framework presents a complete solution to ILBS and introduces many contributions by: (a) suggesting a more accurate localisation approach than the existing approaches, (b) proposing ontology chain to capture context when the user is mobile, (c) using composition of web service to accommodate for unavailable services, (d) building service discovery on web service technologies to add flexibility and scalability, and (e) incorporating QoS and user profile in the service discovery process. The details of the contributions and the performance evaluation are discussed in more detail in Sections 4–7.

The rest of the paper is organised as follow: Section 2 presents the problem formulation. Section 3 overviews the work done on indoor localisation strategies and location context-aware services. Section 4 details the service discovery architecture, which includes the environmental set-up, QoS properties description, and service discovery components description. Section 5 explains the fingerprinting and matching phases of our localisation approach. Section 6 describes our movement-aware service discovery in which we describe the context using ontology chain, detail our service discovery operations using an illustrative scenario, and define the context-to-web services matching scheme. Section 7 presents the experimental evaluation of our localisation strategy, a comparative study of ILBS with other existing solutions, as well as the experimental and qualitative evaluation of our service discovery scheme, followed by an overall discussion and summary of obtained results. Section 8 presents the conclusion and points to possible future research directions.

2 Problem formulation

In this section, we present a formulation of the indoor service discovery problem. Hence, our approach for indoor service discovery is based on user’s context, QoS and ontology chain solves the following formulation.
Movement-aware and QoS-driven indoor location

- Let \( U_1, \ldots, U_n \) be a set of \( n \) indoor users.
- Each user \( U_i \) is associated with a profile \( P_i \).
- Each user’s profile \( P_i \) exhibits a set of \( QoS_i \) requirements. \( QoS_i \) properties may include parameters such as availability, reputation and response time.
- Each \( QoS_i \) property is defined by a couple of values \((t, w)\), where \( t \) is the targeted minimum/maximum value of the property and \( w \) is the weight of the property assigned by the client.
- Let \( WS_1, \ldots, WS_n \) be a set of \( n \) web services offering similar functionality but with different \( QoS_i \).
- Two types of services can be considered: Single Web Service (SWS) and Composed Web Service (CWS).
- A composite web service \( CWS_i \) is created by aggregating a set of single services \( (SWS_i) \).
- A context \( C \) describes the information from the user’s surrounding environment. The context is described using ontology \( O \), and is regularly mapped to the user profile \( P_i \).
- An ontology \( O \) describes and classifies context information along with their elements dependencies.
- Each service \( SWS_i \) responds to a single context \( C \), while each service \( CWS_i \) responds to a set of contexts \( (C_1, \ldots, C_n) \).
- A context information can be mapped to an ontology \( O \).
- An aggregation of contexts information could be mapped to a Chain of Ontologies \( O_{n}, \ldots, O_{m} \). Each service typically matches/provides context information with a certain \( QoS \). \( QoS \) parameters are normalised within values between 0 and 1.

3 Related work

In this section, we start by presenting the latest work on indoor localisation strategies. We then move to discussing the state of the art work on location context-aware services. We conclude the section by pointing out the differences between our approach and the presented work.

Throughout the years, there has been a lot of interest in indoor localisation systems most of which have used received signal strength indication for positioning (Sabbour, 2007; Bahl et al., 2000; Ni et al., 2004; Hermersdorf, 2006; LaMarca et al., 2005; Prasithsangaree at al., 2002; Roos et al., 2002; Nasipuri and Najjar, 2006; Kjærgaard, 2010; Tauber, 2002; Záruba et al., 2007). All these suggested RSSI-based approaches, including ours, use a two-phase mechanism to locate mobile users. The first phase is an offline phase where a radio map of the building is created (vector of signal strengths). The second phase is an online phase where nearest neighbour and probabilistic techniques are used to match the signal strength of the mobile user to that on the map. All these approaches share a common disadvantage in the sense that an offline phase is needed for the localisation to work. On the other hand, the main advantage of using RSSI this way is that the un-predictable RSSI variations in space are handled, which makes the approach a little more accurate. Errors using this method are reduced to an average of 1–2 m.

Our localisation strategy is also based on RSSI where it uses a fingerprinting offline phase and a matching online phase similar to the works referenced above. The major variation that our strategy presents is in the fingerprinting phase. By enhancing the fingerprinting phase, we were able to pinpoint the exact location (zero error) of the mobile user 75% of the time. In the remaining 25%, our method produces an error of less than 1 m, which is better than the other fingerprinting methods. Another feature of our localisation strategy is that it records the path that the
A number of research initiatives proposed solutions for improvement of wireless networks and mobile computing. Building context-aware mobile services based on short-range technologies like Bluetooth has been extensively studied (Cruz et al., 2011; Ye, 2005), but they all suffer from technological limitations. Localisation approaches based on Bluetooth are still under evaluation of five localisation systems introduced by Tauber (2002). Localisation systems are also proposed in the literature (Wang et al., 2011; Kjærgaard, 2010) proposed a method called Hyperbolic Location Fingerprinting (HLF) to address signal strength heterogeneity. In the fingerprinting phase, HLF does not record the absolute signal strength of the base station. It records the signal strength ratios between pairs of base stations. The advantage of HLF is that it can address the problem of clients measuring signal strength differently without requiring extra calibration. The disadvantage of HLF is that it heavily relies on the number of deployed base stations for accurate positioning. Moreover, the percentage of correct estimates is only 50%.

Záruba et al. (2007) proposed a system to locate mobile users using only one access point. This is in contrast to previous works where at least three access points were required to localise users inside office buildings restricted to the corridors of these buildings. The disadvantage of the suggested approach is that it applies for small structures such as homes and offices, but does not perform well in larger structures such as malls. A nice survey presenting an evaluation of five localisation systems is introduced by Tauber (2002). Localisation approaches based on Bluetooth technology were also proposed in the literature (Wang et al., 2011; Cruz et al., 2011; Ye, 2005), but they all suffer from coverage problems given the short range of Bluetooth.

We present next the up-to-date related work on location-based context-aware mobile services. Building context-aware services became crucial to cope with the expansion and improvement of wireless networks and mobile computing. A number of research initiatives proposed solutions for service discovery in LBS and ILBS; however, these solutions range from a middleware/platform/application-based solution, architecture-oriented solution, to a database-driven LBS solution.

Tao et al. (2005) discussed the possibility of using Service-Oriented Context Aware Middleware (SOCAM) architecture for the building and prototyping of context aware service. In addition, they conferred the usage of web ontology language to tackle issues including semantic representation, context reasoning, context classification and dependency. Tao et al. (2004) presented an idea very close to the one presented Tao et al. in the 2005 which highlighted the importance of designing SOCA architecture for the building and rapid prototyping of context-aware mobile services; knowing that the latter can be employed in many areas such as presenting context information to mobile users, triggering action/behaviour on the occurrence of a set of contexts, and adapting presentation of services to mobile users. Concerning the implementation phase, the authors incorporated the SOCA middleware into the open service gateway Initiative OSGi (www.osgi.org), since their prototype consists of an OSGi-compliant, provides secure and reliable service delivery. The authors concluded that their experimental results showed logical performance for context reasoning and searching, and the SOCA middleware is capable to meet the requirements of context-aware systems concerning limited memory and CPU resources.

Jessica and Christopher (2005) surveyed two types of mobile computing middleware: (a) context awareness based and (b) service discovery oriented, and how each one of them is used to facilitate a crucial portion of the function space for applications. They noted that middleware is needed to allow applications to be aware and adapt to changes in the user’s environment. However to be able to install middleware on a mobile device, reflection is required. CARISMA (Capra, 2003) and MobiPADS (Chan and Chuang, 2003) are two examples of projects that used reflection to allow mobile applications to achieve a context aware state.

Kapitsaki et al. (2009) attempted to decouple the techniques that enable the exploitation of contextual information in services known as context handling. To do so, they enumerated all context management categories, but only focused on the ones that are most appropriate for service engineering: (a) source code level, (b) model-driven (Schmidt, 2006), and (c) message interception. The authors’ main objective is to design and develop approaches that will ease business logic adaptation.

Pawar and Tokmakoff (2005) proposed Context-Aware Service Discovery (CASD) architecture to handle context matching using unambiguous context representation, disseminated seamlessly context information, and considered the dynamic nature of the context information. Ni and Sloman (2005) proposed a new Ontology-enabled Service Oriented Architecture (OSOA) for pervasive computing that is built on web services architecture, and incorporated Universal Plug and Play (UPnP). OSOA aimed to combine context awareness and human-centric requirements related to ubiquitous computing with a new ontology-based approach to ad-hoc and goal-driven service composition.
Broens et al. (2004) presented an approach for service discovery that uses ontologies to capture the semantics of the user’s query of the services and of the contextual information that is considered relevant in the matching process. Campo et al. (2005) proposed a new service discovery middleware. This middleware is composed of a service discovery protocol, a Pervasive Discovery Protocol (PDP), a service description language, and Generic Service Description Language (GSDL). Doulkeridis et al. (2005) presented context-aware service discovery architecture. They described the system implementation and presented the system evaluation as a trade-off between the increase of the quality of service discovery when context-awareness is taken into account and the extra cost/burden imposed by context management. Van Halteren and Pawar (2006) discussed the requirements for nomadic mobile service provisioning and proposed the Mobile Service Platform (MSP) as a supporting infrastructure and middleware extending the service-oriented architecture paradigm to the mobile device. The MSP design is based on the Jini surrogate architecture specification.

In order to meet the objective of realising a context and preference-aware location-based services, i.e. to insert different forms of preferences and context in core processing of location-based queries, Mokbel and Levandoski (2009) pioneered the system architecture of preference and context-aware location-based database server (CareDB). The ultimate intention of the authors was to ‘redefine the answers of existing location-based queries’ aiming at making them more appropriate to users’ small devices. Other location-based systems were discussed in the literature to provide database support for location-based queries. DOMINO (Wolfson et al., 2005) provided location-based services to support moving objects databases by integrating and adapting a critical set of capabilities on the top of existing database management systems. SECONDO (Guting et al., 2005) provided a ‘generic’ database system frame that can be filled with implementations of various DBMS data models, e.g., location-based services, through algebra modules. The PLACE server (Mokbel and Aref, 2005; Mokbel et al., 2004) provided a built-in approach to support location-based services through specialised location-based query operators inside DBMSs. It employed incremental evaluation, shared execution and load shedding techniques to support large number of concurrent spatio-temporal queries.

Although the literature review presented above includes rich work on ILBS, many drawbacks still hinder the good performance of the suggested solutions. The aforementioned works perform syntactic matching in the sense that they retrieve services’ descriptions that contain particular keywords from the user’s query, which often leads to poor discovery results. Moreover, the contextual information does not include the user QoS preferences, which are considered relevant in the selection of a given mobile service. Most of the above solutions use one-to-one matching (match context information to a single service) which may fail if a single service is unavailable and/or cannot provide the required user’s QoS preferences. Furthermore, they consider only a single user’s location which does not always provide sufficient information about the user’s context, and do not take into account the dynamic nature of the context information as the context changes frequently due to user movement.

On the other hand, our location-based services discovery approach addresses the drawbacks presented above and introduces the following contributions:

- The suggested localisation module provides better localisation accuracy than the state of the art localisation approaches. It supports movement-aware applications where the path the mobile user takes is recorded and used to better serve the clients.
- Ontology chain, which is a new approach, is proposed to capture a sequence of locations/context while the user is in movement.
- Composition of web services are activated when a mapping service is unavailable (or it cannot provide the client’s QoS preference) to replicate functionality of the unavailable service by merging similar functionalities of some other available services.
- Ontology is used not only to capture the user context/preferences but also to trigger dynamic service matching of services once composition of Web services is required.
- Service discovery is fully based on web service technologies that add more flexibility, scalability, and allows advanced business logic, especially once a composition of web service is triggered to respond to varying user contexts and preferences.
- Service discovery is based not only on the user context but also on the QoS of discovered service, and the user profile.

4 Service discovery architecture

4.1 Environment preparation

In what follows, we present the set-up that we have considered for the indoor location-based solution and the mobile user’s environment.

- Location selection application is installed and deployed on the user’s device.
- Mobile users are aware about the QoS provided by each service.
- Semantic representation of client preferences is registered in advance and made available to the application. Eventually, a user is able to update his/her preferences, if needed, via the application interface.
- The application server captures the device profile/characteristics (e.g., memory size, screen size).
- QoS properties of available services are advertised within the service description.
- Mobile users are guided via the application interface to specify their QoS preferences in the service discovery request.
4.2 QoS properties description

A large number of QoS properties have been used for mobile applications and services for instance response time, cost and reputation. These properties depend extensively on the nature of the services and the preferences the mobile users are considering. In this work, we consider four main QoS properties: response time, cost, availability and reputation. 

Response time represents the time needed between issuing a request and getting its response. Cost represents the cost charged for using a service. The service cost may be estimated by operation, volume of exchanged data, and/or a flat rate plan. Availability represents the probability that a service is accessible (available for use) or the percentage of time that the service is operating. Reputation represents the measure of service trustworthiness. It depends on clients’ experiences in using the service.

With regard to QoS-aware client request, the user can include his/her QoS requirements in the request via the application interface. For example, a service discovery request of a given user in a shopping Mall may include the closest restaurant to his location, his preferences, menu prices, the reputation of the requested service, and the response time.

4.3 Service discovery components

Figure 1 describes the n-tier model that we propose to support user location determination and service discovery for mobile users. The service discovery scheme is context-driven and QoS-aware, in a sense that client’s requirements in terms of QoS are taken into consideration in the selection of the requested services. Moreover, the context-awareness refers to the ability of the service to adapt to the changes in the surrounding environment.

As shown in Figure 1, mobile users using their mobile phones, PDAs or laptops to issue QoS-aware requests to the back end system, these requests are interpreted by a dispatcher. The localising procedure is executed first before any other operation. It is executed under two scenarios: (a) once the mobile user enters a WLAN coverage, he/she will be automatically discovered and located; and (b) once the client issues a discovery request to the back end system, the location identification module is executed first. Once the user is located within the covered network area, he/she discovers the services/resources available within this area. All services are implemented as web services that are deployed on web servers and invoked by the discovery module of the architecture. To handle the location of mobile users and the discovery of services, a couple of modules were implemented to support the following: request reception and admission, service discovery, QoS management, context representation management, and user profile management. A brief description of these modules is presented hereafter.

Request dispatcher: in charge of intercepting client’s requests from/to the client and forwarding them to the appropriate module(s) that handle the client’s requests. It also queues requests for later processing if the server is loaded, and may reject requests in heavy load situations.

Admission manager: classifies incoming requests and verifies the provisioned QoS with the QoS manager. It is responsible for determining whether the received requests are allowed to use the requested services. This means that web services access is denied if the client’s requests did not agree with the QoS requirements of the selected web services providers.

QoS manager: in charge of keeping track of the updated QoS information about all web services. It manages the QoS contract, stored in the QoS information database, which was previously agreed upon between the user and the service provider. It also monitors the QoS provision to report any probable violation of the QoS contract.
User/device profile module: in charge of managing users’ profiles, which includes their preferences, in terms of personalised services, current location, and required QoS. The user/device profile information is stored in the user profile registry. This module is also in charge of managing the user/device profile and mapping it to its context.

Service discovery module: in charge of service discovery and provision is based on the user location. Services are implemented as QoS-aware web services that are published on a QoS-aware registry. The discovery module receives the request from request dispatcher, retrieves the QoS requirements specified in the request and applies a selection policy (based on QoS) to select the appropriate web services that match the QoS requirements of the request. The details of the service discovery strategy are described in Section 6.

Location identification module: in charge of determining the user location. All other features of the architecture, for instance service discovery, rely on this location. This module applies fingerprinting and matching techniques to determine the location of mobile device and retrieves information associated with the user (device) location. The details of the localisation strategy are described in Section 5.

Context representation module: in charge of context registration, detection, handling, and update. Context is stored in a registry and updated and/or augmented whenever new context is added to the surrounding environment. The context is described using ontology and is regularly mapped (matched) to the user profile. Ontology chain is used to describe varying user contexts.

The application developer, who may get input from the client, for instance, when his profile changes, handles all the above operations. We detail next the fingerprinting and matching phases of our localisation module.

5 Localisation strategies

As mentioned earlier, the localisation approach we adopted is based on RSSI. It is composed of a fingerprinting phase and a matching phase (Sabbour, 2007; Bahl et al., 2000; Ni et al., 2004; Hermersdorf, 2006; Prasithsangaree at al., 2002; Roos et al., 2002; Nasipuri and Najjar, 2006; Kjærgaard, 2010; Tauber, 2002; Záruba et al., 2007). Our work differs from the others in the fingerprinting phase, which we enhanced to make the matching more accurate and more stable with respect to the outliers in the signal measures. It also includes a module to record the path the user takes rather than the absolute location. This section details the fingerprinting and matching phases.

Our test bed is deployed on the third floor of the Faculty of Information Technology (FIT) at UAE University. The dimensions of the floor are 36 m by 72 m, which includes more than 70 rooms distributed between faculty offices and conference rooms. Fifteen Cisco Aironet 1131AG Series access points are scattered around the floor, each using dual integrated radios (IEEE 802.11g and IEEE-802.11a). The mobile hosts are Dell laptops running Microsoft Windows XP. It is worth noting that due to the walls and radio interference, a single access point cannot cover the whole floor. In fact, a single access point can cover a radius of 25 m with good signal strength. Beyond 25 m, the signal strength starts deteriorating depending on the existing obstacles. Figure 2 shows the considered reference points.

![Figure 2 Floor map with reference points](image-url)
5.1 Fingerprinting phase

The purpose of the fingerprinting phase is to build a table of reference points to be used in the matching phase to determine the user’s location. For every reference point, we measure the RSSI of all the Access Points (AP) in proximity. The sensing lasts for 30 s that is long enough to acquire clear readings, yet short enough to finish scanning in short time. The period of 30 s could be calculated adaptively according to the user’s moving speed as the user can stop, move slowly or move fast which may affect the readings. Making this sensing period adaptive is beyond the scope of the paper. During this duration, the mobile device at a given reference point hears from many APs and for every AP it hears many signals. We group the readings according to the AP Media Access Control (MAC) addresses. For each group (the signals heard from a particular AP), we calculate the frequency (occurrences) distribution of every RSSI signal. Based on the frequency, we determine the leader signal of every group. Traditionally, the leader is the RSSI signal that is the most frequent. In our work, we consider a different approach that starts by eliminating the RSSI signals that are very low and then choosing as a leader the median of the remaining signals. For every group (AP), we sort all the RSSI readings and discard the ones that fall below the first quartile. This step cuts off the lowest 25% of the RSSI readings, which we consider noisy signals and consequently should be ignored. From the remaining 75%, we find the median RSSI and consider it the leader RSSI for that group (AP). The mentioned procedure is repeated for the rest of the groups. When all the groups are processed, we add to the database the RSSIs of all the leaders. If at a certain reference point, only two APs can be heard, the leaders of these two APs will be added to the database. For instance, assume that reference point P12 (in Figure 2) hears five APs. The leader RSSI of every AP is calculated using the aforementioned procedure producing a vector for P12:

\[ \tilde{F}_{p12} = (L_{AP1}, L_{AP2}, L_{AP3}, L_{AP4}, L_{AP5}) \]

where \( L_{API} \) is the leader RSSI of \( AP_i \). \( \tilde{F}_{p12} \) is then sorted in descending order and stored in the database. A similar vector is stored for every reference point.

Hence, our fingerprinting approach, which we call Quartile-Median Fingerprinting (QMF), differs a lot from the traditional fingerprinting approaches. To choose the leaders, we conducted an elimination step based on quartile followed by a selection step based on the median. Traditionally, neither elimination phase is performed nor are thresholds used to eliminate some RSSI. Also, leaders are selected based on the average RSSI or the RSSI that occurs most often. The way we used median and quartile allowed our approach to be more robust to the outliers of signal measures. It also produced more accurate results than the traditional fingerprinting approaches and is more resilient to AP failures, i.e. when only two APs can be heard. This is illustrated in Section 7 where we compare QMF with the approach that selects as a leader the most frequent RSSI. Hence, this latter approach is the most widely used method for fingerprinting.

5.2 Matching phase

In what follows, we explain how the database populated in the fingerprinting phase is used to pinpoint the position of a moving user. This phase is called the matching phase and it uses the Euclidean distance method. Hence, this phase is close to the matching phases presented in the literature, but with an added feature to record the user path, not only the user location.

A moving user (mobile device) takes readings of the audible access points for 30 s. It groups the readings based on the MAC addresses and elects the leaders according to the procedure discussed in Section 5.1. The user then constructs a vector composed of the leader RSSIs sorted in descending order, i.e.

\[ \tilde{O} = (O_{AP1}, O_{AP2}, \ldots, O_{APn}) \]

Vector \( \tilde{O} \) will then be embedded in an XML message and sent to the server. If the user can only hear two APs, only two entries will be embedded in the XML message. The server performs a server side matching and responds with an XML message indicating the user’s location.

To find the user’s position, the server calculates the Euclidean distance between the observed vector \( \tilde{O} \), and every vector \( \tilde{F} \) stored in the database. This operation produces \( |P| \) distances (the number of reference points), each being the distance between the location of the user and one of the reference points. The reference point that results in the minimum distance is selected to be the position of the user. The Euclidean distance between vectors \( \tilde{O} \) and \( \tilde{F} \) is given as follows:

\[ \text{Distance} = \sqrt{\sum_{i=1}^{n} (o_i - f_i)^2} \]

where \( n = \min(|\tilde{O}|, |\tilde{F}|) \), \( o_i \) is the observed signal value and \( f_i \) is the signal value stored in the database. An XML reply is sent from the server to the user indicating his estimated location.

We appended the matching phase discussed above with a module that allows the matching to be done every other 30 s. In other words, the new location of the mobile device is calculated every other 30 s, leading to the construction of the path the user is taking. This update is done to support movement-aware applications, which allow our service discovery module to provide better services to the mobile users. In the next section, we describe the movement-aware service discovery scheme and its main components mainly the context representation, the ontology chain description, and the service composition.

6 Movement-aware service discovery

In this section, we describe the service discovery approach supported by the service discovery module of the architecture described in Figure 1. The service discovery is context-driven, movement-aware, and relies on the user’s QoS requirements. It uses semantically described context using a concept of ontology chain. We define an ontology chain to describe context description, as well as rule definition and execution in order to support service discovery.

6.1 Context description using ontology chain

In this section, we define a new concept of ontology chain and its description illustrated to context of FIT, UAE University. Furthermore, we describe example of rules that developed for context to ontology reasoning.
To describe our ontology, we use the Ontology Web Language OWL 2 (W3C: OWL 2, 2009a), which considers four entities: computational entities, location, person and activity. We extended the entities description by adding new entities such as resource, device type, and network. We also defined a new concept of ontology chain where we record many instances of indoor user context locations for which a number of web services are available to respond to the needed service. Therefore, an ontology chain can be defined as a description of ontology that captures varying user’s context information and map it to suitable web service or aggregation of web services that respond to user’s requirements in terms of QoS. Since there are a number of available competing web services, the selection strategy of best match web services is fully based on the user’s context and QoS preferences. This selection is supported by the ontology chain description since for each series of locations a number of candidate services are considered.

Figure 3 shows a partial definition of a domain-specific ontology for FIT. We use the OWL 2 Rule Language (RL) (W3C: OWL 2, 2009b), which is a rule subset of OWL 2, for context ontology reasoning. We define rules using Semantic Web Rule Language (SWRL) and infer those rules using Java Expert Shell System (JESS). All rules are expressed in terms of OWL concepts (e.g. classes, properties, individuals and literals). Table 1 presents some examples of SWRL rules that have developed for FIT environment.

The above rules are subject to being interpreted (inferred) using the context interpreter within the context representation module, which in turn validates these rules and manages their execution.

6.2 Service discovery sequence of operations

Figure 4 describes the process of capturing context information and user preferences (QoS) of mobile users using an ontology chain; the latter is used afterward to activate dynamic web service composition. Although the user might be moving, the ontology might not change and thus the user context remains the same. In this case, we only map the ontology to a single web service (one-to-one mapping). In the case when the context changes, the ontologies will also change from one location to another, and therefore we will have a variation of ontology (ontology chain) to which we map different services (many-to-many mapping). Therefore, a composition of web services becomes a necessity to reflect richer awareness of user’s context/QoS.
The process of initiating the movement-aware service discovery starts by recording the series of locations using the localisation approach, followed by capturing context information and user preferences (QoS), then analysing a series of context information and mapping them to ontology chain, and finally matching ontology to different services and triggering service composition once required. The details of the matching operations are described in the next section.

6.3 Matching ontology chain to web service(s)

To facilitate the mapping process of the ontology described using OWL 2 to Web Service Description Document (WSDL) written in XML, we use the XML serialisation for OWL 2 (W3C: OWL 2, 2009a) that mirrors the structural specification of OWL 2 to an XML representation. The XML serialisation presents a notational variant of the functional syntax. Therefore, the mapping will be between the functional aspect of the ontology description and the functional behaviour described as web service operations in the WSDL document.

The matching strategy we have implemented follows the following steps:

1. Process the ontology described with OWL 2.
2. Select candidate web services that correspond to the described ontology chain.
3. Decide about a single service selection or a composition of web service by considering the user’s context and QoS requirements.
4. Process the selected web service(s) WSDL(s) documents.
5. Extract information of web service operations (e.g. operation name, inputs and outputs) and classify these operations to ease the mapping.
6. Perform OWL 2 to WSDL(s) mapping based on the functional properties described in both documents.
7. Invoke a single web service once a composition of services is not required.
8. Build dynamically the BPEL process by invoking the candidate web services partners if a composition of web services is essential.

6.4 Service discovery scenario

Dubai Mall is the world’s largest shopping Mall built on 12 million square feet with eight levels parking of 14,000 car parking spaces, hotels, mega cinema, aquarium and underwater zoo, more than 1200 shops, around 200 cafes and restaurants, and lots of other attraction areas.

We assume that the mobile user preferences are registered and updated by the user frequently and the mobile device characteristics are captured. Consider Adam, a person equipped with a smartphone mobile device, who is used to visit Dubai Mall very frequently as his work location is 5 min driving from Dubai Mall. Today, Adam is visiting Dubai Mall at 12 noon during his one and a half hour midday break. Adam, using our indoor location application, will be guided to the nearest parking area in the adequate level that is closest to the prayer room as the prayer time is at 12:15 (direction web service). Then he will be directed to the nearest restaurant(s) that meets Adam’s preferences (dining web service). Afterwards, and since he still has time, Adam will be advised to take coffee (dining web service), knowing that he may request this service himself via the application interface and specify his QoS preferences (e.g. proximity and price). If he still has some time left, he will be reminded to buy a birthday gift to his workmate (shopping web service); Adam should have registered in his device’s electronic agenda that his friend’s birthday is today. Finally, Adam will be guided to the parking spot where he left his car (direction web service).
According to the above scenario, the service discovery is based on varying Adam’s contexts that is captured by ontology chain and requires a composition of Web services. Figure 5 summarises the service discovery main steps.

7 Performance evaluation

In this section, we conduct experiments to evaluate the accuracy of the localisation approach and show how it outperforms state-of-the-art approaches presented in the literature. We also evaluate the service discovery based on the location strategy defined in the previous experiments.

7.1 Experimental evaluation of location accuracy

7.1.1 Experiment set-up

Table 2 summarises the main tools, languages, APIs and technologies that we have used to implement our indoor location solution.

<table>
<thead>
<tr>
<th>Tools and languages</th>
</tr>
</thead>
<tbody>
<tr>
<td>MySQL 5 as the back end database for storing the different data required by the application.</td>
</tr>
<tr>
<td>Web services as the gateway to provide the users with the services and content in the form of HTML webpage.</td>
</tr>
<tr>
<td>JAVA, JSP and XML as the programming languages to interface with the database and render the content with the help of the web services.</td>
</tr>
<tr>
<td>OWL language for ontology description</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Platforms/APIs</th>
</tr>
</thead>
<tbody>
<tr>
<td>NetBeans IDE version 6.9.1 as the programming environment used to develop the application.</td>
</tr>
<tr>
<td>PlaceLab NDIS Wrapper (Capra, 2003; OSGi, <a href="http://www.osgi.org">www.osgi.org</a>) to add functionality to Java and enable the collection and use of raw data acquired from the wireless card.</td>
</tr>
<tr>
<td>JDOM v1.1 as a Java-based solution for accessing, manipulating, and outputting XML data from Java code.</td>
</tr>
</tbody>
</table>

7.1.2 Experimental scenarios and results discussion

As mentioned in Section 5, we deployed our test bed on the third floor of FIT building. We installed our customised portable Java-based software on a couple of dell laptops running Windows XP. Although, the major feature of our software is to test the localisation strategy, we extended it to record and show the user his way around the building. Once the application is launched, the user is located on the map and lines are drawn to show the user how to get to his/her destination. We present next the accuracy of our localisation module (QMF).

We compared QMF to the most used localisation approach in the literature which elects as a leader the most frequent RSSI signal. We tested the accuracy of both approaches when each reference point can sense two, three, four, and five access points, respectively. For each reference point, we calculated the error rate which is defined by the difference between the estimated location and the actual one. Figures 6 and 7 show the obtained error rates. The output presented in the figures is the average outcome of ten runs. Figure 6 shows the accuracy of locating the users when the leader is elected to be the most frequent RSSI signal (the most used approach in the literature). This experiment indicates that the exact location of the user was pinpointed around 19% of the time when five APs can be heard. In the remaining 81%, the error ranges between 1 and 14 m. When the user can hear four APs, this approach was not able to find the exact location of the user at least once. It produced an error of 2 m almost 25% of the time. When the user can hear three APs, the location was off by an average of almost 8 m. The worst results were achieved when the user can only hear two APs with an average error of 11 m.

On the other hand, when QMF was used (Figure 7), the user was exactly pinpointed around 75% of the time when four or five APs can be heard. In the remaining 25%, the average error was less than 1 m. When the user can hear four APs, this approach was not able to find the exact location of the user at least once. It produced an error of 2 m almost 25% of the time. When the user can hear three APs, the location was off by an average of almost 8 m. The worst results were achieved when the user can only hear two APs with an average error of 11 m.

As mentioned in Section 5, we deployed our test bed on the third floor of FIT building. We installed our customised portable Java-based software on a couple of dell laptops running Windows XP. Although, the major feature of our software is to test the localisation strategy, we extended it to record and show the user his way around the building. Once the application is launched, the user is located on the map and lines are drawn to show the user how to get to his/her destination. We present next the accuracy of our localisation module (QMF).

We compared QMF to the most used localisation approach in the literature which elects as a leader the most frequent RSSI signal. We tested the accuracy of both approaches when each reference point can sense two, three, four, and five access points, respectively. For each reference point, we calculated the error rate which is defined by the difference between the estimated location and the actual one. Figures 6 and 7 show the obtained error rates. The output presented in the figures is the average outcome of ten runs. Figure 6 shows the accuracy of locating the users when the leader is elected to be the most frequent RSSI signal (the most used approach in the literature). This experiment indicates that the exact location of the user was pinpointed around 19% of the time when five APs can be heard. In the remaining 81%, the error ranges between 1 and 14 m. When the user can hear four APs, this approach was not able to find the exact location of the user at least once. It produced an error of 2 m almost 25% of the time. When the user can hear three APs, the location was off by an average of almost 8 m. The worst results were achieved when the user can only hear two APs with an average error of 11 m.

On the other hand, when QMF was used (Figure 7), the user was exactly pinpointed around 75% of the time when four or five APs can be heard. In the remaining 25%, the average error was less than 1 m. When the user can hear four APs, this approach was not able to find the exact location of the user at least once. It produced an error of 2 m almost 25% of the time. When the user can hear three APs, the location was off by an average of almost 8 m. The worst results were achieved when the user can only hear two APs with an average error of 11 m.
QMF, which is based on quartile and median techniques, clearly gives very accurate results and outperforms the well-known method that uses as a leader the most frequent RSSI signal. We can also note that the more access points the user can hear, the more accurate the position determination is. QMF is also very resilient to AP failure where the user can hear only two APs. An average error of 7 m in such unusual scenarios can still be considered acceptable.
7.1.3 Comparison with other indoor localisation techniques

In addition to the experimental evaluation of our localisation scheme, here we present a comparison with other well-known localisation techniques while considering a set of properties such as localisation accuracy, precision, cost and robustness. This comparison was mainly inspired from the survey done by Liu et al. (2007) in addition to other experimental results published in the work of Sabbour (2007), Bahl et al. (2000), Ni et al. (2004), Hermersdorf (2006), LaMarca et al. (2005), Prasithsangaree et al. (2002), Roos et al. (2002), Nasipuri and Najjar (2006), Kjærgaard (2010), Tauber (2002) and Zářuba et al. (2007).

The results presented in the previous section and those presented in Table 3 prove the merit and contributions of QMF. QMF performs very well in terms of accuracy and precision, which are very crucial comparison criteria for indoor localisation. It also performs quite well with regard to the other comparison criteria for instance cost, robustness and ease of deployment. Moreover, it significantly outperforms the other techniques in recording mobile user movement (via path labelling), producing better services provisioning.

7.2 Quantitative experimental evaluation of service discovery

7.2.1 Experimentation set-up

We conducted an experimental quantitative evaluation of service discovery-driven measurement scheme using real-experimental set-up. We consider the measurement and the evaluation of the following quality parameters: service response time, availability, scalability, service discovery success (accuracy) and throughput. We tested our indoor location and service discovery application on the FIT building (3rd floor), which is equipped with 15 Cisco Aironet access points. The application is deployed on a computer hosting an Apache application server. We used the same set-up which is described in Section 5 (indoor localisation). The software/hardware requirements we used for service discovery are listed as follows.

1. Dell computer running Microsoft windows XP and hosting the application server Apache Tomcat 6.0.
2. A couple of smartphone mobile devices: Nokia mobile N900 running Linux OS Maemo, CPU, Cortex (600 MHz), 32 GB mass storage, 256 MB RAM.
3. JMeter software for requests generation, and trace collection.
5. Netbeans 6.9.1 for web service implementation and testing.
6. A couple of indoor services were developed as web services (SOAP and restful web services). These web services include user localisation, user direction and user search web services
7. Microsoft Outlook is used to register user’s preferences and schedules.

7.2.2 Web service description

Table 4 briefly describes the list of web services that have used to test our service discovery application.

<table>
<thead>
<tr>
<th>Comparison Criteria</th>
<th>Accuracy</th>
<th>Precision (metre m)</th>
<th>Relative cost</th>
<th>Ease of deployment</th>
<th>Scalability</th>
<th>Robustness</th>
<th>Adaptive context-aware service discovery support</th>
</tr>
</thead>
<tbody>
<tr>
<td>Techniques</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Our enhanced fingerprinting-based RSSI (QMF)</td>
<td>0–1 m</td>
<td>0 m 75% of the time, and less than 1 m in the remaining 25%.</td>
<td>Very low</td>
<td>Minimum set-up work</td>
<td>High</td>
<td>Very Good</td>
<td>Full Support</td>
</tr>
<tr>
<td>Fingerprinting WLAN RSS/RSSI</td>
<td>1 to 5.9 m</td>
<td>50%–90% within 2.1 m to 5.9 m.</td>
<td>Low</td>
<td>Easy to deploy</td>
<td>Moderate</td>
<td>Good</td>
<td>Partial support</td>
</tr>
<tr>
<td>RFID</td>
<td>&lt;2 m</td>
<td>50% within 1 m</td>
<td>Low</td>
<td>Moderate to deploy</td>
<td>Nodes placed densely</td>
<td>Poor to Good</td>
<td>No support</td>
</tr>
<tr>
<td>Hybrid (TDOA, AOA, UWB, RSS)</td>
<td>15 cm to 2 m</td>
<td>50%–99% within 0.3 m and 1 m.</td>
<td>Medium to High</td>
<td>Hard to deploy</td>
<td>Moderate</td>
<td>Poor</td>
<td>No Support</td>
</tr>
<tr>
<td>Bayesian methods</td>
<td>1 to 1.5 m</td>
<td>50% within 1 to 1.5 m.</td>
<td>Low</td>
<td>Easy to deploy</td>
<td>Moderate</td>
<td>Good</td>
<td>No support</td>
</tr>
<tr>
<td>Geometry based methods</td>
<td>2 to 3 m</td>
<td>50% within 3 m.</td>
<td>Medium</td>
<td>Moderate to deploy</td>
<td>Moderate</td>
<td>Good</td>
<td>Partial support</td>
</tr>
<tr>
<td>Others</td>
<td>&lt;1 to 10 m</td>
<td>50%–99% within approximately 1 m to 10 m</td>
<td>Low to high</td>
<td>Varies from Easy to complex to deploy</td>
<td>Low to High</td>
<td>Poor to Good</td>
<td>Partial to No support</td>
</tr>
</tbody>
</table>
Table 4 Description of web services used in the experiments

<table>
<thead>
<tr>
<th>Context</th>
<th>Web Services</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Discovery web service</td>
<td>This web service returns the results of service discovery request (search for a specific service, e.g. cafeteria).</td>
</tr>
<tr>
<td>2</td>
<td>Localisation web service</td>
<td>This web service returns the location of a faculty office, library, lab, administration, registrar, etc.</td>
</tr>
<tr>
<td>3</td>
<td>Direction web service</td>
<td>This service returns the direction to a pre-specified in door location.</td>
</tr>
<tr>
<td>4</td>
<td>Composite web service (discover, locate and get direction)</td>
<td>This service is based on the composition of the above three web services. It allows discovering a given service locate it and get the direction to it.</td>
</tr>
</tbody>
</table>

7.2.3 Experimented scenarios and results discussion

In this section, we evaluate the main features of service discovery scheme including the mobile user context, QoS information retrieval, ontology chain mapping, and the service discovery operation’s invocation. These scenarios aim to evaluate the effectiveness, and the reliability of our discovery scheme with regard to the following metrics: service response time, service availability, service throughput, mobile user mobility and context-service matching.

The description of these experimental scenarios and the discussion of the obtained results are following:

1 Service response time: we generate an increasing number of service discovery requests to our application and we measure the average response time using JMeter. We classify requests into two types: requests that are handled by a single web service and requests that are handled by a composite web service. Figure 8 illustrates the variation of the response time calculated for couple of web services while increasing number of requests to 400 requests. As shown in the figure the response time increases proportionally with the number of generated requests to all web services under testing. The curve of composite WEB service response time is considerably higher compared to the response time of the single web services (e.g. direction web service). This can be explained by the fact that a composite web service requires extra processing time as it handles a number of invocations to different single web services, then gather the results, and consolidate the response back to the service requester.

2 Service availability: we generate an increasing number of discovery requests to the same web services that used in the previous scenario. Then, we measure the average availability of web services both single and composite. Figure 9 illustrates the average service availability through increasing number of requests. For all single web services, the availability is relatively maintained high within an interval of [90%–100%]. With regard to composite web service, the availability is still good and maintained within an interval of [78%–86%]. Since the composite web service is made up of aggregating different single web services, its availability should reflect the lowest availability of one of single composing web service. The obtained results prove that our web services both single and composite maintain a high availability even with a high number of requests (400 simultaneous requests) invoked from mobile devices.

3 Service discovery success: in this scenario, we generate an increasing number of requests to the same web services single and composite as used in the previous scenarios (50, 100, 150,…, 400 requests). Then, we evaluate the success degree of matching user context to web service(s). The success degree measures whether the mobile requestor was able to discover the service that meets his/her preferences, context, profile and QoS. Figure 10 illustrates the degree of matching user context and QoS to appropriate service or composition of services. The curve shows that this degree is quite high for all web services both single and composite services. This result confirms that our matching scheme works well even if an aggregation of services is triggered to respond to user context and QoS preferences.

4 Throughput: in this scenario, we evaluate the web services both single and composite throughput in terms of the number of requests handled per seconds. We increase the number of requests and measure the service throughput. As shown in Figure 11, the service throughput is relatively high for location and discovery service and quite low for direction service and composite web service. For all web services under testing, the throughput decreases gradually with the increasing number of requests. This is explained by the fact that the throughput is affected by the amount and the size of data handled by each request; the heavier the requests are, the less requests a web server can handle. Therefore, for heavy requests and/or composite requests the throughput is low and for light requests the throughput is high. The overall results show an acceptable web service throughput although these web services are accessed from a mobile device.

5 User’s mobility: in this scenario, we evaluate the effect of user’s mobility on service discovery success. We set the total number of generated requests to 150 requests and increased gradually the mobile user speed to reach a maximum of 8 ms⁻¹. In this scenario, we use the same web services that used above as single and composite web services. Figure 12 illustrates the service discovery success ratio versus the mobile user’s speed. The result shows that the service discovery success ratio is maintained very high and it is insignificantly affected by the user mobility. We noticed only a minor degradation in the web service response time within the same experiment. The obtained results prove that our indoor service discovery scheme is...
not affected by the user’s mobility and exhibit a high service discovery success ratio for both single and composite web services.

7.2.4 Comparison with other service discovery techniques

In this section, we compare various mobile context-aware service solutions using a set of criteria inspired from Tao et al.’s (2005), Jessica and Christopher’s (2005) and Kapitsaki et al.’s (2009) work, as well as from our previous experience. This comparison can help context-aware services designers and developers to choose among the list of competing approaches based on the degree of satisfaction with regard to the comparison criteria. Table 5 highlights the strengths and limitations of the studied approaches.

The comparison described above shows that our technique applies dynamic matching of user’s context not only to a single service but also to a composition of services, which translates to richer service features and functionalities. It also records multiple users’ locations through an ontology chain scheme, which better reflects the user’s context and preferences. Moreover, it performs quite well with regard to the other comparison criteria for instance QoS and user’s preferences considerations and flexibility and ease of use.

Figure 8  Average response time vs. number of requests

![Figure 8](image)

Figure 9  Average availability vs. number of requests

![Figure 9](image)
Figure 10 User context-web service matching degree

![Figure 10](image)

Figure 11 Throughput vs. number of requests

![Figure 11](image)
Table 5 Comparison of selected service discovery techniques

<table>
<thead>
<tr>
<th>Comparison Criteria</th>
<th>Matching Technique</th>
<th>QoS and User’s Preferences</th>
<th>Localisation Data Support</th>
<th>Context Adaptation</th>
<th>Service Composition</th>
<th>Flexibility and Ease of Use</th>
</tr>
</thead>
<tbody>
<tr>
<td>Our movement-aware service discovery</td>
<td>Dynamic matching (one to many)</td>
<td>QoS and user’s preferences are both considered</td>
<td>Support multiple location</td>
<td>Dynamic and movement-aware</td>
<td>Dynamic service composition done on the fly</td>
<td>Easy to use (user is supported in executing the application’s features)</td>
</tr>
<tr>
<td>Middleware for service discovery (Tao et al., 2005; Tao et al., 2004; Jessica and Christopher, 2005; Capra, 2003; Chan and Chuang, 2003; Campo et al., 2005; van Halteren and Pawar, 2006)</td>
<td>Syntactic matching</td>
<td>Few middleware supported limited QoS consideration</td>
<td>Only Current localisation data is stored</td>
<td>Static</td>
<td>Workflow based</td>
<td>Need extensive knowledge and require some configuration</td>
</tr>
<tr>
<td>Agent-based solutions for service discovery (Soldatos et al., 2007)</td>
<td>One to One matching</td>
<td>Partially considered</td>
<td>Only Current localisation data is stored</td>
<td>Static</td>
<td>Composite contextual information</td>
<td>Need an agent platform to be supported</td>
</tr>
<tr>
<td>Architecture-based service discovery (Ni, and Sloman, 2005; Doulkeridis et al., 2005; Mokbel and Levandoski, 2009)</td>
<td>One to One matching</td>
<td>Partially considered</td>
<td>Only Current localisation data is stored</td>
<td>Static</td>
<td>Not supported</td>
<td>Need extensive knowledge and require some configuration</td>
</tr>
</tbody>
</table>

7.3 Qualitative evaluation of service discovery

To complement the quantitative experimental evaluation and the comparative study we have done with other ILBS solutions described in previous sections, we conduct a qualitative evaluation for our service discovery solution. We have presented the application to different mobile users with different context scenarios. The users collect their perception and satisfaction with regard to the service discovery success using different metrics as shown in Table 6. For this purpose, we selected 50 users to evaluate the usability and discovery success of our application and to report their experience by answering a questionnaire. We described five different contexts’ information and we
divided the users into five groups, each group was assigned a given context information described earlier. We also included situations where service composition is required to satisfy user context, its QoS and preferences. All users were computer literates including students, faculty and staff. Participants in this experiment received detailed guidelines on how to access and execute features of the service discovery application as well as other information such as how to express their QoS preferences in requesting a given service. Table 6 describes the evaluation factors and the set of metrics used to assess the degree of their fulfilment using a questionnaire.

Based on the users’ feedback (Table 6), we evaluated the service discovery according to three main metrics: quality of service discovery, functionalities and flexibility, and diversity of our service discovery scheme.

### Table 6 Evaluation results of context-aware service discovery

<table>
<thead>
<tr>
<th>Metrics</th>
<th>Questions</th>
<th>Answers</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Quality</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Service availability</td>
<td>A. Services all time available.</td>
<td>30</td>
</tr>
<tr>
<td>Service cost</td>
<td>B. The cost of services is always appropriate.</td>
<td>29</td>
</tr>
<tr>
<td>Response time</td>
<td>C. The response time is adequate.</td>
<td>25</td>
</tr>
<tr>
<td>Reputation</td>
<td>D. Highly trust the provided services.</td>
<td>21</td>
</tr>
<tr>
<td>Context awareness</td>
<td>E. Discovered service offer richer user’s context awareness.</td>
<td>28</td>
</tr>
<tr>
<td><strong>Functionality &amp; flexibility</strong></td>
<td>Describe your experience in using the service discovery features and interface?</td>
<td></td>
</tr>
<tr>
<td>Perceived usefulness</td>
<td>A. Useful in providing services to mobile users.</td>
<td>36</td>
</tr>
<tr>
<td>Ease of use and access</td>
<td>B. Easy to use and get familiar.</td>
<td>31</td>
</tr>
<tr>
<td>Interface conviviality</td>
<td>C. The interface design is easy to navigate through.</td>
<td>40</td>
</tr>
<tr>
<td><strong>Diversity</strong></td>
<td>How discovered services are diverse?</td>
<td></td>
</tr>
<tr>
<td>Consistency</td>
<td>A. Consistent in tracking services information (Data).</td>
<td>29</td>
</tr>
<tr>
<td>Diversity</td>
<td>C. Serve variety of users with different requirements</td>
<td>37</td>
</tr>
</tbody>
</table>

The evaluation of the quality of service discovery satisfaction was measured by four metrics: availability, service quality, response time, reputation, and context awareness. An overall result of 83% of users showed their satisfaction towards the quality of discovered services. They also expressed their agreements on the ability of the application to provide wealthy awareness of the user’s intent and its context. However, the remaining number of users faced few difficulties in discovering or expressing their services requirements. This can be explained by the high expectations these users are seeking from the provided services. It can also be justified by the difficulty they faced in understanding their context and estimating their expected QoS and preferences.

For the evaluation of the application functionality and flexibility, three metrics were used: usefulness, ease of use and access, and interface conviviality. In this study, the results showed that the majority of users (around 95%) showed a high perception of the application ease of use, its flexibility and its interface conviviality. However, the few users who disagreed expressed a need for extra time to gain experience and be more confident with the application features and functionalities. These results proof that the application is well designed, easy to use, and very useful for mobile users seeking indoor services.

With regard to the evaluation of service diversity, and consistency, most users agreed on the variety of services offered via the application. They eventually agreed on the consistency of the application and expressed their trust in scaling the application to support extra services. All services are developed as web services which make them easy to update, scale and simple to adapt.

### 7.4 Overall evaluation and discussion

Based on the results obtained from the experimental evaluation of the indoor localisation scheme and the quantitative and qualitative evaluation of the service discovery scheme, we draw the following conclusions:

- A user is located accurately and provided with the best match services that capture his/her context and respond to his/her QoS preferences.

- Service composition adds value to the discovery scheme as it responds to varying user preferences and dynamic context.

- The new ontology chain we proposed supported the movement-aware service discovery and efficiently triggered web service composition. Hence, adapted the varying user’s context to diverse services.

- The measurement scheme we implemented to measure the functional and non-functional behaviour of the indoor system properties focused on core features such as richness of context information, variety and adaptability of discovered services.

- Our discovery solution handles a large number of requests generated from varying mobile user’s context and preferences.
• User context is updated dynamically and all updates are considered in discovering services.
• User mobility has no major effect on service discovery success.

8 Conclusion

In this paper, we have proposed an indoor location-based service solution driven by context model and users’ QoS requirements. The service discovery model we proposed is movement-aware where the path the mobile user takes is recorded; the fact that provides richer awareness of the user’s intent/context than pure location. Moreover, our discovery scheme demonstrates that the process of QoS-aware mapping explores alternative ways of web service composition, which provides wider alternatives of web services that guarantee the QoS of the user. A set of modules have been implemented to support various features mainly user location determination, building of context-aware services, support service discovery, and web service composition. The location determination approach we suggested uses an enhanced fingerprinting scheme and matching algorithm that demonstrates exceptional location accuracy. In addition, it supports movement-aware applications by running the localisation module periodically.

The prototype system, the evaluation results and the comparative study with other ILBS solutions demonstrated that a mobile user is accurately located and serviced while meeting the requirements of context-aware environment, client’s QoS requirements and user/device profiles. As future work, we plan to extend our solution to support monitoring of service provision in order to guarantee the QoS of provisioned services. We also plan to propose an adaptation mechanism to adapt services once their QoS are degraded or violated.

References


