# A Context Aware Ontology Based Middleware Framework for Service Discovery

**E.Christopher Siddarth**
Research scholar
Department of Computer Science and Engineering
Annamalai University , Annamalainagar,Chidambaram
Tamil Nadu 608001, India

**Summary**
Persevering advances in the fields of wireless network and mobile computing have necessitated the advanced applications and services to incorporate context-awareness, a process that facilitates adaptation to changes in environment. Context-awareness is considered to be an imperative and beneficial feature in distributed mobile networks. Commonly, mobile devices have certain constraints such as processing, storage space and more. Context awareness is being brought into play so as to overcome these constraints. Context plays a vital role in the filtration of data and services transmitted to the devices thereby resulting in reduced processing cost. In this paper, we have designed a middleware framework, which is based on ontology and is intended towards improving the quality of service discovery. The proposed middleware framework for context-aware service discovery makes use of contextual ontologies in order to permit the serving of semantically enriched contextual requests for services, without restricting the predefined contextual types or values.

***Keywords**:*
*Context-aware computing, Mobility, Ontology, Service Registry, Service Discovery, Middleware, UDDI Registry.*

## 1. Introduction

The advent of Mobile Computing has led to the development of a wide variety of applications in various contexts. An application in a mobile device is prone to drastic changes in physical conditions such as location, bandwidth, availability and economy besides a range of other logical conditions unlike applications on stationary devices. Context awareness is considered to be one of the most significant and beneficial features of mobile applications. The ability of applications to make use of the user's environmental information, so as to enable the execution of services of the user's interest is enhanced by context awareness [1]. Chen et al defined context as a set of environmental states and settings, that either determines the behavior of an application or a state in which an application event of a user's interest occurs [2]. A computing paradigm where applications discover and exploit the contextual information is known as context-aware computing [3]. Context information plays a significant role in the determination of service of relevance to the user from a distributed environment hosting numerous services at a time [4]. A huge variety of information defining physical objects, applications and users from myriad domains such as home domain, office domain, vehicle domain and more are encompassed in the context Information. Tao Gu et al proposed a two-layer hierarchical approach for designing context ontologies was proposed owing to the fact that the processing and maintenance of context knowledge in pervasive computing environments with the constraints on CPU speed and memory usage is highly unlikely [5]. Service discovery depends greatly on the contexts since they play a significant role in filtering out a subset of services relevant to the user's context. The discovery and effectuation of a service are carried out by the middleware Services Layer [6].The confinement of both mobile devices and contemporary wireless communication technology has necessitated the attentive management of resources. In the recent years, resource discovery for pervasive environments has attracted voluminous researchers. The term resource incorporates the services, devices, and components. Mark Weiser [7] suggested that resource discovery, is a paradigm that describes the interaction of discrete devices in the distributed environment and it acts as the nucleus of ubiquitous computing. They also have defined the resource discovery as the ability of an application to determine and make use of other resources of relevance in a particular context. Service discovery is a process in which the user contexts and service contexts are compared with each other so as to fetch services of relevance with the aid of context awareness. Even though a variety of methods are in existence for the context-awareness most of them are prone to difficulties in providing continuous services besides the quality of service being poor. The motivation of the proposed research is principally intended to develop an ontology based middleware framework for context aware discovery

Manuscript received March 5, 2009
Manuscript revised March 20, 2009

so as to provide continuous services of increased quality to the user. Traditionally, the term 'Ontology' was used to refer a subject of existence in philosophy whereas the literature defines ontology as a formal explicit description of concepts in a domain of discourse. Ontology presents a vocabulary for describing the knowledge about a domain and the depiction of particular circumstances in a domain [5]. Ontologies can be expressed with the aid of OWL (Web Ontology Language). The collective interpretation of certain domains, often considered as a set of entities, relations, functions, axioms and instances is known as Ontology [8]. OWL, an ontology language, is specially designed to express the ontological data semantically with the use of predefined classes and properties available with it. Some features in OWL that facilitate efficient representation of Ontologies include the ability to import other ontologies, assert or deny equivalence of individuals and classes and many more [9]. Lee and Helal [10] have addressed the limitations of the existing service discovery approaches namely, irrelevancy of returned services back to the requesting user. To over come the limitations, they have suggested to employee the context- awareness through service registries. Constantinescu et al. [11] dealt with service composition and discovery in their work. The directory response times were brought down to a considerable extent in their work with the aid of partial service matching and user defined ranking functions. In [12], [13], [14], Doulkeridis et al. have presented algorithms for constructing, searching, updating and merging a context-aware service directory in their approach. In [15], the authors capitalize on the theoretical foundation and describe in detail the overall architecture and implementation of a system that supports context-aware service discovery through enhanced service directories. A majority of existing works on context-awareness tend to fail in the continuous supply of relevant, accurate, and useful services to the user which eventually leads to the poor quality of service. Additionally, the utilization of resources poses a great challenge in the existing works. In this work, we have proposed a middleware framework for context aware service discovery based on ontologies to serve semantically enriched contextual requests for services without the restriction of predefined contextual types or values. The rest of the paper is organized as follows; Section 2 presents a brief review on some of the works done in context-aware service discovery. The system architecture of the proposed middleware framework is presented in Section 3. The proposed ontology based middleware framework for context-aware service discovery and the algorithms are discussed detailed in Section 4. The experimental results are presented in Section 5 and conclusions are summed up in Section 6.

## 2. Related Works

Dock horn Costa suggested that context awareness can be used to characterize the attributes and properties of an entity. The entity can be a user, a place, a physical or computational object [1]. Literature has witnessed many research efforts on the development of context-awareness toolkits supporting Web services provisioning, including HP'sCooltown project [16], Dey's The Context Toolkit [17], the CB-SeC framework [18], and the Gaia middleware [19]. These toolkits either provide functionalities to help service requesters obtain services based on their contexts or enable content adaptations according to requester's contextual information [20].

In recent years, many researchers [21, 6, 10] have introduced the concept, context aware service discovery, and they felt and incorporated the context aware concept in service discovery. Originally the context aware concept was utilized from the network field. In this technique, the web service descriptions have to be discovered and then they are matched against the user requirements to retrieve the relevant web services. Mostefaoui et al [6] have given a formal definition for service contexts to model a service's contextual information. Many researchers [21, 6, 10, 22] used Composite Capabilities/Preferences Profiles (CC/PP) as interoperable context representation to enable communication and negotiation of device capabilities and user preferences in terms of a service invocation. Later Zhang et al. [22 ] further proposed extensions to CC/PP to enable transformation descriptions between various receiving devices. The major disadvantage with CC/PP specification in these works are they have been implemented as repositories containing static context data of device capabilities and user preferences , through the construction of stored procedures that enable storing CC/PP files to a relational database. This makes the updating and deductive reasoning of sensed and dynamic user/device context data difficult to maintain.

Recently many researchers [23,20,24,25,25] incorporated the semantic, that is ontology based concept in the context aware service discovery process, by which the expressive limitations of the CC/PP specification are avoided ,and they expected that this approach can yield precise results to the requesting user. Christos Doulkeridis et al. [24] have proposed system architecture for service discovery, based on a novel context-aware service directory. They motivated the use of context for service discovery (especially when mobility of both consumer and provider is involved). The results shown that their approach reduces significantly the amount of transmitted data at the cost of a relatively small processing overhead, while at the same time the quality of the search also improved. These researchers utilized Ontology to describe contextual information such as location, time, device, preference and network context data, etc. Weili Han et al. [23] have proposed an approach for semantics-based matchmaking and named process-context aware matchmaking. Their process context aware matchmaking discovered the suitable service during web service composite modeling. During matchmaking, their approach utilized not only semantics of technical process but also that of business process of a registered service, thus for further improving the precision of matchmaking. Irene et al. [20] proposed an, context aware service oriented architecture (CA-SOA) for ubiquitous web service discovery and access based on service and requesters surrounding context. The CA-SOA consists of three different types of agents such as service agents, broker agents, and request agents which have been implemented to enhance the context oriented service description, publication, registration, discovery and access. Bettstetter and Renner [26] have performed a comparative study in various well known service discovery protocols namely, Service Location protocol(SLP), jini, Salutation, Universal Plug and Play(UpnP) , and the Bluetooth service discovery protocol, and have developed their own SLP which enhances the service discovery process by considering and assigning weights to static and dynamic contextual information associated with services. They integrate this selection mechanism into the Service Location Protocol (SLP). Jakob Bardram [27] has discussed the features of the Java Context-Awareness Framework (JCAF), such as its core design principles, its runtime infrastructure, and its programming API. Distinct features of JCAF are its support for distributed cooperating context services, its event-based middleware architecture, its support for a relaxed security model for authenticating context clients, and its support for semantic-free modeling of context information in Java.

Alessandra Toninelli et al. [25] have proposed a middleware level approach to support user-centric semantic service discovery. The approach exploits the context-awareness on basis of user/device/service profile metadata is called as AIDAS. It uses context awareness to provide the personalized views on services of interest besides supporting semantic based matchmaking.

Stefan Dietze et al. [28] have proposed an approach to support fuzzy, similarity-based matchmaking between real-world context characteristics and predefined SWS capability descriptions by incorporating semantic context information on a conceptual level into symbolic Semantic Web Services (SWS) representations utilizing a novel meta-model for Conceptual Situation Spaces (CSS). CSS enable the description of situation characteristics as members in geometrical vector spaces following the idea of Conceptual Spaces. Semantic similarity between situations is calculated in terms of their Euclidean distance within a CSS. Extending merely symbolic SWS descriptions with context information on a conceptual level through CSS enables similarity-based matchmaking between real-world situation characteristics and predefined resource representations as part of SWS descriptions. Thus Semantic similarities between situations are calculated in terms of their Euclidean distance within a CSS. By combining the semantic contextual information, these models can perform matches against both service and receiver's context semantically. The major shortcomings are observed from the literature that the requesting client are not provided with the current and relevant services based on their fast changing social and environmental context data, because the sensed , dynamic and deduced contextual information of the interacting entities are not being taken into consideration.

## 2.1 Overview of the proposed work

In this paper, we propose a middleware architecture based on distributed web service discovery, which is robust, reliable and efficient. In contrast to aforementioned related works, our approach on context aware service discovery stands out in three aspects: (i) This research work presents a new domain ontology based context model. (ii) We have developed a context interpreter which provides a dynamic and it deduces the contextual information for providing more useful services to the user. (iii) We employ a similarity/logic based semantic matching technique to enhance the relevancy and precision of returned set of services. Our middleware framework matches the service and receivers context data semantically with each other, to return relevant and useful services to the requesting user by making use dynamic and sensed contextual information.

The proposed architecture consists of various components such as the Service Registry, Middleware, Context Interpreter, Context Aware Manager, Ontology based context database and the Service Locating component. In the proposed approach, a keyword is supplied, by the requesting client through the mobile interface, to the middleware. The middleware handles the request by passing the keyword to the context interpreter that enriches the keyword with the current contextual information of the user. Those enriched keyword are passed on to the context aware manager, which retrieves the related set of keywords from the domain ontology based context model which are semantically compatible with the user requested keyword. The retrieved sets of keywords are sent to the service locating service which retrieves related services from the public service registries. The retrieved services are filtered again, using user context data and the historical usage pattern of the client.

## 3. System Architecture

This paper presents, in this section, a new middleware architecture which consists of The Service Registry, Middleware, Context Interpreter, Context Aware Manager, Ontology based context Database and the Service Locating component. The system architecture of the proposed middleware framework is depicted in Figure 1. The services are available in the various public service registries. A variety of service registries such as UDDI, ebXML, ISO11179, OASIS, eCo framework and other variants are available for publishing and inquiring services. A service registry is an important part of any service oriented architecture, which allows the developers to register their services, and end-users to locate useful services. In the proposed work the UDDI (Universal Description, Discovery and Integration) service registry is adopted since (i) it defines a standard for businesses to share information;(ii) it describes web services and their business, and allows others to discover the registered web services;(iii) it decides the information which 0is to be made public and that is to be kept private. In our framework the UDDI registry holds information about a business such as company name, contacts, and so on, and also holds both the descriptive and technical information about the service provided by the business. Such information consists of the URL for the web service and pointers to service contracts, and the technical specifications of the service. The mobile clients issue service request or query in the form of keywords to the middleware. The middleware will perform a sequence of operations to discover the services which are relevant to the changing contexts of the mobile clients. The middleware forwards the service request/keyword issued by the mobile clients to the context interpreter. Due to the mobility of clients, the surrounding environment i.e. context is exceptionally dynamic and volatile. The context interpreter enriches the service request or keyword with some information about the clients to ease the context-aware service discovery process. The context interpreter interprets the low level contextual data to a high level contextual information, which is used in the filtering of services. It forwards the enriched high level contextual information with the requested keyword to the context aware manager. The context-aware manager retrieves the keywords that are semantically related to the current service request/keyword by using a semantic searching on the domain based ontology database. The service request or keyword along with the retrieved keywords is again sent back to the middleware. The service locating service will retrieve the services related to the keywords from the service registries using JAXR. Finally, the discovered services are filtered based on the historical usage of the corresponding mobile client and send them back to the client.

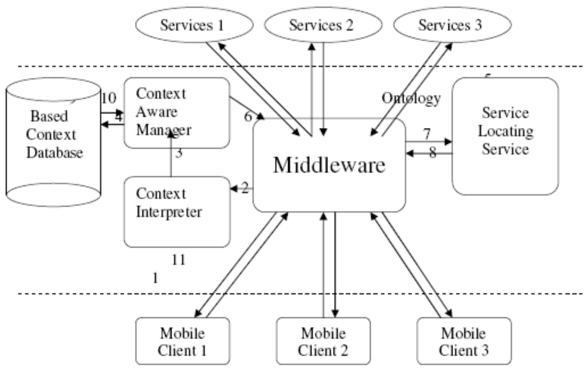


Fig. 1. Architecture of Middleware Framework for Context-aware Service Discovery

### 3.1 Middleware

The middleware is an important component in the proposed framework for context-aware service discovery. The middleware is the only interface between the mobile device user sand the context aware service discovery system. It handles all the functions related to the context-aware service discovery process. It directly interacts with mobile clients for receiving the request and serving the clients. The middleware receives the service request or keyword from the client, and passes it over to the context interpreter and to the other components applicable before delivering the service back to the requesting user. The context aware manager and the service locating service component provide the necessary inputs to the middleware to locate a desired service. All the services discovered from the service registry based on the associated keywords are filtered in the middleware using the historical usage context data of the clients. Finally the services appropriate to the service request and the current context of the mobile client are delivered to the requesting client.

### 3.2 Context Interpreter

The context interpreter is a component, which uses logic reasoning for contextual information processing. The context processing task includes: deriving high-level contexts from low-level contexts; querying context knowledge; maintaining the consistency of context knowledge and resolving context conflicts. It also acts as a context provider as it provides deduced contextual information. The context interpreter enriches the service request or keyword issued by the clients with some information about the clients to ease the context-aware service discovery process. The enriched context along with the keyword is then passed over to the context-aware manager. The enriched contextual information means, the information extracted from the current context of the requesting client by the interpreter.

## 3.3 Context-Aware Manager

The context-aware manager takes the responsibility of finding the keywords from the ontology based context model that are semantically associated with the service request or keyword issued by the clients. To achieve this, the context manager performs semantic search on the domain ontology based context model, where the associated category of keywords are stored. The context manager compares the enriched contextual information which is received from the context interpreter with the available user contextual information, to identify if there is any change in the available user context data. If there is a change, the manager performs the necessary updates on the available user contextual information. The manager forwards the service request or keyword along with the retrieved related set of keywords, to the middleware. The context aware manager also provides the middleware with beneficial currently available user contextual information, which the middleware can employ for filtering the services, which are returned from the service registries before being delivered back to the requesting user. Thus the context aware manager is supplied with input by the context interpreter and considers as relevant those services that conform both semantically and contextually to the user's request. The contextual reasoning between the user and service context data are done using advanced similarity based matching technique. In other words, relevant services are those that belong to the appropriate service category and at the same time return results that can be processed and used by the requesting user.

## 3.4 Ontology Based Context Model

The ontology based context data in our model contains ontologies related to various domains, which is used in the semantic searching process. The proposed discovery approach typically locates services within administratively defined domains, which might correspond to logical or physical environments. The ontology model contains a detailed conceptual schema about a domain, which is typically a hierarchical data structure containing all the relevant entities and their relationships within that domain. The domain-specific ontology used in our model is a collection of low-level ontologies which defines the details of general concepts and their properties in each sub-domain such as a vehicle domain or smart home domain as depicted in Figure 2. We have made use of the Subsumption relation between the classes available under a particular domain for developing our model. In the proposed technique, a class can subsume or be subsumed by other classes in a particular domain; a class subsumed by another is called a subclass of the subsuming class. For example, in Figure 2, in domain specific ontology, depicted for 'vehicles', the domain vehicle subsumes Four Wheeler. Since (necessarily) anything that is a member of the latter class is a member of the former. Our model makes use of the subsumption relation for representing each domain - specific ontology, which in turn makes use of the 'is-super class-of, is-subclass-of, and is-a subtype-of relationships for representing that domain.

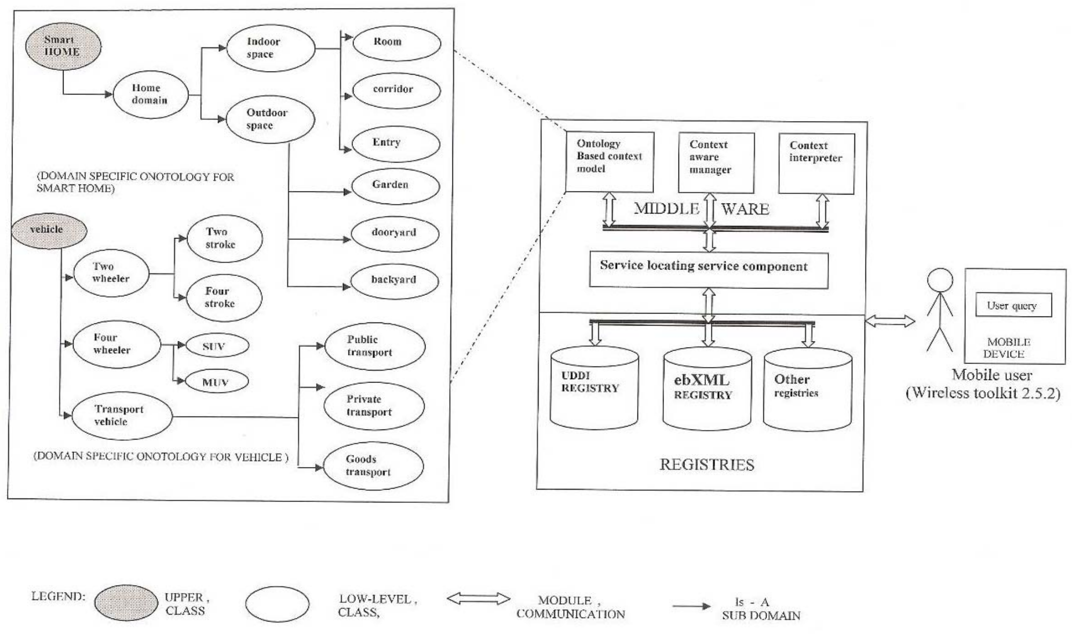


Fig. 2. A Partial definition of various domain specific ontologies in Ontology Context Model

For instance, the domain `vehicle' in Figure 2, the `vehicle' is considered as a `super class'; The next level that is `Four Wheeler' is considered as a `Sub-class' of vehicle, and the following levels are treated as `Sub-types' of the above classes. The system models the data as ontology based semantic descriptions which are independent of the programming language and the underlying operating system.

The proposed ontology model has two main advantages: (i) A hierarchical taxonomy is created with the use of `is-a' (subsumption) relationships which creates a tree-like structure that clearly depicts how the objects relate to one another in a particular domain; (ii) Due to the hierarchical taxonomy like structure of our domain model, it helps to determine the appropriate service categories that are related to the users request based on the use of domain ontology precisely. The two main advantages of our ontology model are, the usage of the 'is-a (subsumption) relationship' which has created a hierarchical taxonomy; a tree-like structure (or more generally said, a partial ordered set) that clearly depicts how objects relate to one another in a particular domain. Due to the hierarchical taxonomy like structure of our domain ontology model, it helps to determine the appropriate service categories that are related to the user's request based on the use of domain ontology precisely.

### 3.5 Service locating service

The Service Locating Service (SLS) component receives the client's service request and the related set of keywords from the middleware, and it retrieves the services related to the different keywords from the UDDI service registry based on the UDDI entry for the web service. The SLS retrieves the necessary services from the registry using JAXR(Java API for XML registry) The UDDI entry(*t*model) for a service contains the following information such as the name of the business, contact information, industry codes, product classification, description of the web service, requirements for rights to access the web service, and the technical reference to the interface and properties. The SLS supplies the services retrieved from the registry along with the service contextual information to the middleware. The middleware in return allows only the subset of filtered web services that are contextually compatible with the user's context, to be returned automatically and transparently back to the requesting client.

#### 3.5.1 JAXR

In typical service oriented architecture, a variety of service registries is generally available for publishing services. Access of this information is provided by means of a standard interface, namely JAXR. The JAXR specification is a general- purpose standardized Java API that allows the requesting client to access the two most dominant registries viz., UDDI and ebXML, while making it sufficiently general to support other registries in future. JAXR is a new API developed under the Java Community Process (JCP).The JAXR specification tries to unify the access of the various available public registries and probably the future registries by defining a new Java API. The JAXR achieves to abstract the particular underlying service registry and allows transparent access to service descriptions. Similar to JAXR, the context-aware information acts as an interface to the above mentioned registries using functions available with JAXR, and also it supports the filtering of retrieved services based on available user contextual information.

## 4. Ontology Based Middleware Framework Representation

The proposed approach is intended to retrieve the necessary services frequently specified by the user. For that we have proposed a semantic searching algorithm to get the keywords related to the service request specified by the users. Our system allows users to issue keyword based requests for services, so that a user can describe his necessities to the best of his knowledge.

$C_p$ → the context informations from the context provider

$I_p$ → Given informations

The user provided context information has been stored in the form of 2-d representation in a file called $C_p$ .

$$\{C_{p(ij)}\} << I_p[i]$$

$$C_p = \begin{bmatrix} \{K_1 \ \mathrm{K}_2 \ \mathrm{K}_3 \ \cdots\cdots\cdots \ \mathrm{K}_\mathrm{i}\} \\ \{K_1 \ \mathrm{K}_2 \ \mathrm{K}_3 \ \cdots\cdots\cdots\cdots\cdots \ \mathrm{K}_\mathrm{j}\} \\ \{K_1 \ \mathrm{K}_2 \ \mathrm{K}_3 \ \cdots \ \mathrm{K}_\mathrm{k}\} \\ \{K_1 \ \mathrm{K}_2 \ \mathrm{K}_3 \ \cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots \ \mathrm{K}_\mathrm{l}\} \\ \vdots \\ \{K_1 \ \mathrm{K}_2 \ \mathrm{K}_3 \ \cdots\cdots\cdots\cdots\cdots\cdots\cdots \mathrm{K}_\mathrm{m}\} \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \\ \vdots \\ y_n \end{bmatrix}$$

Where $k < i < j < m < l$ .

The 2-D representations of the context information are then converted into the 1-D representation and stored in $C_I$ .

$$C_I = [y_1 \ \ y_2 \ \ y_3 \ \ y_4 \ \cdots \ y_n]$$

$$N = |C_I| = \sum_{i=1}^{n} sizeof\left(C_p[i]\right)$$

The information's in $C_I$ are sorted on basis of their indexes and are stored in another file $SC_p$.

$SC_p$ → Sorted context informations

$$SC_p = sort(\Delta C_I)_{asc}$$

Here $\Delta C_I$ represents the information in $C_I$ along with their indexes. Further, the indexes have been separated and stored in another file $C_{IND}$.

$C_{IND}$ → Represents the indexes of the context informations

$$C_{IND} = \{x : p(x)\}$$

Where $p(x) = \Delta SC_p[i] \;; i > 1 > N$

Here $\Delta SC_p$ represents the index of the sorted context informations.

## 4.1 Semantic Searching

The desirable keyword $Kw$ has been given by the user. By using the binary search algorithm, the corresponding keyword has been searched out from the sorted contextual information's $SC_p$. The pseudo code of the searching technique is given as follows.

```
Set start = 0;
Set end = size of C_I;
Set mid = 0;
while (start <= end)
    $mid = (start + end)/2;$
    if ((C_I[mid] compare To key) < 0)
        $start = mid + 1;$
    else if ((C_I[mid] compare To key) > 0)
        $end = mid - 1;$
    else if ((C_I[mid] compare To key) == 0)
        $IU_k = C_{IND}[mid];$
    else
        keyword does not exist;
    end if
end while
```

Then the index of the corresponding requested keyword has been retrieved from C and stored in $IKw$ .Using the index value of the user requested keyword obtained from $C_{IND}$, the keywords corresponding to that index value have been retrieved from the context provider and stored in $C_k$.

$IU_k$ → Index of the user keyword.

$$IU_k = C_{IND}[IKw]$$

$$C_k = C_p[IU_k]$$

Now $C_k$ contains the selected row i.e., $C_k = \{K_1\ K_2 \cdots K_n\}$

The service names and the descriptions of each keyword present in the $C_k$ have been given by UDDI registry $U_R$ of different environments. The selected keywords will be sent to the URLs retrieved from the UDDI registry as parameters. The servers corresponding to the URL will respond the service name and description of the keywords. The Cartesian product of the selected keywords and the URLs gives the service names and description of each keyword.

$$\{K_1, K_2, \cdots\cdots, K_n\} \times \{url_1, url_2, \cdots\cdots, url_n\} = \begin{Bmatrix} S_{11} & S_{12} & \cdots\cdots & S_{1n} \\ S_{21} & S_{22} & \cdots\cdots & S_{2n} \\ \vdots & & & \\ S_{n1} & S_{n2} & \cdots\cdots & S_{nn} \end{Bmatrix}$$

$S_N(C_{k[i]})$ → service names of each keywords in $C_k$.

$$U_R \rightarrow S_N(C_k[i])$$

$U_{IP}$ → Vector of inquire and publish URLs

// load function to load inquire and publish URLs
// getServiceName function returns service names from $U_{IP}$ as vector $T_1$
// getServiceDescription function returns service descriptions from $U_{IP}$ as vector $T_2$

```
for each (i) value in C_K
    for each (j) value in U_IP
        load (U_IP[j]);
        $T_1 = getServicename(C_K[i]);$
        $T_2 = getServiceDescription(C_K[i]);$
        for each k value of T_1
            $S_N[count] = T_1[k];$
            $S_D[count] = T_2[k];$
            count++;
        end for
    end for
end for
```

### 4.2 Filtered Keyword Set Rule

In this filtered keyword set rule, the returned services from the service registries are being filtered out by the users' already used services and the services chosen from the UDDI registry. The services which are common to both the services and those services can be provided to the users' mobile continually.

$U_{SER}$ → {The services already used by the user}

$F_K$ → Filtered Service keywords

$$F_K = S_N(C_k[i]) \cap U_{SER}$$

The pseudo code for this filtered keyword set rule is given as follows**,**

```
for each (i) value in $U_{SER}$
    for each (j) value in $S_N$
        if $(U_{SER}[i] == S_N[j])$ then
            $S_N[i] \rightarrow$ provider to user;
            $S_D[i] \rightarrow$ provider to user;
            break;
        end if
    end for
end for
```

The overall interactions among the various middleware components in the proposed framework are diagrammatically represented in Figure 3.

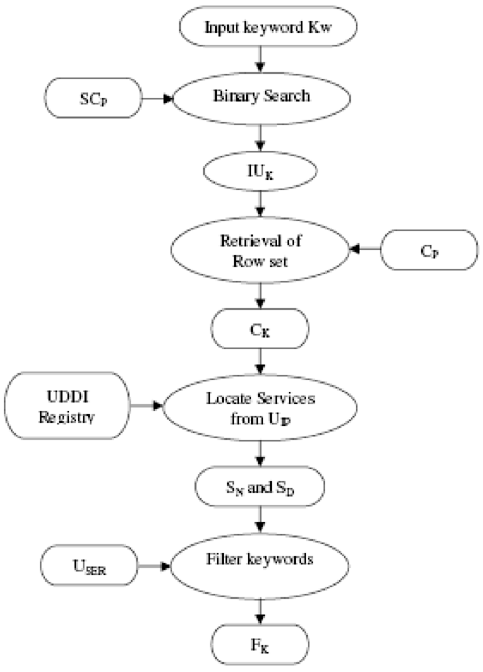


Fig. 3. Flow diagram of the proposed approach

## 5. Implementation & Results

To implement the concepts discussed in the previous sections, the following software specifications are used: the wireless tool kit 2.5.2 for CDLC is used to simulate the mobile clients; the Apache Tomcat 6.0.14 is used as the web server. The JAXR (Java API for XML Registries) which is available with the Java Web Services Developer Pack (JWSDP 2.0) is used for the retrieval of the services from the UDDI registry; the database MySQL 5.0 is used to store the users' profile and the historical usage patterns. Initially the service request is given by the mobile clients to the middleware using the mobile device emulator. The service request are keyword received from the user is forward to the context interpreter, which encriches the keyword with the details available in the database and the enriched keyword is sent to the contextaware manager. The semantic search algorithm is used to retrieve the keywords associated with the users service request from the ontology based context model, and the keywords related to the service request is supplied to the SLS through the middleware. The services associated with the keywords are extracted from the UDDI registry, and the discovered services are filtered using the historical usage pattern of the clients available with the database. Finally, the filtered services are delivered back to the requesting client.

In order to demonstrate the efficiency of the proposed discovery model, the keyword "university" is given as request to the system. The related set of keywords obtained form the ontology model which are associated with the users request are presented in Table 1. The services discovered from the UDDI registries for the obtained related keywords are given in Table 2. The services that are filtered form the discovered services based on the historical usage pattern of the clients are presented in Table 3. The snapshot depicted in Figure 3 represents the client user interface (UI) of the proposed middleware framework.

Table 1: Client's service request and the related keywords

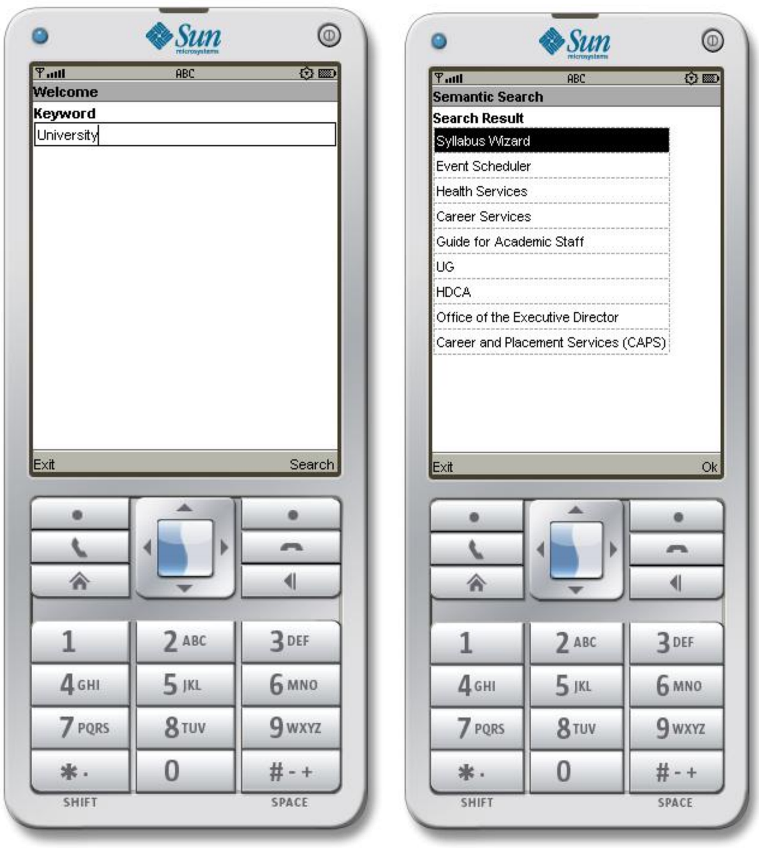
| **Service Request** | **Related Keywords** |
|---|---|
| **University** | Professor<br>Lecturers<br>Department<br>Courses offered<br>Library<br>Students |

Table 2: Discovered Services for the keywords

| Keywords | Service Names |
|---|---|
| Professor | Class Schedule<br>Syllabus Wizard<br>Group Schedules<br>Event Scheduler |
| Lectures | |
| Department | Counseling Center<br>Career Services<br>Health Services<br>Library<br>Web Development & New Media |
| Courses Offered | UG<br>PG<br>Diploma<br>DCA<br>PGDCA<br>HDCA |
| Library | Partnership Institutions<br>External Readers<br>Guide for Academic Staff |
| Students | Office of the Executive Director<br>Services for Students<br>Career and Placement Services (CAPS)<br>Chaplaincy<br>Counseling<br>First Peoples' House<br>First-Year Office<br>Library |

Table 3: Filtered Services

| Filtered Services |
|---|
| Syllabus Wizard |
| Event Scheduler |
| Health Services |
| Career Services |
| Guide for Academic Staff |
| UG |
| HDCA |
| Office of the Executive Director |
| Career and Placement Services (CAPS) |



Welcome
Keyword
University
Exit Search

Semantic Search
Search Result
Syllabus Wizard
Event Scheduler
Health Services
Career Services
Guide for Academic Staff
UG
HDCA
Office of the Executive Director
Career and Placement Services (CAPS)
Exit Ok

Fig.4. The keyword and the filtered services displayed in the mobile device emulator

## 6. Conclusion

Due to the growth in the fields of mobile computing and networking, there is a stipulation for certain advances in context- awareness. Context plays a vital role in the filtration of data and services transmitted to the mobile devices thereby resulting in reduced processing cost. In this paper, we have proposed a middleware framework for context aware service discovery. The services have been discovered from the service registries and filtered according to the context of user and delivered to the client. In our work, the user has attained the appropriate services when they travel from one place to another. We have used contextual ontologies in our work for the semantic searching of related keywords. The advantage of the proposed approach was stability in providing the services .The future work involves solving the QoS issues of the service discovery framework.

## References

[1] Dock horn Costa, P., “Towards a Services Platform for Context-Aware Applications”, Master Thesis, University of Twente, Enschede, Netherlands, August 2003.

[2] Chen, G., and Kotz, D., “A survey of context-aware mobile computing research”, Technical Report: TR2000-381, Dartmouth College, Computer Science, Hanover, NH, November 2000.

[3] Lopes, A., Botelho, L.M., “SEA: a Semantic Web Services Context-aware Execution Agent”. In Proceedings of the AAAI Fall Symposium on Agents and the Semantic Web, Arlington, Virginia, USA, $4^{th}$-$6^{th}$ November 2005.

[4] Edwards, W.K., Bellotti, V., Dey, A.K., Newman, M.W., “The Challenges of User-Centered Design and Evaluation for Infrastructure”, In Proceedings of the SIGCHI conference on Human factors in computing systems, Ft. Lauderdale, Florida, USA, pp: 297 - 304, 2003.

[5] Tao Gu, Hung K Pung, Da Q Zhang, “A service-oriented middleware for building context-0aware services”, Journal of Network and Computer Applications, Volume 28, Issue 1, pp: 1 18, January 2005.

[6] Mostefaoui, S. K. and B. Hirsbrunner, “Towards a Context Based Service Composition Framework”, In Proceedings of the 1st International Conference on Web Services (ICWS’03), Las Vegas, USA, June 2003.

[7] Mark Weiser, “The Computer for the 21st Century”, ACM SIGMOBILE Mobile Computing and Communications Review, Volume 3, Issue 3, pp: 3 - 11, July 1999.

[8] T Gu, XH Wang, HK Pung, DQ Zhang, “An Ontology-based Context Model in Intelligent Environments”, In Proceedings of Communication Networks and Distributed Systems Modeling and Simulation Conference, pp: 270--275, 2004.[26] J. Zhang, L.-J. Zhang, F. Quek, and J.-Y. Chung, “A Service-Oriented Multimedia Componentization Model”, *International Journal of Web Services Research* (JWSR), Vol. 2, No. 1, Jan-Mar. 2005, pp. 54-76.

[9] H. Chen, T. Finin, A. Joshi, “An Ontology for Context-Aware Pervasive Computing Environments”, The Knowledge Engineering Review, Volume 18, Issue 3, pp: 197 - 207, September 2003.

[10] Lee, C. and S. Helal, “Context Attributes: An Approach to Enable Context-awareness for Service Discovery”, Proceedings of 2003 Symposium on Applications and the Internet, pp: 22 – 30, 27-31 Jan. 2003.

[11] Constantinescu, I., W. Binder and B. Faltings, “An Extensible Directory Enabling Efficient Semantic Web Service Integration”, Lecture Notes in Computer Science, Springer Berlin / Heidelberg, Volume 3298, pp: 605-619, 2004, ISBN: 978-3-540-23798-3.

[12] Doulkeridis, C., E. Valavanis and M. Vazirgiannis, “Towards a Context-Aware Service Directory”, In the Proceedings of the 4th VLDB Workshop on Technologies on E-Services (TES’03), pp: 54--65, 2003.

[13] Doulkeridis, C. and M. Vazirgiannis, “Querying and Updating a Context-Aware Service Directory in Mobile Environments”, Proceedings. IEEE/WIC/ACM International Conference on Web Intelligence, pp: 562 - 565, 20-24 Sept. 2004.

[14] Doulkeridis, C. and M. Vazirgiannis, “Updating a Context-Aware Service Directory for M-services”, in: 3rd Hellenic Data Management Symposium (HDMS’04), Athens, Greece, June 28-29, 2004.

[15] C. Doulkeridis, N. Loutas, and M. Vazirgiannis, "A system architecture for context-aware service discovery", Electronic Notes in Theoretical Computer Science, Volume 146, Issue 1, pp: 101-116, 24 January 2006.

[16] The CoolTown Project at HP Labs, (http://www.cooltown.hp.com)

[17] Anind K. Dey & Gregory D. Abowd, The Context Toolkit: Aiding the Development of context-Aware Applications, in; Proceedings of the 1stInternational Workshop on Managing Interactions in Smart Environments (MANSE '99), 1999, pp. 114-128

[18] Mostefaoui, S. K. and B. Hirsbrunner, Towards a Context Based Service Composition Framework, in: Proceedings of the 1st International Conference on Web Services (ICWS’03),2003.

[19] Manuel Román, Christopher Hess, Renato Cerqueira, Gaia: A Middleware Infrastructure to Enable Active Spaces,in: 9th SIGOPS European Workshop, Kolding, Denmark, 2000.

[20] Irene Y.L. Chen; Stephen J.H. Yang; Jia Zhang, “Ubiquitous provision of context awareweb services”, In proceedings of IEEE International conference on Services Computing, pp: 60-68, September 2006.

[21] T. Lemlouma and N. Layaida, “The Negotiation of Multimedia Content Services in Heterogeneous Environments,” Proc. of the 8th International Conference on Multimedia Modeling (MMM 2001), Amsterdam, the Netherlands, Nov. 5-7, 2001, pp. 187-206.

[22] Composite Capabilities/Preference Profiles (CC/PP), http: // www. w3. org/ Mobile/ CCPP/ .

[23] Weili Han, Xingdong shi and Ronghua Chen, “Process-context aware matchmaking for web services composition”, Journal of Network and Computer Applications, vol. 31, no. 4, pp. 559 – 576, November 2008.

[24] C. Doulkeridis, N. Loutas, and M. Vazirgiannis, “A system architecture for context-aware service discovery”, In Proceedings of International Workshop on Context for Web Services (CWS'05), pp. 101-116, Paris, France, July 5, 2005.

[25]Alessandra Toninelli, Antonio Corradi and Rebecca Montanari, “Semantic- based discovery to support mobile context-aware service access”, Computer Communications, Volume 31, Issue 5, pp: 935-949, March 2008.

[26] Bettstetter, C. and Renner, C., “A comparison of service discovery protocols and implementation of the service location protocol”, In Proceedings of the 6th EUNICE Open European Summer School: Innovative Internet Applications, 2000.

[27] Jakob E.Bardram, “The java context awareness framework: A service Infrastructure and programming framework for context-aware applications”, Lecture notes in computer science, vol. 3468, pp: 98 to 115, 2005.

[28] Stefan Dietze, Alessio Gugliotta and john Domingue, “Towards context-Aware semantic web service Discovery through Conceptual situation spaces”, In proceedings of ACM International Conference on context enabled source and service selection, integration and adaptation, vol. 292, no. 6, 2008.