Towards a Two-tier Context-based Service Discovery Framework for Mobile Web Services

1Dominic Afuro Egbe, 2Bethel MutangaMurimo

1Department of Information Technology, Durban University of Technology, South Africa
Email: domafuro@gmail.com
2Department of Information and Communication Technology, Mangosuthu University of Technology, Durban, South Africa
Email: mutangamb@mut.ac.za

Abstract
Mobile web services (MWS) enables the realization of a decentralized web service provisioning platform that is powered mainly by mobile devices. This paradigm has, in the last decade, gained research attention because of the growing demands for e-services. However, establishing such an e-service platform faces the fundamental and technical challenge of dynamic context change, which can affect service discovery. Basically, in a MWS scenario, mobile devices are enabled not only as conventional web service requesters but as providers too. Furthermore, mobile environments are prone to changing context. The challenge therefore, is that discovering the right service provider and service are, at any point in time context dependent. This dependence can create a problem where, based on current context, a wrong service provider or service may be chosen or invoked respectively. Consequently, discovery frameworks must be able to support the use of context information to enhance the efficiency of discovering service providers and services in dynamic environments. This paper proposes a two-layer service discovery framework that relies on the resource context of devices to discover the most appropriate service provider and provided service.

Keywords: Device context; mobile web service; relevant service; service discovery; Ubiquitous

1.Introduction
Advancements in the domain of mobile data services has, in part, inspired the current notion of MWS [1]. In the last decade, the MWS paradigm has significantly gained research interests. The paradigm seeks to establish the use of resource-constrained mobile devices as both web service clients and providers while still maintaining the de-facto web services architecture. This interest is being attributed to the strong drive in recent years to achieve ubiquitous computing - a trend that has led to the proliferation of e-markets, supported by the surge in number of fascinating handheld devices, advanced wireless communication technologies and an increasing mobile subscriber base.

For instance, the Juniper Research had forecasted that consumer and enterprise market for cloud-based mobile applications would rise to $9.5 billion in 2014 [2]. In the same vein, the global wireless subscribers index as at 2008 stood at 2 billion[3]. In fact, by the projection in [4], it is expected that by 2020, mobile devices, MWS and wireless communications will take centre stage in all aspects of live.

Owing to the prospect presented by the MWS model, research focus in this domain centres on standardizing MWS provisioning. Such research efforts are reflected in several proposals aimed at developing frameworks that can support decentralized web service hosting and context-based and resource-efficient service discovery [4], [5].

Generally, the insight into the above research direction basically advocates a change in the architecture of cloud and mobile cloud computing (MCC) in order to address challenges fundamental to mobile service consumers[6]. These challenges include, but not limited to lack of
or intermittent connectivity, limited bandwidth and high-latency, which results in high energy consumption. Consequently, the emergence in recent years of MCC variances like Ad-hoc Mobile Cloud (AMC) [7], [8], MobiCloud[6], Cloudlets, [9]etc. is clearly driven by the above insight. Therefore, to create a functional model for MWS that can fit into these cloud variances, the last decade has seen a number of MWS discovery frameworks proposed in the literature[1]. In general, these frameworks have converged research interest centred on easing resource burden and improving service discovery via context-awareness.

This paper focuses on the latter research interest and is motivated by the AMC paradigm, which presents an ideal infrastructure-less platform for offering MWS[10]. However, while current MWS discovery frameworks have mainly concentrated only on the client device’s context (local context), we argue that achieving efficient MWS discovery requires a more comprehensive device context approach. That is, considering the context of both the client device and the device providing service. The above argument is substantiated by the fact that while a relevant service may be discovered, the context of the service providing device can greatly affect quality of service or the entire discovery process.

For instance, a typical MWS provisioning scenario requires that mobile devices be enabled not only as conventional web service requesters but even as providers. This requirement introduces a two dimensional but interwoven challenge. First, discovering a MWS entails first discovering the right service provider. In the context of this paper, a right service provider is a device (mobile host) that offers the needed service and has the capability in terms of its current context to provide the required service at the time of request. Second, the key resources of mobile devices (battery and memory) are dynamic in nature. Meaning, these resources can change state unanticipated during operation as described in[10]. This inherent nature has phenomenal impact on discovering relevant services or right service providers. Example, a mobile device with a hypothetical battery state, $b_k$, capable of providing a service at a given time, say $t_j$, might be incapable of doing so again at a later time $t_{j+1}$, when the state of the device’s battery would have changed from $b_k$ to $b_{k+1}$. In the same vein, a client device may not be able to consume an offered service if the service fails to fulfil the device’s current resource capabilities[11], [12].

Therefore, in principle, a mobile device or node is a composite service – a service that contains a service or services. This consideration raises the need for a service discovery framework that implements context-based two-dimensional approach to MWS service discovery in AMC domain. The framework is aimed at supporting the dual operations of: i) discovery of a node-as-a-service (NaaS) and ii) discovery of the actual web service.

This paper is a response to this need.

2. Related Works

Although the field of MWS is emergent and most research efforts in the domain still at an early stage, current literature reports several state-of-the-art proposals directed at achieving efficient MWS discovery. In all, existing works can be aligned along two research interests, namely: (i) Solving the challenge of resource-burden that may be created by computationally-intensive operations; (ii) Solving the challenge of the effects of context change on discovering relevant services.

In this section, some of the notable works that attempted to address the second challenge are reviewed. This concept of context awareness has been wide employed by discovery mechanisms in mobile environments. Therefore, the work in [8] suggested the need to expand the scope of device context to include resource context in order to consider service relevance as a function of meeting both client requirements and device resource capabilities.

The research work reported in [13] presented an idea of context-based service discovery called device-awareness. The authors’ proposal extended the WSDL to make it more expressive to support the incorporation of device capabilities into MWS description. However, their focus did not include device context, in relation to hardware resources such as memory and battery.

In[14], a Cloud-assisted MWS discovery framework was proposed in which the concept of discovery-as-a-service (DaaS) was introduced into the conventional Cloud distribution model.
The work dealt with the problem of using various contexts information to tailor and rank MWS. Nonetheless, being a Cloud-assisted discovery mechanism made the challenge posed by the context of the device providing service a nonissue. In the same vein, the authors in [15] presented a Cloud-based framework for discovering MWS in mobile computing environments. In this framework, a keyword-based discovery process enhanced by semantic techniques was adopted. However, the only considered client device’s context, namely device profile and environment context. Also, resource context was not investigated.

In our earlier work [10], emphasis was laid on determining service relevance as a function of resource capability of mobile devices that act as Cloud providers. Again, this work did not offer a comprehensive MWS discovery framework that considers a mobile device Cloud provider as a composite service.

Another research reported in [16], presented a context-based service discovery algorithm for mobile MWS. Although all the works reviewed offered MWS discovery solutions that utilize one form of context information or other, none considered a context-based framework that handled both node and service discovery.

Towards creating ad hoc network, several researches have employed Wi-Fi Direct network technologies for supporting multi-hop device-to-device communication[17], [18].

3. Mobile Web Service Discovery Scenario

Conceptually, in a typical MWS provisioning and discovering scenario, the mobile device (node) providing a service is also considered to be a service. That is, a service providing node is a composite service – it contains a service or services. By this concept, the service providing device must be discovered first before the service it offers. This concept, as discussed in section I makes imperative for MWS discovery process to be structured into two levels:

3.1 Node Discovery Stage

At this level, we introduce the idea of discovering a node-as-a-service (NaaS). This requires that the discovery mechanism be able to search for the device that offers the required service and has the capability in terms of its current resource context to provide the needed service at the time of request.

3.2 Service Discovery Stage

This level handles the discovery of services within a device after a device has been discovered. This level of discovery is challenging due to unpredictable change in resource context. Therefore, it is extremely useful to enhance service discovery using context information. At this service discovery level, resource context is extracted from the client device and incorporated to search queries. Incorporating context information helps to tailor services based on how best the services match the resource capability of the client device.
Based on the above structure, a typical MWS discovery scenario can be painted diagrammatically as depicted in Fig. 1. This discovery scenario becomes highly imperative in AMC environment where mobile devices exploit a self-organizing network to support direct communication between each other in order to act as cloud providers and or clients[10].


Based on the MWS paradigm and the concept of composite service, a two-tier context-based framework was proposed for service discovery in AMC environment. The architectural illustration of the proposed framework is given in Fig. 2, showing the components that help to establish effective MWS discovery while utilizing device context.
Basically, every participating device or node has all the supporting components of the framework to be able to function both as client or service provider. However, for clarity in the illustration, the architecture presented in Fig. 2 designates one device as playing the client role while the other takes up the service provider role. Consequently, in each case, only components that support the respective role are in the architecture.

Essentially, there are seven components in the framework:

4.1 Service Request Controller

Service Request Controller (SRC) coordinates and analyzes users’ requests in addition to serving as user-client interaction interface. Therefore, SRC consists of two sub components –
Service Request Interface (SRI) and Request Analyzer (RA). The SRC component utilizes XML or JSON string (a data format supported by Wi-Fi Direct) to build requesters that are forwarded to the discovery engine.

4.2 Device Monitor

Device Monitor (DM) tracks device resource status - current battery level and available memory, to provide relevant data needed for context-based service discovery. This operation is achieved through the Device Resource Monitor (DRM) and Context Manager (CM) sub-components. While DRM helps to poll device resource context data by invoking the FreeMemory() and getBatteryLevel() methods provided in the Android utility functions, CM is responsible for making context information available for the discovery process by interacting with DRM to retrieve these dynamic context.

4.3 Discovery Engine

Discovery Engine (DE) performs context-based node and service discovery operations. To facilitate this process, the DE component obtains context information from the Device Monitor. The DE employs Wi-Fi Direct Device Discovery and Service Discovery methods, defined in the WifiP2pManager class of Wi-Fi P2P APIs[19]. Therefore, DE executes the algorithms that implement the discovery of nodes and services.

4.3.1 Node Discovery Procedure

Definition: Supposed C is a client device, \( MWS = \{mws_1, \ldots, mws_n\} \) is a set of MWS offered by a service providing node, and \( P = \{p_1, \ldots, p_i\} \) is a set of service providing nodes.

Definition 3.2: Supposed \( X = \{x_1, \ldots, x_t\} \) is a set of resource status weights, the Resource Capacity Ratio is defined in equation (1):

\[
R_{cr} = \frac{\sum_i^k x_i}{k} \text{..........................................................(1)}
\]

and the Optimal Node, denoted by \( N \), is the node that offers the kind of service requested and has the highest Resource Capacity Ratio.

The node selection solution can be stated as follows: Given a client, C and a set of service providing nodes, P, select the Optimal Node, N from the candidate set of participating service providers (P) to offer the requested service to the client so that its resources are not depleted while providing a service.

4.3.2 Service Discovery Procedure

Definition: Supposed \( DSF = \{dsf_1, \ldots, dsf_j\} \) is a set of devices supported features, \( MSF = \{msf_1, \ldots, msf_f\} \) is a set of MWS supported features.

WSDL-M [11] is adopted, which enables the inclusion of service and device features (additional context information) into MWS descriptions as non-functional parameters.

To determine a match between requested services and capability of the client device, DE compares the service features of each discovered service to the device supported features of the client device, using the Normalized Google Similarity Distance function \( f \) [20].

For instance, for each retrieved MWS, it follows that:

\[
MWS_{recovered} = f\{msf_q, dsf_j\} \text{..........................................................(2)}
\]
The function $f$ returns a value in the range $x \geq 0.5$ and $x \leq 1$ if there is a match otherwise $x \leq 0.5$. From equation (2) the degree-of-match is derived thus:

$$dm = \sum f(msf_y, dsf_j),$$

which is a numeric value that indicates whether a returned service has a weak, strong or zero match to the requested service.

### 4.4 Request Listener

Request Listener (RL) listens and responds to incoming service requests broadcast by potential clients within the network. RL component utilizes the broadcast receiver method provided in the Wi-Fi P2P framework to send and receive intents to and from other participating nodes. When RL picks up service request intent, it checks whether the requested service is listed in the service provider’s advertisement register and then interacts with the DE component for appropriate action.

### 4.5 Service Advertisement

The Service Advertisement acts as a quick access register that hold names of the services offered by each service providing node. This register is automatically updated as new MWS are deployed to the mobile host. The resource capacity weight of each device providing services is also held in this register. This weight is used to rate the capability of a device to provide service at any point in time.

### 4.6 SQLite Database

SQLite is an embedded, lightweight database with self-contained library with no server component, and small code footprint. It has limited resource requirements, intended to support resource-constrained devices. This component is used to host MWS and store other forms of static context information.

### 5. Experiment and Evaluation

A preliminary prototype system was implemented to test the applicability and benefits of the proposed framework. The prototype either executes in client or provider mode and run as a service at the background of the device while utilizing Wi-Fi Direct technology. The implementation was done with Java Language on Android SDK v21. Fig. 3 depicts the service request interface of the framework.

A live test-bed experiment was conducted consisting of twenty (20) Android mobile devices, each running the prototype App. The default mode of one of the devices was set to “client”, while the remaining nineteen (19) devices ran in “provider mode”. This setup was meant to enable the
device designated as client to execute service discovery in the pool of other devices acting as servicer providers.

Each service providing device had at least fifteen (15) MWS. The MWS were represented by service descriptions obtained from online sources and modified based on WSDL-M stored in the SQLite database.

Two set of experiments were conducted. The first investigates the impact of context on node discovery efficiency. This experiment was repeated fifteen (15) times, and at each run of the experiment, two (2) service requests were executed independently – one using resource context and the other without context with the results shown in Table 1 and 2.

<table>
<thead>
<tr>
<th>Resource capacity ration</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>10</th>
<th>11</th>
<th>12</th>
<th>13</th>
<th>14</th>
<th>15</th>
</tr>
</thead>
<tbody>
<tr>
<td>Context-based</td>
<td>64.5</td>
<td>69.5</td>
<td>78</td>
<td>83</td>
<td>77</td>
<td>64</td>
<td>88.5</td>
<td>76</td>
<td>57</td>
<td>71</td>
<td>71</td>
<td>66</td>
<td>97</td>
<td>80</td>
</tr>
<tr>
<td>Non-context-based</td>
<td>64.5</td>
<td>29</td>
<td>28.5</td>
<td>77</td>
<td>66</td>
<td>55.5</td>
<td>45</td>
<td>21.5</td>
<td>79</td>
<td>55.5</td>
<td>44</td>
<td>51</td>
<td>80</td>
<td>49</td>
</tr>
</tbody>
</table>

Table 2: Degree of Match of Retrieved Services

<table>
<thead>
<tr>
<th>Returned Serviced</th>
<th>Request 1: Device 5</th>
<th>Request 2: Device 15</th>
<th>Request 3: Device 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Strong match</td>
<td>11</td>
<td>7</td>
<td>10</td>
</tr>
<tr>
<td>Weak match</td>
<td>3</td>
<td>5</td>
<td>6</td>
</tr>
<tr>
<td>Zero match</td>
<td>0.02</td>
<td>0.03</td>
<td>0.01</td>
</tr>
</tbody>
</table>

The experiment validated the framework’s ability to extract and use context information to achieve effective service discovery in AMC. The demonstration of result in Table 1 and 2 as shown in Fig. 4 and 5 indicates the positive effect of context information on service discovery.
For instance, without context information, as depicted in Fig. 4, there was a high probability of selecting an inadequate service providing with regards to resource capability. However, the reverse was the case when context was employed since all selected nodes had very high resource capability ratio. The second experiment examined the effect of context on service discovery efficiency based on the concept of degree-of-match discussed in section IV. Results obtained indicated that the prototype recorded superior performance with regards to service retrieval. As shown in Fig. 5, out of three requests to three different service providing nodes, a total 42 MWS
were returned. Out of these, 28 had strong match while 14 were of weak match and none was a zero-match service.

6. Conclusion

The concept of mobile web services is an emergent e-services paradigm that has, in recent years drawn research attention to the challenge of creating a decentralized and infrastructure-less e-service platform that imports the concepts of mobile cloud computing. To address this challenge, research efforts has also been focused on formulating discovery mechanisms that takes cognizance of the inherent nature of mobile environments, such as dynamic context. In this paper, a two-tier context-based MWS discovery framework was presented. A preliminary prototype was developed to evaluate the framework with regards to impact of context information on efficient service discovery. Though the prototype development is ongoing, initial results demonstrated that using context led to efficiently discovering services that match client capabilities.

References


Authors Biography

**Dominic Afuro Egbe**, holds B.ScHonours. degree in Computer Science and Information Systems from University of Venda, South Africa and research Master degree in Computer Science from University of Zululand, South Africa. His research interest includes ICT for Development and Context-aware Service Discovery in Mobile/Ad-hoc Mobile Cloud.

**Bethel Mutanga Murimo**, holds a Master degree in Computer Science from the University of Zululand, South Africa. He is currently doing his Ph.D with the same institution. His research interest includes wireless ad-hoc networks, Mobile Cloud Computing and the development of low cost technologies for Africa.