Preference Querying for Context-Aware Location Based Services

Tin Maung, Ni Lar Thein
University of Computer Studies, Yangon, Myanmar
tinmaung84@gmail.com, nilarthein@gmail.com

Abstract

This paper presents system architecture of a Context and Preference-Aware Location-based Services system (CPALBS, for short), that delivers personalized services to its customers based on the user preference query. CPALBS goes beyond the traditional scheme of "one size fits all" of existing location-aware database systems. Instead, CPALBS tailors its functionalities and services based on the preference of each customer. One of services provided by CPALBS includes a hotel finder application in which CPALBS does not base its choice of hotels solely on the user desired location. Instead, CPALBS will base its choice on both the user desired location and preferences (e.g., user dietary restriction).In this system, user requests the available services via mobile device and the system returns the optimal answers related to user requested location range and preferences. Within the framework of CPALBS, a proximity detection algorithm based on symmetry approach is proposed to find user desired location range and a simple tree matching method is proposed by analyzing the structure and content of query tree to efficient match and extract user preferences query within specified location range. The proposed system uses label order rooted tree data structure to efficiently match user preference query based on the user preferences.
Keywords-Location-Based Services, tree matching, Simple Tree Matching, preference querying, location range query

I. INTRODUCTION

The main promise of location-based services is to provide services to customers based on the knowledge of their locations. Examples of these services include real-time traffic information, digital map services which are delivered to mobile terminals according to user's location to minimize data transmission, dynamic guidance services are provided according to the users' location and current traffic condition; requesting the nearest business or service (e.g., the nearest restaurant or hotel) and location based advertising ("Send e-coupons to all cars that are within two miles of my gas station").

Unfortunately the current state-of-the-art location based services are rigid as they cannot make good use of contextual information. Services are provided at inappropriate time without considering user's intention and changing environment. Also services are rigid as processing completely isolates various forms of user "preferences" and/or "context". For example, in a hotel finder application users actually want to find the "best" hotel according to their current preferences and context. Existing location-based query processors reduce the meaning of "best" to be only the "closest" hotel. Any query processing that produces results based on preference and/or context is applied after the location-based database operations. In other words, preference and context are considered afterthought problems in terms of query processing. The rigidness of such an approach is due to two main reasons: (1) the lack of personalized customer services. For example, if two persons are asking the same query in the same location, they will get the same answer, even if their personal preferences are different. (2) The lack of context awareness as the only considered context is the user location, while other kinds of context (e.g., personal preference) are completely ignored.

This paper aims to raise the challenges and provide research directions to enable practical realization of context and preference-aware location-based services. The main idea is to embed various forms of preferences in core processing of location-based queries. To this end, the proposed system is not aiming to define new location-based queries; instead, the system aims to personalize the answer of existing location-based queries. As the query answer may be returned to the users on their mobile devices with limited screen capabilities, it is essential to enhance the quality of the answer and limit the answer only information that are major interest to the users according to their preferences (e.g., dietary restriction, range of price, and acceptable rating for hotel services) and user specified location range.

Towards the goal of realizing a context and preference-aware location-based services, the system architecture of a context and preference-aware location-based Services system (CPALBS, for short) that delivers personalized services to its customers based on the user desired location and preference query. CPALBS tailors its functionalities and services based on the preferences of each customer. To show the capabilities of CPALBS, consider the example of hotel finder application. The user requests the available services via mobile device as his/her predefined location range and preferences. Before reporting the query answer, CPALBS will check the user preferences and desired location .The system maps user preference query and extracts the optimal answers related to user preference query. Then, the system returns the most relevance query answers to the users. Thus the proposed system would not report expensive hotels,

hotels without user interest amenities, or hotels with conflicting dietary offerings and hotels without desired location.

The rest of this paper is organized as follows: Related work is highlighted in section 2. The proximity detection algorithm to find location range introduces in section3. Section 4 presents proposed tree matching algorithm to efficiently mapping preference queries. Section 5 covers proposed system architecture. Description of the system is presented in section 6. Finally, Section 7 gives the conclusion.

II. RELATED WORK

With the explosive growth of location based services, several systems have been developed for location based queries.M.Pannevis and M.Marx [5] designed context-aware location and time based system on a normal mobile phone for providing services to the users using web sources. The system returned possible query answers for user requests based on location. The user get too many results returned from the web sources because the system did not consider user preferences. T.Zhi, C. Wang, G. Jia and J.Huang [6] presented context-aware location based services which support context awareness. The system used pre-defined rules to do context reasoning and solve conflict. And then, it deduced user preferences after user history checked. L.M.Wang, M.Qi and C.E.Lin [3] presented user preference awareness in city traveler helper system based on Naïve Bayes classification. The system just learns the user's preferences based on user browsing history from city web server and classifies user preferences and other information. And then, the system returned the filtered preference information to the user. L.XiDong, Y. Ghoashi and T.Hai [4] proposed android based wireless location and surrounding search system designs for finding the banks, supermarkets, gas stations and other places around user and providing navigation function. The system searched user query based on location without considering user preferences. C.Ahn and Y.Nah [1] designed location based web-service framework for providing users to access desired services by interacting with service providers at any places using mobile device. But, the system did not consider user preferences. Therefore, the returned query answers from service provider contain information that users are not interested in. The proposed system distinguishes itself from all of the previous systems. The system considers user preferences to serve the most relevance information to the user according to user preference query and then uses tree structure database for efficient mapping between user preference query and database record trees.

III. PROXIMITY DETECTION ALGORITHM WITH SYMMETRY APPROACH

The proximity detection algorithm in Figure 1, first inputs user desired location range denotes as *r* and current location point*(lat, long)* denotes as *(x,y)* and then obtain the first surrounding area point of location circle range centered on the origin as *(0, r)*. Then, the algorithm calculates the boundary area decision parameter value denotes *p0* as *5/4-r*. The algorithm calculates the surrounding location range points based on boundary area decision parameter. Finally, the algorithm moves each calculated location points onto the circular path centered on user current location point. In this algorithm, the shape of the location range is circle shape. This algorithm can be reduced computation time by considering the symmetry of circles. The shape of the circle is similar in each quadrant. The algorithm can generate the circle section in the second quadrant of the XY plane by noting that the two circle sections are symmetric with respect to the Y axis. And circle sections in the third and fourth quadrants can be obtained from sections in the first and second quadrants by considering symmetry about X axis. The system can take this one step further and note that there is also symmetry between octants. Circle sections in adjacent octants within one quadrant are symmetry with respect to the 45 degree line dividing the two octants. Taking advantage of the circle symmetry in this way, we can generate all points positions around a circle by calculating only the points within the sector from X=0 to X=Y. After calculating all users desired surrounding range distance points, the system extracts requested services within surrounding range.

Algorithm 1: Proximity Detection Algorithm

1. Input:
   - r= distance range
   - x_c,y_c= mobile user position
   - Center on the origin as: $(x_0,y_0) = (0,r)$
2. Calculate the boundary value as
   - $p_0 = 5/4 - r$
3. Starting at k=0,
   - If $p_k < 0$ then {
     - Next point = (x_{k+1},y_k)
     - Calculate $p_{k+1} = p_k+2x_{k+1} + 1$
   - }
   - else {
     - Next point = (x_k+1, y_k-1)
     - Calculate $p_{k+1} = p_k+2x_{k+1} +1 - 2y_{k+1}$
   - }
4. Determine the symmetry point in other seven octants
5. Move each calculated points on (x, y) to (x_c, y_c)
   - $x_c = x +x_c$ $y = y+ y_c$
6. Repeat step 3 through 5 until $y \le x$

Figure1. Proximity Detection Algorithm

The point positions along a circle path within 1 Km are calculated using proximity detection algorithm and the symmetry positions in the first quadrant is shown in Figure 2.

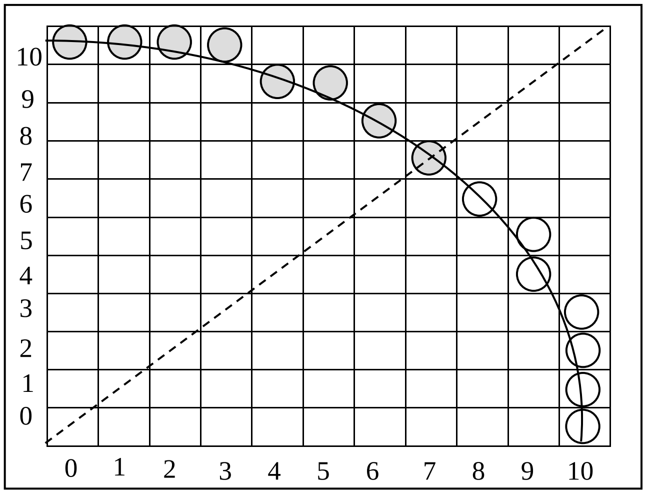

Figure2. Calculated point positions (solid circle) along a circle path with r=1Km centered on the origin, using proximity detection algorithm. Open circles show the symmetry positions in the first quadrant.

Once the surrounding area is determined, an information request is issued to the database server to retrieve the area's requested services. By extracting only the services within a certain range the system are allowing for the possibility that database server could make use of spatial data structures in order to find the relevant data set. When this data set is returned to the CPALBS application, a filtering step takes place to select only those services that are within a pre-determined distance from the mobile user. This algorithm proves to be more efficient compared to an exhaustive computation of distance over all available services. Figure 4 shows requested range and the relevant information area within predefined location range.

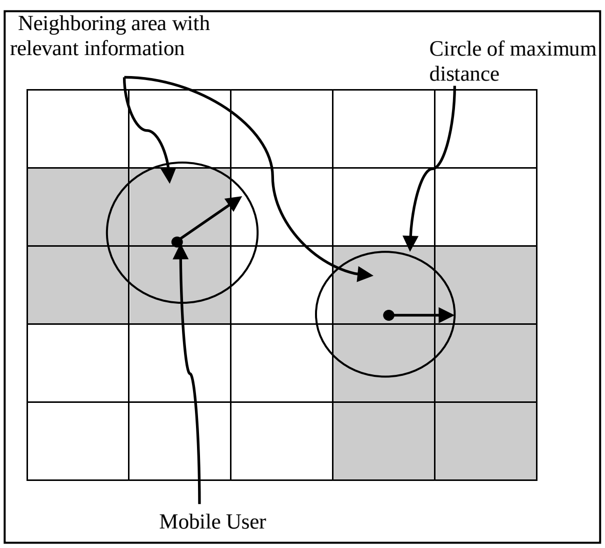

Figure3. Selection of relevant area within specified location range

IV. EXTRACTING OPTIMAL ANSWERS USING SIMPLE TREE MATCHING

Several algorithms have been proposed to address the problem of finding the minimum set of operations (i.e., the one with the minimum cost) to match one tree with another. In general, the minimum cost of mapping between two trees can be cross-layer matching and nodes replacement. All the formulations have complexities above quadratic. In [2], a solution based on dynamic programming is presented with the complexity of *O (n1n2h1h2)*, where *n1* and *n2* are the sizes of the trees and *h1* and *h2* are the heights of the trees. In [7], Yang et al. proposed a simple tree matching algorithm, which makes use of dynamic programming to calculate the maximum number of node-pair between two trees. This algorithm does not allow cross-layer matching and nodes replacement. Compared with the general tree matching algorithms, this algorithm significantly reduces the time complexity. This system uses the simple tree matching algorithm to calculate the maximum mapping between use preference query tree and database record tree.

Tree matching algorithm call the simple tree matching to get optimal answers based on user preference query. Finally, the algorithm shows optimal k-answers from the result set. Let $P=R_P$:<$P_1,P_2,\ldots,P_m$> and $D=R_D$:<$D_1,D_2,\ldots,D_L$> be preference query tree and database record tree, where R_P and R_D are the roots of P and D respectively. Let R be the result sets of the return value from simple tree matching. The algorithm is shown in Figure 4.

Algorithm 2: TreeMatching

1. Initialization: m= number of preferences in P
   L=number of database tree records in D
2. R=Ø
3. for j=1 to L do
4. r = SimpleTreeMatching (P,D_j)
5. R{r} =R{r} U D_j
6. end for

Figure 4. Tree matching algorithm

Simple tree matching algorithm calculates the similarity by using dynamic programming to produce the greatest matching, the algorithm complexity is *O(mn)*, where m and n are the size of A and B. The specific algorithm [6] is shown in Figure 5.

Let's A and B are two trees, i and j for two nodes in A and B, respectively. Following [6], a mapping, M, between two trees as follow: For any node pair *(i , j)*∈ M((neither i nor j is root), let (*parent(i),parent(j)*) ∈ M .The maximum matching of two trees is the matching which has the maximum number of matching pairs.

Let $A=R_A$:< $A_1,\ldots,A_m$> and $B=R_B$:< $B_1,\ldots,B_n$> be two trees, where R_A and R_B are the root of A and B, respectively; A_i and B_j be the ith and jth node of the first-level subtrees of A and B, respectively. If the symbols of R_A and R_B

are the same, then the maximum matching of A and B (i.e. W (A,B) is $M(<A_1,A_2,\ldots,A_m>, <B_1,B_2,\ldots,B_n>)+1$, where $M(<A_1,A_2,..,A_m>, <B_1,B_2,..,B_n>) = \max(M(<A_1,A_2,..,A_{m-1}>, <B_1,B_2,..B_{n-1}>) + W(A_m, B_n), M(<A_1,A_2,\ldots,A_{m-1}>, <B_1,B_2,\ldots,B_n>), M(<A_1, A_2,\ldots,A_m>, <B_1, B_2, \ldots, B_{n-1}>))$. If R_A and R_B contain distinct symbols, then W (A, B) =0.

Algorithm3: SimpleTreeMatching (A, B)

1. if the roots of the two trees A and B contain distinct symbols then
2. return 0;
3. else m = the number of first-level subtrees of A;
4. n = the number of first-level subtrees of B;
5. Initialization: M[i, 0] = 0 for i = 0, … , m;
   M [0, j] = 0 for j = 0, …, n;
6. for i = 1 to m do
7. for j = 1 to n do
8. M [i, j] = max (M [i, j-1], M [i-1, j], M [i-1, j-1] + W [i, j]) where W [i, j] = SimpleTreeMatching (A_i, B_j);
9. endfor
10. endfor
11. return (M[m, n]+1);
12. endif

Figure 5. Simple Tree Matching Algorithm

First, the roots of two trees are compared. If they contain distinct symbols, the two trees totally do not match. If they contain the same symbols, the algorithm recursively finds the maximum matching between the first-level subtrees of the two trees, and saves the matching in the W matrix. Then, we calculate the value of matrix M according to W.

V. SYSTEM ARCHITECTURE DESCRIPTION

This section introduces the detail architecture for our location based service. The architecture consists of Transmission/Reception Layer, Service Layer, Data Processing Layer and Database Service Layer. Figure 6 shows the outline of our system architecture.

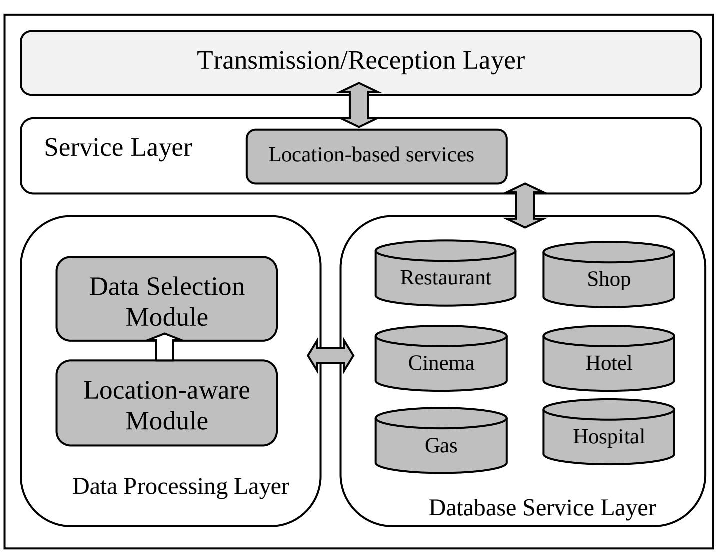

Figure 6. System Architecture

A. Transmission/ Reception Layer

The function of the Transmission/Reception Layer is to make the diversities of different wireless communication networks (e.g., 3G, IPv4, and IPv6) transparent to users. Interfaces of different wireless communication network are encapsulated into a standard self-adaptive interface to client.

B. Service Layer

This layer provides location-based services to users. The Location-based service starts when Service Layer receives the initial request from the Transmission/Reception Layer and begins the initial request with the user preference query to the Database Service Layer. The Data Processing Layer process with the address information in Database Service Layer according to user preference query.

C. Database Service Layer

As shown in Figure, the Database Service Layer contains various types of original information such like restaurant information, shop information, cinema information, hotel information, gas station information and hospital information.

D. Data Processing Layer

There are two modules in this layer: Location-aware Module and Data Selection Module. Location-aware Module tries to relate the information in the database to their locations within specified location range. The function of the Data Selection Module is selection parts of the information relevant to the user preferences to support future services.

(1) Location-aware Module

Location-aware Module analyzes the information in database based on symmetry approach of location range query, extracts the address information within calculated location range. The extracted address information within user requested range becomes the input of Data Selection Module to filter optimal services based on user preferences.

(2) Data Selection Module

There are massive data in the calculated location range, but just part of them according to user preferences will be used in location based services due to time and resources limit. Accordingly, we must select the information which users are interested in or really need. Taking restaurants for an instance, there are lots of restaurants around a user, if we return the information of all the restaurants to him, it will make the service boring and take a long time because of the transmission of huge amount of data. Moreover, the information of large amounts of restaurants may just puzzle him but not guide him to choose one. Therefore, a data selection is important and the core function of the Data Selection Module.

For the specified location range, this module will select parts of the information related to this range according to user preferences based on STM and extract all the matched address information within this location range. Finally, Data Selection Module ranks the extracted address information according to similarity values and returned top-k optimal answers.

VI. DESCRIPTION OF THE SYSTEM

This section describes how the proposed system organizes detailed services information in the database and explains how it extracts the necessary information based on user preference query and desired location and then shows the experimental results for clarify.

A. Service Database Structure

A database D is a directed, labeled tree. The system uses N(D) and E(D) to denote the nodes and edges, respectively, of D. We use l(n) to denote the label of a node n $\in$ N(D). A node n can have textual content, denoted tc(n). Given n,m$\in$ N(D), m is a child of n if (n,m) $\in$ E(D).
Listing1 contains database records that uses in the proposed system. In Database, the hotels, restaurants and other services information store as XML structure in the database is shown as the following.

```
<Hotels>
 <Hotel>
  <Name>Sedona Hotel</Name>
  <Street>Kabaaye Pagoda Street</Street>
  <Phone> 09-73017266/01-293487/01-287694</Phone>
  <Amenity>
        <Cuisine>Chinese/Myanmar/Italian</Cuisine>
        <Gym>1</Gym>
        <Spa>1</Spa>
        <Internet>1</Internet>
        <Valet>0</Valet>
  </Amenity>
  <PriceHigh>200</PriceHigh>
  <PriceLow>100</PriceLow>
  <Latitude>16.84857662857465</Latitude>
  <Longitude> 96.19422912597656</Longitude>
  <Rating>5</Rating>
 </Hotel>
…………..
</Hotels>
```

Listing1. XML structure for hotel information

B. User Preference Query

A preference query, or simply a query for short, Q = (V, E, C) is a rooted directed graph with labeled variables (i.e., nodes) V, edges E and constraints C. The set C contains constraints. Each constraint c is either simply true or is of the form $v\ \theta\ s$ where $\theta \in \{=, \neq, \nsim, \sim, \leq, \geq, <,>\}$ and s is a constant. Constraint C uses$\sim$ ($\nsim$) to denote (none) containment of the value s in the textual content of a node.

For example, David finds a hotel within 1 Km around him. Ideally, David would like a cheap (at most150$ per night), 5-star hotel in Yangon, Myanmar. Since he would like to taste the eastern cuisine, he would like Chinese food to be served at the hotel. David needs an Internet connection, to keep in touch. Finally, David wants Spa and Gym to be at the hotel. Query Q1 in Figure 7 expresses David's hotel desires precisely.

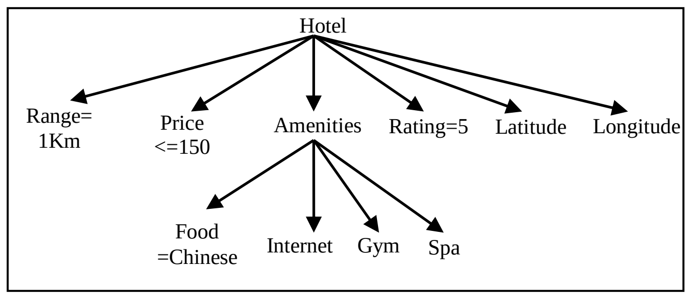

Fig 7. User Preference Query Structure

V (Q) denotes the variables of Q, E (Q) denotes the edges of Q and C (Q) denotes the set of constraints in Q. If E (Q) forms a tree, then Q is a tree query. Let D be a database and Q be a query. Let γ be a mapping of the variables in Q to the nodes in D.

Given a database D and a query Q, the result of applying Q to D is the set Ans(Q, D) of mappings γ: V (Q) $\rightarrow$ N (D) that satisfy all variables, edges and constraints in Q.

C. Mapping and Extracting Optimal Answers within Location Range

The system considers the result of evaluating user preference query Q over database D. At first, the system looks for hotels within 1km distance range from his current location by applying midpoint circle calculation approach. The algorithm calculates the surrounding location range points based on boundary area decision parameter. Finally, the algorithm moves each calculated location points onto the circular path centered on user current location point. After calculating all users desired surrounding range distance points, the system extracts hotel information within surrounding range. Then, the system uses simple tree matching algorithm to look for the relevant hotels as user desires. As a user preference query in figure 8, the price of hotel must be less than or equal to 100, food must be Chinese, and hotel must have internet, gym and spa and then the hotel rating must be five stars. The system matches user preference query tree with the extracted hotels information within user requested range. The system extracts the hotels information {Sedona, Micasa, Innya Lake, Marina, Hotel Yangon, ….} as the maximum mapping of user preference query shown in Table 1. Finally, the system ranks the optimal answers according to descending order of similarity values as {Sedona, Innya Lake, Hotel Yangon, Micasa, Marina, ….} and returns the most relevant top k-answers to the user according to user preferences as shown in Table 2.

Table1. Preference query results

Hotels	Similarity
Sedona Hotel	6
Micasa Residence	3
Innya Lake Hotel	5
Marina Residence	2
Hotel Yangon	4

……………….

Table2. Optimal Top-K Answers after ranking query results

Hotels	Similarity
Sedona Hotel	6
Innya Lake Hotel	5
Hotel Yangon	4
Micasa Residence	3
Marina Residence	2

The result of this step is a modified list with Hotel elements that include Distance, Direction and Preference Level sub-elements (bold entries) as shown in Listing 2.

```
<Hotel>
    <Name>Sedona Hotel</Name>
    <Street>Kabaaye Pagoda Street</Street>
    <Phone> 09-73017266/01-293487/01-287694</Phone>
    <Amenity>
        <Cuisine>Chinese/Myanmar/Italian</Cuisine>
        <Gym>1</Gym>
        <Spa>1</Spa>
        <Internet>1</Internet>
        <Valet>0</Valet>
    </Amenity>
<PriceHigh>200</PriceHigh>
    <PriceLow>100</PriceLow>
    <Latitude> 16.84857662857465</Latitude>
    <Longitude> 96.19422912597656</Longitude>
    <Rating>5</Rating>
    <Distance>2.6015</Distance>
    <Direction>SE</Direction>
    <PrefernceLevel>6</PreferenceLevel>
</Hotel>
```

Listing2. Hotel element modified with distance, direction and preference level

The experimented results are shown in Figure 8.

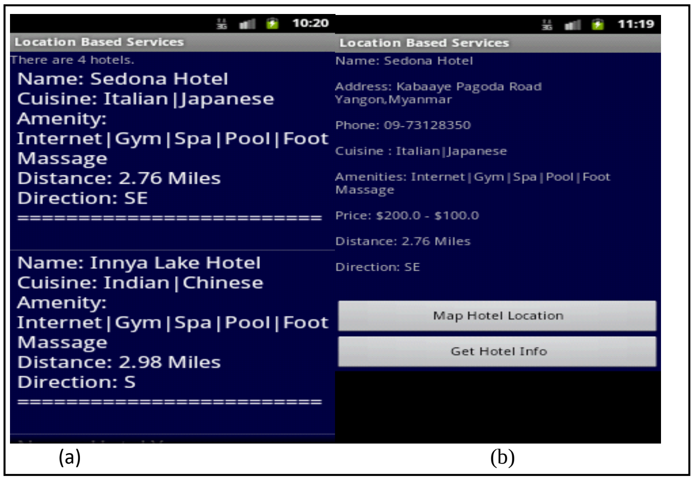

Figure 8.Client terminal display. (a) The terminal lists hotels by user preferences and then by distance within user specified service area. (b) The user selects a hotel on the list to get detailed information.

VII. CONCLUSION

In this paper, the rigidness in current location-based applications that provide services based only on the location context while ignoring various forms of user preferences have been discussed. To overcome such rigidness, this paper presents the system architecture of a Context and Preference-Aware Location based service system (CPALBS) that delivers personalized services to its customers based on the user desired location range and preference query. Within the framework of the CPALBS, the system proposes a proximity detection algorithm for location range query and an extended simple tree matching to find optimal answers according to a user preference query. Then, the proposed system uses tree structure database to improve performance of extracting possible answers with user preferences. Finally, the system serves users with the most relevant optimal top k-answers within predefined location range according to the user preference query efficiently and accurately.

REFERENCES

[1] C.Ahn and Y.Nah [5]. “Design of Location-based Web Service Framework for Context-Aware Applications in Ubiquitous Environments. 2010 IEEE Internation Conference on Sensor Networks, Ubiquitous, and Trustworthy Computing.

[2] K.C.Tai. The Tree-to-Tree Correction Problem. Journal of the ACM, 26(3), pp. 422-433,1979.

[3] L.M.Wang, M.Qi and C.E.Lin. “User Preference Awareness in City Traveler Helper System Based on Naïve Bayes Classification”.IEEE, 2008 ISECS International Colloquium on Computing, Communication, Control, and Management.

[4] L.XiDong, Y. Ghoashi and T.Hai. “Android Based Wireless Location and Surrounding Search System Design”. IEEE, 2010 Ninth International Symposium on Distributed Computing and Applications to Business, Engineering and Science.

[5] M.Pannevis and M.Marx. “Using Web –sources for Location Based Systems on Mobile Phones”.MobIR’08, July 2008, Singapore.

[6] T.Zhi, C. Wang, G. Jia and J.Huang. “Toward Context-Aware Location Based Services”. IEEE,2010 International Conference on Electronics and Information Engineering (ICEIE 2010).

[7] W. Yang, “Identifying Syntactic Differences Between Two Programs,” Software Practice and Experience, vol. 21(7), pp. 739-755, 1991.