Int. Computer Symposium, Dec. 15-17, 2004, Taipei, Taiwan.

Exploiting Data Mining Techniques to Support Context-aware M-services in Mobile Web Service Systems

Vincent Shin-Mu Tseng and Kawuu Wei-Cheng Lin
Department of Computer Science and Information Engineering
National Cheng Kung University
Tainan, Taiwan, R.O.C.
tsengsm@mail.ncku.edu.tw

***Abstract**- The rapid advance of wireless and web technologies enables the provision of rich kinds of mobile services (m-services) for mobile users. The past studies in mobile services focused mainly on the provision of anytime anywhere services. However, this is insufficient for mobile web systems. The **context** characteristics, such as the user's location, the user's preference and the user's historical behavior, should also be considered in order to provide the users with context-aware services that will benefit the users. In this paper, we propose a framework of intelligent mobile web service system that supports the personalized (preference-aware) and history-aware m-services. In addition, to help the provision of history-aware m-services, we devise a novel data mining method, namely **SMAP-Mine**, that can discover mobile users' sequential movement patterns associated with requested services. Through empirical evaluation under various simulation conditions, **SMAP-Mine** is shown to deliver excellent performance in terms of accuracy, execution efficiency, and scalability. The implementation of this system is also demonstrated to show its practicability.*

Keywords: Context-awareness, m-services, service discovery, data mining, personalization.

1. Introduction

With the progress of information technology, the mobile users are permitted to access World Wide Web (WWW) via portable wireless devices, and this is so called "Mobile Web". In the mobile web environments, the mobile users are able to request various services and applications by cellular phone, PDA, or notebook from arbitrary locations at any time. Hence, the behavior of mobile users becomes more complex than before. In the mobile web environments, the users have limited time to do browsing on moving. Hence, they need an intelligent agent that knows who they are, where they are, how they feel, and what they want, so that personalized services can be provided for the users appropriately.

The past studies for mobile services (m-service) focused mainly on the provision of anytime anywhere services, i.e., the mobile users can request the services transparently with the physical location and time [8]. However, this is insufficient for mobile web systems. The context characteristics [6], such as the user's location [4], the user's preference and the user's historical behavior, should be considered in order to provide the users with context-aware services for benefiting the users.

In this paper, we concentrate on the provision of context-aware m-services. More specifically, we propose a system that can provide the personalized (preference-aware) and history-aware m-services in mobile web environments. One of the context characteristics is the user's preferences. Since the users might have different preferences, it is very important to provide the personalized m-services. For example, a geologist traveling in Hawaii may be most interested in the volcanoes more than the other amusements. However, for a young boy, he might prefer the beach playing instead of the volcano visit in the same place.

Another important context characteristic is the historical behavior of the user. By utilizing the historical behavior records, prediction of suitable services users desired is feasible. Consider the following scenario as an example. Suppose most travelers in San Francisco who had reserved hotel A will browse information on hotel A at the train station S. Afterwards, when they get to hotel A, they will submit a query on "Fisherman's Wharf". Finally, the travelers will request the taxi service at Fisherman's Wharf. Hence, if a new traveler is currently at Fisherman's Wharf with the previous behavior as "Query on Hotel A at station S" and "Query on Fisherman's Wharf at Hotel A" in order, we shall have strong confidence to recommend the taxi service information to this traveler.

The crucial part of this paper is to efficiently discover this kind of patterns, namely sequential mobile access pattern (SMAP), which is composed of sequential movement associated with requested

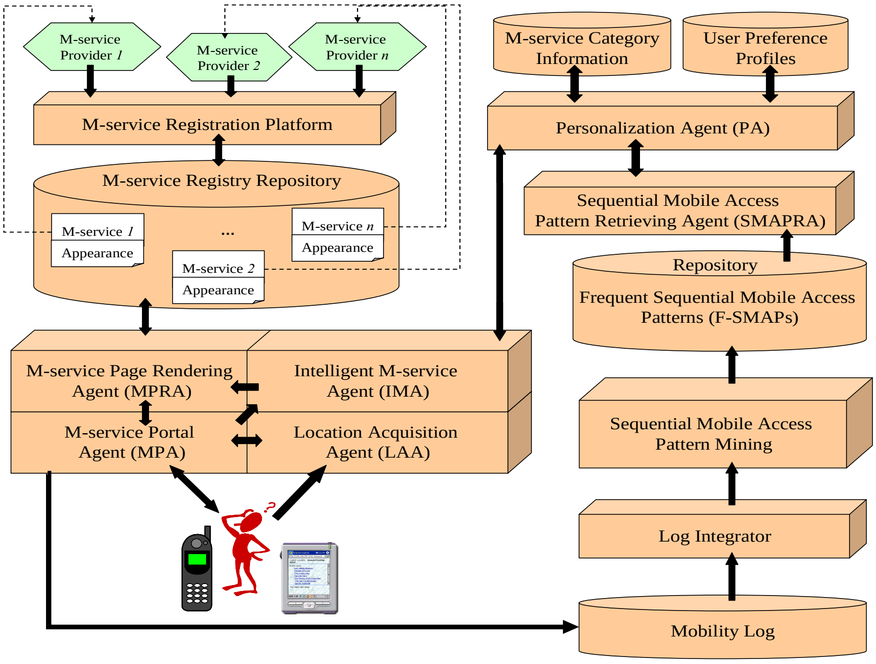


Figure 1. The system architecture of iMoWeS.

services. Over the past few years a considerable number of studies have been made on using data mining techniques to discover interesting rules/ patterns from WWW [5, 9] or large database [1, 2]. However, only a few studies as in [10, 11, 12] have been done on mining the mobility data. Most of these past studies focused only on the aspect of movement behavior analysis or location tracking [3] and did not consider the service request patterns at the same time. This is apparently deficient in understanding the mobile user's behavior. Tseng et al. [12] first studied the problem of mining associated service patterns in mobile web environments. To our best knowledge, however, no studies have explored the issue of mining sequential mobile access patterns that contain both movement and service requests simultaneously.

In this paper, we propose an effective system named iMoWeS (stands for intelligent Mobile Web Service), which supports the personalized and history-aware m-services. In addition, to help the provision of history-aware m-services that can discover patterns of sequential movement associated with requested services for mobile users in mobile web systems. Through empirical evaluation on various simulation conditions, SMAP-Mine is shown to deliver excellent performance in terms of accuracy, execution efficiency, and scalability. The proposed system is also implemented to demonstrate its practicability.

The rest of this paper is structured as follows. In Section 2, the system architecture is described in details. In Section 3, we describe the proposed mining method and the underlying data structures. The empirical evaluation for performance study and real implementation are given in Section 4. The conclusions and future directions are given in Section 5.

2. System Architecture

iMoWeS provides the mobile users location-dependent, context-aware and personalized m-services by the interaction among distributed agents. Figure 1 shows the architecture of iMoWeS system. By the functionality, the iMoWeS system can be further divided into four parts as listed below:

1) M-service registration mechanism
2) Mining mechanism for discovering sequential mobile access patterns
3) Personalization mechanism
4) Context-aware m-service furnishing mechanism

Under the iMoWeS system, a user might request an m-service registered by some service providers from the M-service Portal Agent (MPA). Upon the request submitted from the mobile user is received by MPA, the Location Acquisition Agent (LAA) is asked to return the user's current location. Subsequently the MPA will fill the location attribute of the request and redirect it to the corresponding service providers. Under this mechanism, the service providers are able to provide location-dependent services based on the location explicitly auto-filled by MPA. In addition, this request will also be sent to the M-service Page Rendering Agent (MPRA) and the Intelligent M-service Agent (IMA). In order to obtain personalized recommendation, the IMA then submits a recommendation query which contains the user's current location (acquired from LAA), current requested m-service (acquired from MPA), and the historical behavior (sequential movement associated with requested services) to Personalization Agent (PA), and waits for the recommendation. Once PA receives the recommendation query, it asks the Sequential Mobile Access Pattern Retrieving Agent (SMAPRA) to retrieve the previous discovered large sequential mobile access patterns (L-SMAPs, which will be described in details in Section 3) according to the user's historical behavior for preference scoring by utilizing the user's preference profile and the m-service category information. After the preference scoring and ranking are done on the L-SMAPs,

PA returns the top N patterns in which the user might be most interested to IMA.

For IMA, it immediately passes the recommendation to MPRA after receiving the recommendation from PA. Afterwards, MPRA incorporates the recommendation into the pages user requested (redirected by MPA), and then returns the rendered pages to the user via MPA.

We describe the four mechanisms in details in the following subsections.

2.1 M-service Registration Mechanism

All of the provided m-services should be registered on the m-service registration platform. On registering an m-service, the service providers must provide two XML files in which the detailed descriptions of logic structure and visual appearance are separately presented. After the format validation on the new registered m-service is passed, these two XML files are stored into the service registration repository. Below we describe the m-service registration mechanism in detail.

- **M-service Registration Platform.** This platform allows service providers to register and modify the provided m-services. Every service registered in the iMoWeS System is represented by two XML files: a logic structure file and a visual appearance file. We separate the logic structure and visual appearance for each m-service. This eases the maintenance for service developers and the automation for agents in iMoWeS. Each service has a service name for semantic understanding, an unique ID for service identifying, a pertained category for personalized service provision, a location attribute for location-dependent service provision, and an universal resource location (URL) and relevant parameters for execution. Notice that each location-dependent m-service must carry an attribute, namely *UserCurLocation*, specifying the user's current location in the logic structure file. The value of UserCurLocation is *auto-filled* by the MPA. This is to say that the user can have location-dependent m-service without explicitly informing the server of the location information.
- **M-service Registry Repository.** Each format-validated XML file will be stored into the repository with service ID as the primary key. The repository is an XML database and it is replaceable by any database that supports XML storage such as Oracle 10g, MS-SQL Server 2000, etc.

2.2 Data Mining Mechanism for Discovering Sequential Mobile Access Patterns

We propose a data mining mechanism to discover the SMAPs which form the bases of personalized and history-aware recommendations. The discovered patterns will be retrieved by PA for further scoring. The mining mechanism is constituted by these components.

- **Mobility Log and Log Integrator.** In the runtime, the MPA (described in Section 2.4) acquires the current location of the user from LAA (described in Section 2.4). Each service request together with the location information is logged onto this repository in the format of (UserID, Location, ServiceID, time). Table 1 is an example of the integrated mobility log, which is joined with UserID as the major key.

Table 1. An integrated mobility log.

User ID	Access Pattern
1	<(a,1)(b,2)(c,5)(d,8)>
2	<(a,1)(b,3)(c,5)(d,8)>
3	<(a,3)(b,2)(d,7)>
4	<(c,6)(b,2)(d,7)>
5	<(c,8)(b,1)>
6	<(a,3)(b,6)(c,8)(d,7)>

- **Sequential Mobile Access Pattern Mining.** In this paper, we devise a novel mining method, namely SMAP-Mine, which can discover patterns of sequential movement associated with requested services for mobile users in mobile web systems. Unlike the traditional web log mining [5, 9], SMAP-Mine deals with *two-dimensional* datasets, namely the location and service dimensions. The discovered patterns will be stored in a database for providing personalized and history-aware recommendation. The crucial part of iMoWeS is the discovery of L-SMAPs. The detailed algorithms and data structures are given in Section 3.

2.3 Personalization Mechanism

Two agents, namely Sequential Mobile Access Pattern Retrieving Agent (SMAPRA) and Personalization Agent (PA), are devised to provide personalized recommendations. In the runtime, PA receives the historical behavior of a user and then asks SMAPRA to return the previously discovered L-SMAPs which start with the given historical behavior. Then, PA scores and ranks the L-SMAPs by utilizing the user preference profile and m-service category information and returns to IMA the top N patterns. These two agents are described as follows:

- **Sequential Mobile Access Pattern Retrieving Agent (SMAPRA).** SMAPRA provides an interface for retrieving the discovered L-SMAPs by the given pre-behavior. Table 2 is a part of discovered L-SMAPs. Under the given pre-behavior set to (a,1)(b,2), three patterns starting with (a,1)(b,2) are returned, i.e., (c,3), (e,8), and (f,6)(m,3). Figure 2 shows the interface and message format of SMAPRA. Each message is associated with a message ID for message identification.

Table 2. An example of discovered L-SMAPs.

L-SMAPs
(a,1)(b,2)(c,3)
(a,1)(b,2)(e,8)
(a,1)(b,2)(f,6)(m,3)
(c,2)(d,3)
(b,3)(d,1)
(a,1)(d,1)

```
<SAMPRA_Invocation>
  <msgID></msgID>
  <historicalBehavior></historicalBehavior>
</SAMPRA_Invocation>
```

Figure 2a. Invocation interface of SMAPRA.

```
<SAMPRA_Message>
  <msgID></msgID>
  <L-SMAPs>
    <L-SMAP></L-SMAP>
  </L-SMAPs>
</SAMPRA_Message>
```

Figure 2b. Returned message of SMAPRA.

- **Personalization Agent (PA).** For the purpose of providing personalized recommendation, all of the users have their own preference profiles. The profile records user's preferred categories by assigning each category a *preference score*. Note that each m-service structure file has an attribute, <Category> that indicates to which category it belongs. PA uses the following function to score a given pattern by employing the profile and category information.

$$\sigma \cdot S(C(I_1),u) + (1-\sigma) \cdot \frac{\sum_{m=1}^{l} S(C(I_m),u)}{l}$$

$$0 \le \sigma \le 1$$

$C(I_m)$: the category of service I_m

$S(c,u)$: the preference score for category c assigned by user u

```
<PA_Invocation>
  <msgID></msgID>
  <RecommendationType></RecommendationType>
  <preBehavior></preBehavior>
  <UserID></UserID>
  <TopN></TopN>
</PA_Invocation>
```

Figure 3a. Invocation interface of PA.

```
<PA_Message>
  <msgID></msgID>
  <Recommendation>
    <postBehavior></postBehavior>
  </Recommendation>
</PA_Message>
```

Figure 3b. Returned message of PA.

Figure 3 shows the interface and message format of PA. PA supports three kinds of recommendation (via the <RecommendationType> of the interface), 1) next service, 2) next location together with service, and 3) entire pattern. IMA can choose a suitable one to call according to the application scenario.

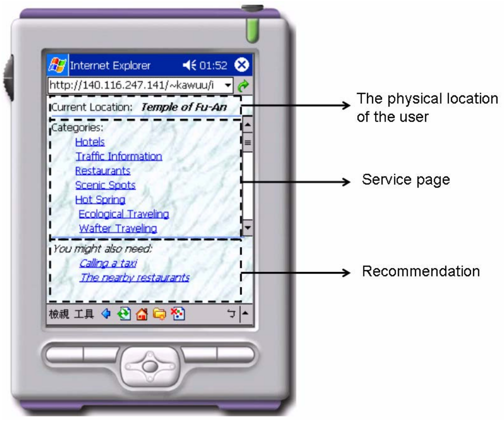
2.4 Context-aware M-service Furnishing Mechanism

iMoWeS furnishes the context-aware m-service through these agents, Location Reporting Agent (*LRA*), Location Acquisition Agent (*LAA*), M-service Portal Agent (*MPA*), M-service Page Rendering Agent (*MPRA*), and Intelligent M-service Agent (*IMA*). The main purpose of this mechanism is to furnish the mobile users location-dependent, personalized, and history-aware m-services. For locating the users under this system, we place an agent called LRA on each user's mobile device. LRA is an agent responsible for periodically reporting the user's current location to LAA in GPS form. iMoWeS is able to track the location of each mobile user through the communication between LRA and LAA.

When a user requests the service listed on the m-service portal, MPA will fill the location attribute of this request and redirect this request to the service providers. This helps the service providers to provide location-dependent service. The result pages are then sent to MPRA for page rendering. At the same time this request will also be redirected to MPRA and IMA with the same message ID for message identification among MPA, MPRA, and IMA. IMA uses the user's historical behavior to obtain personalized and history-aware recommendation from PA, and sends this information to MPRA. Once the result pages and recommendation are available, MPRA renders new pages by incorporating the recommendation into the result pages, and returns to the user's device via MPA. Through this mechanism, the location-dependent, personalized, and history-aware m-services are furnished.

- **M-service Portal Agent (MPA).** iMoWeS has an m-service portal for the users to browse the m-services. All of the m-services are listed on the portal by their pertained categories. In our system, this portal is implemented by Java Servlet and simplified JavaScript. Users can access the m-services with any browser supporting standard HTML technology. However, the mobile devices might differ from users. We use the CC/PP profile to help iMoWeS with the provision of device-adaptive m-service. A CC/PP (Composite Capabilities/ Preference Profiles) profile is a description of a device capabilities and user preferences. This profile characterizes the delivery context of a device and can be used for content adaptation to that device

- **M-service Page Rendering Agent (MPRA).** The main functionality of MPRA is to render pages containing the result pages from service providers and the recommendation from IMA. Figure 4 is an example of rendered page. As indicated in Figure 4, this page contains three frames, each of which lists the physical location of the user, the requested page, and the recommendation, respectively.



Internet Explorer 01:52
http://140.116.247.141/~kawuu/i
Current Location: *Temple of Fu-An*
Categories:
Hotels
Traffic Information
Restaurants
Scenic Spots
Hot Spring
Ecological Traveling
Wafter Traveling
You might also need:
Calling a taxi
The nearby restaurants

Figure 4. An example of rendered page.

- **Intelligent M-service Agent.** IMA has a temporary storage that records the history of each mobile user under the system. When an user submits a request, IMA will receive it from MPA associated with an unique message ID and query the PA with the user's historical behavior to obtain personalized and history-aware recommendations. IMA then sends this recommendation to MPRA for rendering new pages. We show the m-service furnishing process in Figure 5.

(*) The LRA reports its location to LAA periodically.
(0) User submits a query.
(1) MPA acquires the location of this user from LAA.
(2) LAA returns the location.
(3) MPA asks IMA to obtain recommendation from PA.
(4) IMA sends the recommendation to MPRA.
(5) MPA redirects the requested pages to MPRA.
(6) MPRA incorporates the recommendation into the pages, and returns the pages to the user via MPA.

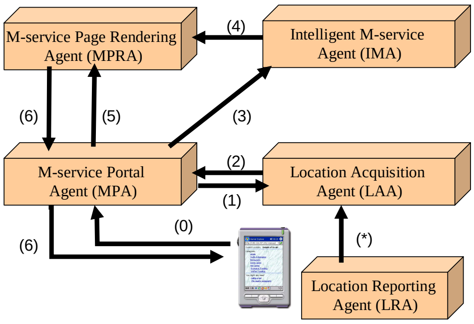


Figure 5. The m-service furnishing process.

3. Mining of Sequential Mobile Access Patterns

3.1 Problem Definition

Consider two sets L and S. For each element l in L and each element s in S, we form an ordered pair $p=(l, s)$, where l and k are taken as the first and second element of p, respectively. Two ordered pairs (l_1, s_1) and (l_2, s_2) are said to be equivalent if and only if $l_1 = l_2$ and $s_1 = s_2$. Let P be the set of all ordered pairs and we write it as:

$$P = L \times S = \{(l,s) \mid l \in L \text{ and } s \in S\}.$$

Let T=< (p_1, t_1) (p_2, t_2) ... (p_m, t_m) >, where element (p_i, t_i) is composed of an ordered pair p and a time point t. We say T is a mobile access pattern with length equal to m, namely m-pattern.. Meanwhile, (p_i, t_i) is defined as *earlier* than (p_j, t_j) if and only if $t_i < t_j$, and written as $(p_i, t_i) < (p_j, t_j)$ or simply $p_i < p_j$. Note that the value of each time point t is unique in an access pattern, i.e., t_i will never be equal to t_j. The ascending order of elements of access pattern is sorted by using t as the key. Considering only the ordered pairs p_i in T, we rewrite T as $T_s = < (p_1) (p_2) \dots (p_m) >$.

Definition 1. An access pattern $t_s' = < (p'_1)\ (p'_2) \dots (p'_m) >$ is a *sub-pattern* of another access pattern $t_s = < (p_1)\ (p_2) \dots (p_n) >$, written as $t_s' \subset t_s$, if $m \leq n$ and there exists a strictly increasing sequence $(i_1, i_2, \dots, i_m)$ of indices such that for all $j=1, 2, \dots, m$, $p'_j = p_{i_j}$. Here, t_s is called the *super-pattern* of t_s'.

Definition 2. Given a database $D = \{T_{s1}, T_{s2}, \dots, T_{sN}\}$ that contains N access patterns. The *support* of pattern t_s is defined as:

$$\sup(t_s) = \frac{|\{T_{si} \mid t_s \subset T_{si} \text{ and } 1 \leq i \leq N\}|}{N}.$$

Definition 3. t_s is called a *large sequential mobile access pattern* (*L-SMAP*) if sup(t_s) is greater than the user-specified support threshold δ .

With the above definitions, the problem of sequential mobile access patterns mining is defined as follows. Given a database D containing the logs of the mobile users' access patterns and a specified support threshold, the problem is to discover all the L-SMAPs existed in this database. Once all the L-SMAPs are discovered, it is easy to generate the final form of sequential mobile access patterns, as indicated in [1, 2, 11].

3.2 Proposed Method: SMAP-Mine

In this section, we describe the mining method we propose, namely SMAP-Mine. Before conducting the SMAP-Mine algorithm, the dispersed logs recording user's behaviors must be integrated in advance. For the SMAP-Mine algorithm, two phases are included, namely i) construction of SMAP-Tree and ii) mining of sequential mobile access patterns. In the following, we describe the above aspects in details.

3.2.1 Construction of SMAP-Tree

The purpose of constructing SMAP-Tree is to aggregate the access patterns into the memory in a compact form so that the mining of large patterns can be done efficiently. The main advantages of SMAP-Tree are 1) only one physical database scan is needed to mine all of the large patterns, and 2) the SMAP-Tree is compact so that the huge amount of data can be handled efficiently. The construction algorithm is given in Figure 6.

```
Algorithm: SMAP-Tree Construction
Input: the log of mobile access patterns D
Output: a SMAP-Tree
Method: SMAP_Tree(D)
Step 1: For a sequential mobile access pattern smap in D, extract
        the movement sequence ms and the service request sequence
        srs from smap.
Step 2: SMAP-Tree Insertion
        a. Insert ms into the SMAP-Tree by traversing from the
           root node of SMAP-Tree.
        b. Increase the count of each node's label traversed by the
           sequence in the node table.
        c. Set the parent-link and next- link for the node.
Step 3: SR-Tree Insertion
        a. Insert srs into the SR-Tree on the tail node of ms
           in SMAP-Tree.
        b. If the SR-Tree is newly constructed, set the head
           table to record the first node on each level.
        c. Set the parent-link and peer-link for each inserted
           node.
Step 4: If there are more patterns in D, go to Step 1. Otherwise,
        return current SMAP-Tree.
```

Figure 6. Algorithm for SMAP-Tree construction.

3.2.2 SMAP-Mine Algorithm

Figure 7 shows the detailed algorithm for SMAP-Mine, which is based on the depth-first search (DFS) approach extending from [6]. It recursively constructs the SMAP-Trees and mines the trees till termination condition is met. First, we list all the labels with count greater than the support threshold δ by scanning the node table of current SMAP-Tree, and store the labels into a temporary set M_L1. If M_L1 is empty, the prefix pattern of current SMAP-Tree is output as return. The output prefix pattern will be one of the L-SMAPs. Otherwise, for each label *l* in M_L1, all the nodes with label name *l* are stored into a temporary set l_tmp. For each node n in l_tmp, we sum the count for each label in n's cross-peer nodes and store the large labels into a temporary set SR_L1. If SR_L1 is empty, the prefix pattern is output as the L-SMAP and the procedure is ended. Otherwise, each label *s* in SR_L1 is combined with *l* to form a new prefix pattern, denoted as (l,s), and (l,s) is appended to the prefix pattern of current SMAP-Tree. In this way, a new SMAP-Tree is constructed with the new prefix pattern, and the mining procedure is invoked recursively to discover all the L-SMAPs.

```
Algorithm: SMAP-Mine
Input: a SMAP-Tree ST, a specified support threshold δ
Output: all large sequential mobile access patterns (L-SMAPs)
Method: SMAP_Mine (ST, δ)
Step 1: Scan the node table of ST. Get all the labels with count
        greater than δ and store them into M_L1.
Step 2: If M_L1 is empty, output the prefix pattern of ST and return.
Step 3: For each label l in M_L1
           a. Store all the nodes with label=l into l_tmp
           b. For each node n in l_tmp, sum the count for each label in
              n's cross-peer nodes. Store all the labels whose count is
              greater than δ into SR_L1.
           c. If SR_L1 is empty, output the prefix pattern of ST and
              return.
           d. For each label s in SR_L1, reconstruct a SMAP-Tree ST'
              with (l, s) being the prefix pattern.
           e. Call SMAP_Mine(ST', δ)
```

Figure 7. Algorithm for SMAP-Mine.

4. Empirical Evaluation and Implementation

4.1 Simulation Model

To evaluate the performance of SMAP-Mine, we used a simulator [7, 12] that simulates a mobile web environment to generate the workload data. Table 3 summarizes the primary parameters in the simulation model.

Table 3. Primary parameters of our simulation model

Parameter	Description	Default Value
|W|	W*W nodes of Network	5
|E|	The number of mobility events	400
L_e	Average event length	4
P_e	Average probability of each event	0.01
Ph	Probability of popular service per node	0.05
N	The Number of different Services	10,000
Pb	The probability of backward movement	0.1
Pn	The probability of next-node movement	0.2
Ua	User alive time units	10
|U|	The number of Users	100,000
Tc	The time units of Transaction cut time	10

4.2 Experimental Results

We conduct four experiments to evaluate the performance of SMAP-Mine under different system conditions by varying the parameters for number of users, support threshold, length of user events and network size, respectively. Meanwhile, the effects of varying these system parameters were also studied. All of the experiments run on a P4-2.4GHz machine with 1G MB main memory.

Figure 7 shows the execution time and number of L-SMAPs by varying the number of users in the network with other parameters fixed as the default values. It is obvious that the execution time increases linearly with the number of users increased. This demonstrates the scalability of SMAP-Mine under various user scales. Figure 8 shows the experimental results when the support threshold is varied from 0.001 to 0.005. The lower support value leads to more L-SMAPs and higher execution time. This experiment shows the excellent performance of SMAP-Mine even under very small support value.

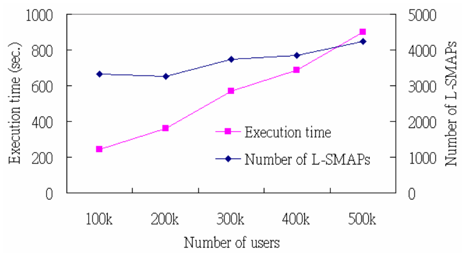


Figure 7. Execution time and L-SMAPs with |U| varied

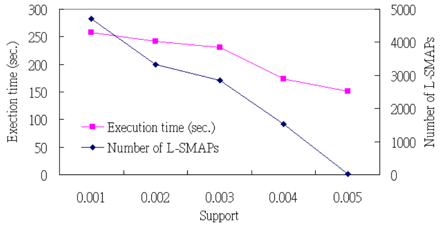


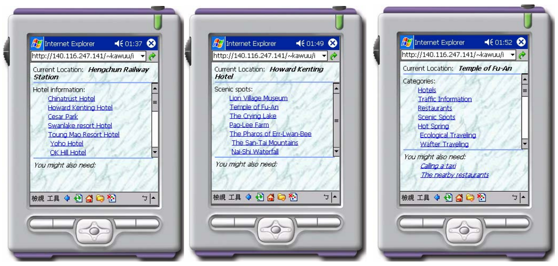
Figure 8. Execution time and L-SMAPs with support varied

4.3 Implementation

We implemented the iMoWeS system by integrating several Java technologies, namely J2EE, J2SE, and J2ME. We implemented the m-service registration platform and m-service portal by Java Servlet and JavaServer Page technologies. In this system, the user can have the m-services with simply a browser that supports the browsing of HTML pages. In the wireless device, an LRA is placed to report the current location acquired from the embedded GPS to LAA periodically (defaulted as 10 minutes). Because the devices may differ from users, the LRA is implemented by PersonalJava 1.2, which is a platform-independent J2ME technology.

Furthermore, the agents in the server side are all implemented by J2SE 1.4.2, and the communication among distributed agents is implemented in Java Message Service (JMS) technology with the communication message encapsulated in XML format. JMS is a message service that provides a reliable and flexible service for the asynchronous exchange of critical data throughout a system.

The implementation was done on the theme of travel guide to a famous scene site in Taiwan named Kenting. Consider a practical pattern (Hengchun station, query the Howard Hotel) (Howard Hotel, query the Temple of Fu-An) (Temple of Fu-An, request the nearby taxi service). We found that most of the travelers browse the reserved hotel (Howard Kenting Hotel) at the Hengchun station. Afterwards, when they get to Howard Kenting Hotel, they will submit a query about the Temple of Fu-An. Finally, the travelers request the taxi service at the Temple of Fu-An. Hence, as shown in Figure 9, when a new traveler is currently at the Temple of Fu-An with the previous history (Hengchun station, query the Howard Hotel) and (Howard Hotel, query the Temple of Fu-An) in order, we now have strong confidence to recommend the taxi service to the traveler.



Current Location: *Hengchun Railway Station*

Hotel information:
Chinatrust Hotel
Howard Kenting Hotel
Cesar Park
Swanlake resort Hotel
Toung Mao Resort Hotel
Yoho Hotel
OK Hill Hotel

You might also need:

Current Location: *Howard Kenting Hotel*

Scenic spots:
Lion Village Museum
Temple of Fu-An
The Crying Lake
Pao-Lee Farm
The Pharos of Err-Lwan-Bee
The San-Tai Mountains
Nai-Shi Waterfall

You might also need:

Current Location: *Temple of Fu-An*

Categories:
Hotels
Traffic Information
Restaurants
Scenic Spots
Hot Spring
Ecological Traveling
Wafter Traveling

You might also need:
Calling a taxi
The nearby restaurants

(a) (b) (c)

Figure 9. (a) The traveler browsed the ordered hotel, Howard Kenting Hotel, at the Hengchun station. (b) The traveler went to Howard Kenting Hotel and prepared to submit a query about the Temple of Fu-An. (c) The traveler moved to the Temple of Fu-An with the recommended services, "Calling a taxi" and "The nearby restaurants" shown on the PDA.

5. Conclusions

With the rapid development of wireless and web technologies, mobile web applications become an emerging field in which rich kinds of services are provided for mobile users. In this paper, we propose an intelligent mobile web service system named *iMoWeS* that can provide personalized (preference-aware) and history-aware m-services. To support the provision of history-aware m-services, we deal with the mobility log by exploiting the data mining techniques to discover frequent patterns for prediction and recommendation. Most of the relevant studies focus only on the aspect of mobility analysis and integrated analysis on both of mobility and service requests are lacked. Hence, we propose a novel data mining method, namely *SMAP-Mine*, which can efficiently discover the patterns of sequential movement associated with service requests for mobile users in mobile web environments. Only one physical scan on the database is needed for *SMAP-Mine* to discover all large access patterns, namely L-SMAPs. To our best knowledge, this is the first work on mining the patterns of sequential movement associated with service requests. Through empirical evaluation and sensitivity analysis under various system conditions, *SMAP-Mine* was shown to perform excellently in terms of accuracy, execution efficiency, and scalability. The implementation and the application to the theme of Kenting Traveling demonstrate the practicability of iMoWeS.

Acknowledgement

This research was partially supported by National Science Council, Taiwan, R.O.C., under grant no. NSC-93-2213-E-006-030 and NSC-92-2522-E-006-150-CC3.

References

[1] Rakesh Agrawal, Ramakrishnan Srikant: Fast Algorithms for Mining Association Rules in Large Databases. Proceedings of 20th International Conference on Very Large Data Bases, Santiago de Chile, Chile (1994) 487-499.

[2] Rakesh Agrawal, Ramakrishnan Srikant: Mining Sequential Patterns. Proceedings of the 11th International Conference on Data Engineering, Taipei, Taiwan (1995) 3-14.

[3] Ian F. Akyildiz, J. Mcnair, Joseph S. M. Ho, H. Uzunalioglu, and W. Wang: Mobility Management in Next-Generation Wireless System. Proceedings of the IEEE, Vol. 87, No. 8 (1999).

[4] Giacomo Cabri, Letizia Leonardi, Marco Mamei, Franco Zambonelli: Location-dependent services for mobile users. IEEE Transactions on Systems, Man, and Cybernetics, Part A, vol. 33, no. 6, (2003) 667-681.

[5] Ming-Syan Chen, J.-S. Park and P. S. Yu: Efficient Data Mining for Path Traversal Patterns. IEEE Transaction on Knowledge and Data Engineering Vol. 10, No. 2 (1998) 209-221.

[6] T. G. Kanter: Attaching context-aware services to moving locations. IEEE Internet Computing, vol. 7 (2003) 43-51.

[7] Weicheng Lin, Shin-Mu Tseng: A Data Generator for Mobile Web Environment. Technical Report, CSIE Dept, National Cheng Kung University, Taiwan, (2002).

[8] Z. Maamar, B. Benatallah, and Q. Sheng: Toward a composition framework for E-/M-services. Proceedings of 1st International Workshop on Software Agents Pervasive and Ubiquitous Environment at AAMAS, Bologna, Spain (2002).

[9] Jian Pei, Jiawei Han, Behzad Mortazavi-Asl, Hua Zhu: Mining Access Patterns Efficiently from Web Logs. Proceedings of 4th Pacific Asia Conference on Knowledge Discovery and Data Mining, Kyoto, Japan, April (2000) 396-407.

[10] Wen-Chih Peng, Ming-Syan Chen: Mining User Moving Patterns for Personal Data Allocation in a Mobile Computing System. Proceedings of the 2000 International Conference on Parallel Processing (2000) 573-580.

[11] Shin-Mu Tseng, Wen-Ching Chan: Mining Complete User Moving Paths in a Mobile Environment. Proceedings of International Workshop on Databases and Software Engineering (held with ICS), Taiwan (2002).

[12] Shin-Mu Tseng, Cing-Fu Tsui: An Efficient Method for Mining Associated Service Patterns in Mobile Web Environments. Proceedings of the ACM Symposium on Applied Computing (SAC), Melbourne, FL, USA (2003) 455-459.