

Developing Personalized Information Services for Mobile Commerce Location-Aware Applications

Christos K. Georgiadis

Department of Applied Informatics

University of Macedonia

Thessaloniki, GREECE

e-mail: geor@uom.gr

Abstract — The mobile setting adds unique characteristics to applications that can be used by mobile commerce client devices, such as ubiquity and location awareness. These devices are known to be limited in terms of computational power, input/output capabilities and memory, thus enhancing the mobile browsing user experience is realistic only if perceptual and contextual considerations are addressed. In this article, we attempt to define and analyze the issues of mobility, taking into consideration factors that would attract users to mobile commerce applications. We focus on how these issues may influence the user acceptance and the quality of personalized location-based services. To improve understanding of the mobile setting, a case study of a context-aware location-based application is designed and carried out. The application is capable to identify and to depict on the map user's current location, to search and detect routes, and to display various user personal points of interest/attractions, along user's current route. It is a personalized application, based on Microsoft MapPoint Web Service technology, in which each user receives information which is strictly related to his identity. Finally, to review certain personalized and user-friendly features of our approach, a typical application scenario is presented.

Keywords - location-based information; context-awareness; mobile commerce adoption; Web Services; mobile setting.

I. INTRODUCTION

Increased sophistication of mobile technology makes itself an ideal channel for offering context-aware personalized services to mobile users [1]. Such services actually, give pace to the rapid development of e-commerce conducted with portable devices, more commonly referred to as wireless or mobile commerce (m-commerce), as they are context-specific to each individual and thus, are capable to attract customer's attention [2]. The interface usability of mobile applications is a critical factor for the acceptance of m-commerce, as a good interface design allows users to achieve high performance [3]. Moreover, as 3G/UMTS services roll out, m-commerce is increasingly used to enable content delivery and payment for personalized and location-based services (LBS) such as image content (maps, photos, etc.), as well as video and audio content, including full length music tracks [4].

This work, undertakes a broad examination of the mobile setting in Sections II and III, to introduce significant

concepts that influence m-commerce user behavior, and to provide major considerations regarding m-commerce user acceptance and quality. In section IV we discuss about location-aware and personalized services to clarify personalization's connection to LBS. Section V entails a case study involving a modern approach to deal with personalized LBS in m-commerce. Here we first analyze the LBS user requirements and then we exploit Microsoft MapPoint Web Service technology to design search processes for addresses, attractions, routes, and creating processes of their respective maps. Section VI deals in detail with the major functional parts (classes and variable types) of our mobile application. Section VII contains an m-commerce information-oriented application scenario to demonstrate some of the major implementation concerns that must be taken under consideration to offer such personalized LBS.

II. ANALYZING MOBILE SETTING

System, environment, and user are actually three different viewpoints to classify the characteristics of the m-commerce applications [5]:

- System perspective - mobile applications present disadvantages, because they provide a lower level of available system resources [6].
- Environmental perspective - mobile applications enable users to access mobile Internet content anywhere and anytime (obviously, a big advantage).
- User perspective – is heavily influenced both by system/environmental characteristics and by certain aspects of the “mobility” concept [7][8][9]:
 - Spatial mobility, the extensive geographical movement of users.
 - Temporal mobility, the ability of users for mobile browsing while engaged in a peripheral task.
 - Contextual mobility, the dynamic conditions in which users employ mobile devices.

As it is depicted in Figure 1, we may distinguish three worth noticing attributes regarding mobile device usage [10]: first, users prefer to treat their mobile device in a quite **personal** way, and favor to access more personalized services (spatial mobility must be considered as the major reason behind this behavior). Second, users have usually **limited attention** as they manage their mobile devices (temporal mobility is the reason of this phenomenon). Finally, users manage their mobile devices in broadly mixed environments, appreciating all accommodations that may provide the **context-sensitive** mobile device **functionality**.

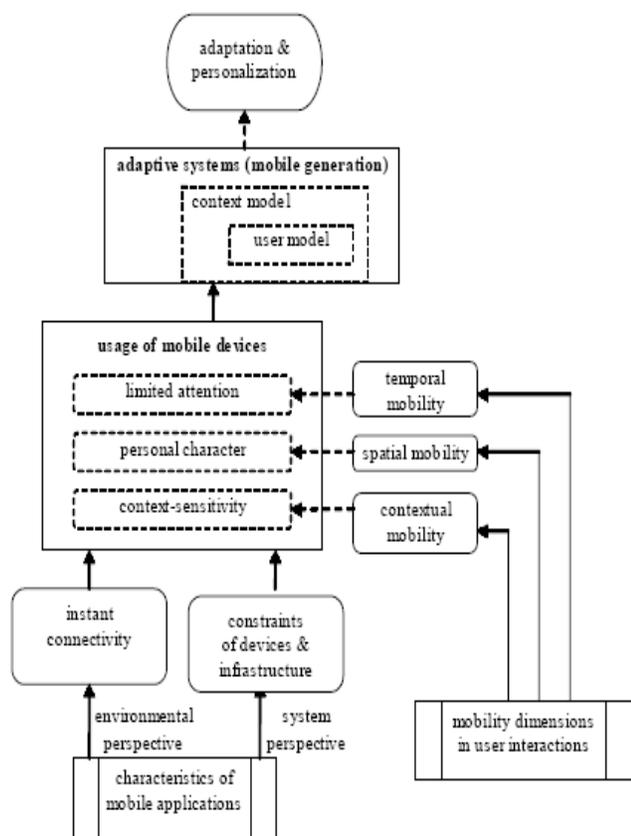


Figure 1. Analyzing mobile setting.

III. M-COMMERCE: USER ACCEPTANCE AND QUALITY

User requirements' satisfaction is among the most useful methods for measuring m-commerce application success. Undoubtedly, a key restraining factor concerning mobile services is financial: cost issues are raised due to confusing billing schemes and their potential dependency on versatile network connections [11]. However in this work, we will leave out cost related issues, and we will analyze the rest of interesting research results related with user satisfaction measuring [12], trying to identify how they can be used to improve the quality of m-commerce applications.

Wu and Wang in [12] revised effectively the technology acceptance model (TAM) [13][14] to cover the additional aspects of the m-commerce environments, as depicted in Figure 2. According to the main research results (which are consistent to a large extent with other researchers' work), various variables influence user adoption of m-commerce applications, as follows:

- M-commerce use can be predicted efficiently from the users' intentions, which are affected significantly by:
 - Perceived risk - the user's subjective expectation of suffering a loss in pursuit of the desired outcome of using m-commerce,

- Compatibility - the degree to which the innovation is perceived to be consistent with the potential users' previous experiences and needs, and
- Perceived usefulness - the degree to which a person believes that using a particular system would enhance his (or her) job performance.

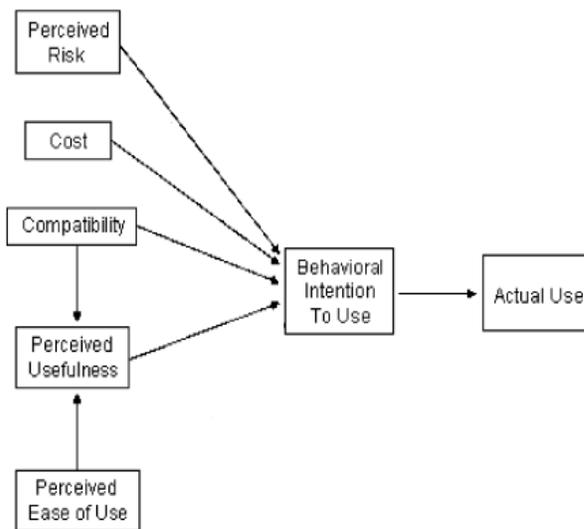


Figure 2. Mobile commerce acceptance model.

- User's possibility to engage in m-commerce transactions (behavioral intention to use), is affected also strongly, by:
 - Perceived risk,
 - Compatibility, which essentially has significant effect on the frequency of using m-commerce (actual use), and
 - Perceived ease of use - the degree to which a person believes that using a particular m-commerce application would be free of effort, which basically influences indirectly, through perceived usefulness variable.

Certain m-service quality criteria are directly or indirectly related with these user adoption variables. In detail, the software quality characteristics as defined by the ISO 9126 software quality standard [15] and adapted to m-commerce systems [16], are usability, reliability, functionality, and efficiency.

Usability aims at simplifying end users' actions and is associated with perceived ease of use. Functionality refers mainly to providing secure and suitable functions to end user. Thus, this quality characteristic is related both with perceived risk and compatibility. Reliability refers to systems tolerance on end users' actions, and consequently is associated also with perceived risk from a different perspective. Finally, efficiency's main attributes are response time and resource behavior, and therefore may be related with perceived usefulness.

The above analysis may provide constructive ideas to m-commerce application developers. It has certainly influenced

our work, to focus on quality issues by supporting user appreciated features satisfactory.

IV. PERSONALIZATION AND LOCATION-BASED SERVICES

A. Personalization

The ‘personal character’ emerges clearly from all quality characteristics in m-commerce context: it is the perception of the ease of use, the perception of risk, the perception of the usefulness and the perception of the consistency of the innovations with a particular user’s experiences. Personalization technology is therefore, one of the most prominent technologies in m-commerce systems: it is not just capable to get the most of the mobile devices’ limited resources. In addition, it provides m-commerce applications the efficiency of supporting users with the content they need in the most optimized approach. This is considered as an effecting way to holding users and to cultivate their loyalty [17].

B. Location-Based Services

LBS are, in general, informative services accessible with mobile devices through the mobile network; they are utilizing the ability to make use of the location of the mobile device [18]. There exist a broad range of different LBS. Figure 3 gives an overview on the main categories of LBS applications (the listing is certainly growing over time) [19].

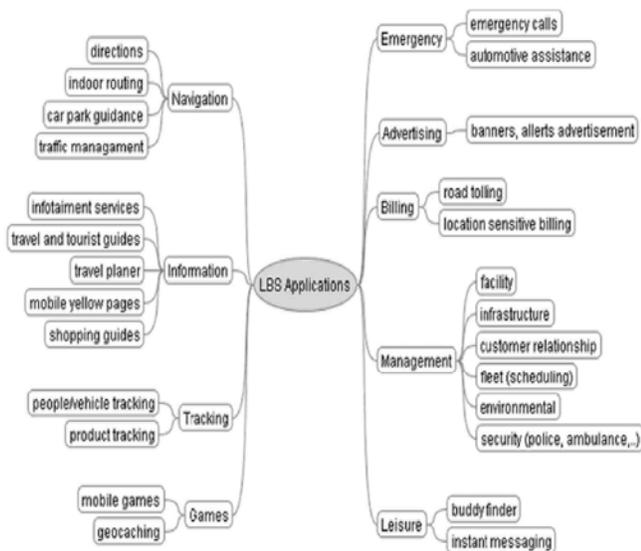


Figure 3. Location-based services application categories.

C. Related Work

In this work we are interested on certain navigation (e.g., providing users detailed directions about how to get to a desired destination starting from their current geographical location) and information LBS (e.g., providing a local street map and notifying about nearby places of interest).

This type of LBS is a characteristic example of mobile device’s context-sensitive functionality: it allows users to

obtain context-specific information to react upon. This makes mobile devices especially fit for providing time-critical and location-sensitive services. Several authors [17] [18] [20] have stressed this functional value of m-commerce, thereby acknowledged the social value that can be derived from mobile services. Moreover, Boll et al. in [21], presented an approach capable to integrate mobile LBS with personalized multimedia presentation. The mobile channel is indeed an extremely personal medium that users carry with them at all times. Consequently, it has become part of the user’s social context and everyday life, influencing all areas of society, such as businesses operation and employees work, advertising opportunities, personal communications, consumer purchases, location-aware services, information locating and retrieval [22].

According to Steiniger et al. in [19], LBS are quite different from more conventional internet based media (traditional guides, directories, maps etc.) because they are aware of the context in which they are being used and can adapt their contents and presentation accordingly. The relation between location and context concepts is very strong: user’s location is considered a major context parameter, because it can be used to characterize the user’s situation [17]. For example, knowing user’s location, the application can determine what other objects or people are near the user and even what kind of activity is occurring in that area. In addition, LBS may make use of a more detailed context. They actually give the possibility of a two way communication and interaction. Therefore the user tells the service provider (i.e., the m-commerce application) his actual context, like the kind of information he needs and his preferences (what), his time (when), and his position (where). This helps the provider of such location services to deliver information tailored to the user needs.

V. OUR CASE STUDY

A case study of a context-aware location-based application is designed and carried out. It is capable to identify the current location of the user and to provide the following information:

- depiction on the map of user’s current location,
- route search/detection and marking on the map of user’s preferred route while he is in move,
- displaying various user personal points of interest/attractions (such as his home, his office, his favorite restaurants, etc.), along user’s current route,
- depiction on the map of general interest type attractions (e.g. museums, subway stations and hospitals),
- editing of user personal attractions (adding new ones, deleting existing, etc.),
- editing of general interest attractions’ categories,
- dynamic update of user’s personal attractions and categories of attractions, based on advanced personalization techniques.

It is undoubtedly a customized application, since each user receives information which is strictly related to his identity. The personalization of services is accomplished by

exploiting identification information available from the mobile device's SIM card. In our case, obviously, there is no SIM card, as we use the .NET mobile device programming environment for simulating the mobile application's behavior.

Thus, in our case, users to enter the system should give some personal codes. That is, there is a form in which a registered user will present his credentials, while an additional form, can be used by a new user to create an account. As a result, a data base in the portable device carries user's data.

The existence of the device' database creates some additional problems. One has to do with the capacity of devices, which affects the size of the database. Since it cannot be very large, data of multiple users is difficult to be saved on the same device. In other words it is not feasible different users to use the same device, and their personal settings (passwords and attractions) to remain stored on the device. But this is obviously not so important, since the mobile device is used mainly as a personal (single user) client machine.

Another similar problem has to do with users' ability to connect to the system from different devices. They should create their account in any device from the beginning, and then adjust the parameters they want. A viable solution is to maintain a database on a server. The device will have to link to this server, and request and receive data whenever it wants (merge replication process).

Location-aware applications may base their operations not only on user's current location, but also on other information about the status of the user, such as time, and weather (context-aware wider character of location-based applications). Finding user's location is possible by various technologies, but as this application was developed and tested locally, finding position and movement-on-a-path operations are made by simulation. That is, a virtual movement of a certain user and its current position changes are defined through the application environment. Certainly, finding the virtual user's current location is accomplished through our application design.

The application consists of three separate units, which interact with each other:

- User interface.
- Mobile database on the device, which deletes its data when session ends.
- Server database, which stores and maintains all mobile users' data, and additionally sends the appropriate data to the proper mobile database (based on user identity information).

A. MapPoint Web Service Technology

MapPoint Web Service (MWS) technology [23] is used to design search processes for addresses, attractions, routes, and creating processes of their respective maps. The MWS is actually a Microsoft's Internet service, which is designed to work using the Graphical Information System (GIS) and responds to various scenarios mapping, which involve different types of applications, such as portals, web pages, but mainly mobile applications. In conjunction with

MapPoint Location Server (server which detects the location of a mobile device) they create a "package" particularly effective for implementing an integrated location-based application.

MWS' typical capabilities are to display maps (render), to find the coordinates of an address (geo-coding), to find the address through given coordinates (reverse geo-coding), to search for addresses within walking distance from user's current location, and to present route instructions.

MWS has four basic services:

- Find Service: finding addresses, attractions, coordinates of points, and points within a distance from the position of the user, depending on the type and how to do the search.
- Render Service: creating the image of the map, based on the data we want
- Route Service: creates a path between two locations we have stated. The type of the route and the points from which the path will pass, depends on the data we provide.
- Common Service: contains classes, which are common in all three previous services, such as the definition of DataSource.

The general categories for the attractions available from MWS are:

- Airport
- AncientSite (Archaeological sites)
- Hospital
- MetroStation
- Museum
- ShoppingCenter (Convention Centers)
- Stadium (Stages-Sports Facilities)

The MapPoint Location Server (MLS) detects the current user's location, identifying the mobile device (including simple mobile devices, Pocket PCs, PDAs, SmartPhones and all registered devices). To find the position, many methods are available, such as Cell-ID (Cell Identification), A-GPS (Assisted GPS), triangular methods, etc.

B. Mobile and Server Database

The mobile device's database contains three tables: ID, POI (Point Of Interest) and CAT_POIS (Category of POIs). The first one maintains user's personal credentials, while the last ones user's personal sights and categories, that is those points of interest and categories that user has added on his profile. Server database contains tables with corresponding names. They exchange data with each other, using Merge Replication process of the Internet Information Services (ISS) web server. The exchange of data is accomplished in both directions, meaning that mobile database tables cannot only send data to server, but also may receive data from it.

Microsoft SQL Server Management Studio is used to create server database. And mobile database tables are automatically generated and updated through synchronization.

Table ID has two columns, the username and password. Table POI contains the name, the coordinates and a brief description for locations that a certain user has personally

selected. Table CAT_POIS consists of two columns, the username and poi-catname. The second column stores the name of user's selected general category. Obviously, a certain username may appear in several records of the table, to represent user's multiple selections on the attractions' general categories.

VI. FUNCTIONAL COMPONENTS OF THE MOBILE APPLICATION UNDER STUDY

We will present in detail the major functional parts (classes and variable types) of our mobile application.

A. Application Classes

The application consists of seven forms and eight code classes. Six of the classes used to connect to the MWS, and perform a different type of work. The seventh class is used for synchronizing server and mobile databases. Finally, the eighth class is used for advanced personalization processes.

1) GetAddress Application Class

This class finds the coordinates of one or more addresses, based on user provided information, such as the address of point (street name and number), or the zip code. Without providing zip code, then most likely to be found more of one addresses, otherwise is found only one. The results are stored on a FindResults variable (MWS variable type).

The class is used during the storage of user's personal attractions. And its purpose is twofold. Firstly, to enable users to choose the desired location-point among others, although they have not entered the zip code and thus several addresses are found. Secondly, to store the coordinates of a point finally chosen by the user, in case that he asks to illustrate the point on the map. In such a case, location's coordinates are ready to be used, and there is no need to search them through the MWS.

2) Generalpois1 Application Class

This class finds all general attractions of a user selected category. Its parameters are:

- entityname: the category of attractions that user has selected,
- distance: the distance over which should be the results desired by the user,
- mylatlongs: holds the coordinates of the user's current location, which will be used if the user wants attractions within a specified distance.

Not setting the distance parameter means that user wants to see all the sights in that category, without distance limits. Another important issue is that the MWS SearchContext parameter cannot limit the search to specific region (e.g. Attica), but only to specific country (e.g. Greece). A viable solution to this issue is by using code statement like the following:

```
If fr.FoundLocation.Entity.DisplayName.IndexOf
("Attiki") <> -1 Then ...
```

This line of code checks over whether found locations are located in Attica. The check is made on whether the found address contains the word Attica.

The search process must be carried out twice, because the size of foundEntities MWS variable must be defined in order to be able to save the results that user desires.

The other case is when user wants the attractions at a specific distance from its current location to be displayed. The parameter distance is expressed in meter units. It should be converted into degrees, so that it can take part in calculations with the attractions' and current location's coordinates.

3) Generalpois2 Application Class

This class accepts the results found by the generalPois1 class and creates the map, with the representation of all the attractions found, plus the user's current location. Depending on the number of results (MWS variable entitiesnum) that have been already found, the size of both mylocations MWS variable (items that will appear) and pushpins MWS variable (the "pins" to be placed on the map and will represent the respective point of mylocations) is defined. Once given the coordinates, then the map is created by using the ViewByBoundingLocations class.

The label parameter of pushpins variable stores a number, which represents the series that the attraction has to the list of results. So, a ListBox user interface control, which stores the name of the attractions, along with the pushpins' label parameter is capable to inform users what exactly represents any numbers displayed on the map.

4) Viewpois Application Class

This class returns an image with the user's current location and the potential various attractions that he personally selected to be displayed. It works just as the generalpois2 class. An important differentiation is the zoom parameter's setting of the views MWS variable.

5) MakeRoute Class

This class creates a path between two points. Depending on the user's selections, various personal attractions can also be presented along this path.

Each time the MakeRoute class is called, it creates a new path, which is part of the journey up the first time that the class was called for this route. Each time the route is divided into segments (sections). In every move, the user's current location is moved to the next segment, to show that the user has moved on set route. In order to have a proper representation of the move, we should have kept the number of parts of the route that was created when the class was invited for the first time on this route.

The MWS variable kindofRoute sets the type of route to be used for the creation of the route. The SegmentPreference.Shortest value creates the shortest route (based on distance), between points, while the SegmentPreference.Quickest value creates the fastest route, based on the time of transition from the beginning to the end.

An example of code to create a certain path follows:

```
route = routeService.CalculateSimpleRoute(latlongs,
    "MapPoint.EU",
    SegmentPreference.Shortest)
```

Three different cases exist: to not display any attraction along the route, to display all the user's personal sights, and to display those user's personal sights, which are in a certain distance from the user. The choice of the case is based on MWS pois variable's value. In general, the movement is divided, for all three previous cases into three sections:

- the movement carried out from beginning to the last but one part,
- the movement carried out in the penultimate part,
- the movement as user has reached the end of the path.

6) MakeRoutePois Application Class

This class creates a path between the user's current location and a certain attraction which is selected by the user. The class is differentiated from the previous MakeRoute class in its way to present the map. The image's center is not the user's current location: what is important here is to display the whole path. For this reason, the previously mentioned MWS ViewByBoundingLocations class is not used. Instead, the MWS variable mapSpec is used to hold the path.

7) Replication Application Class

This class is used to synchronize server and mobile databases. The parameter InternetUrl holds the virtual address, set up on our server, to allow for communication between the bases:

```
replication.InternetUrl =
    "http://192.168.1.2/ReplSync/sqlcesa30.dll"
```

ReplSync is the folder that has been created for the exchange of data, while sqlcesa30.dll is an essential dynamic link library .NET file for exchanging data.

8) AdvPersonalization Application Class

This class is used to apply additional personalization mechanisms which are distinguished as:

- content-based considerations – personal attractions and categories of general attractions are dynamically updated in user profile, based on user interactions with the mobile application. This means that this class monitors user actions and evaluates the frequency of user requests on attractions not belonging in his profile. E.g., even if a particular user has not declared the White Tower attraction in his profile, it may be part of it, if for a certain period of time user frequently asked to see it on his mobile phone. A parameterized threshold is used for deciding the required value of frequency which allows the alteration of user profile.
- collaborative-filtering considerations – server database is queried for examining other users'

profiles. Like-minded users, namely users with many common selections (declared explicitly) on their profiles, are used as a simple recommendation engine: personal attractions and categories of general attractions are dynamically updated in a particular user profile, based on the preferences of other users with similar interests.

B. Application Variable Types

A number of variable types, mainly related to functions of MWS, are used in all classes. Their role in the design of mobile application is of critical importance.

1) FindServiceSoap, RouteServiceSoap Variable Types

These variable types, need to connect to the MWS, and thus before being used they had to be 'activated' by using user name/password combinations, (acquired during registration phase in MWS).

```
findService.Credentials = New
    System.Net.NetworkCredential (myUserName,
    myPassword)
```

For efficiency reasons, PreAuthenticate parameter should be set properly:

```
findService.PreAuthenticate = True
```

2) FindSpecification Variable Type

This object is used by the FindServiceSoap, to carry out searches for Attractions. The parameters that should be set are: the category name of the attractions that we are interested (EntityTypeNames), the map in which the search will be done (DataSourceName), and a number of Options, capable to reduce the volume of returned results. The main options are:

- *Range*: the maximum number of MWS provided results. By default, this number is set to 25. It may be set up to 500.
- *ThresholdScore*: indicates the degree of correlation with the search. Any result "returned" from MWS has such a value, which is stored in *ThresholdScore*. The default option value is set to 0.85. By reducing this value, we increase the amount of results, as we ask practically to reduce the degree of correlation.
- *SearchContext*: an integer that indicates the region, which would limit the search. For example, *DataSource* may specify the map of Europe (MapPoint.EU), but if we are interested only on the results concerning a particular country, then we must set the region option to the corresponding value (e.g., 98 is for Greece).

3) Pushpin() Variable Type

These variables are used to display 'tacks' (pushpins) on the map. A Pushpin represents a mark that we would like to see on the map. Related parameters are:

- *LatLong*: its coordinates.

- *IconDataSource*: a source-repository with images, capable to represent *Pushpins* on maps, as they can be fitted to the point defined by the coordinates. Each user may create his own images, and place them into his personal *IconDataSource*. Or, he may use the default parameter (*MapPoint.Icons*).
- *IconName*: the name of the Pushpin icon, contained in *IconDataSource*. In our application, there are two different icons (namely "0" and "1". Both depict a differently tinted tack ("0" for the blue and "1" for the red one).
- *Label*: the caption that appears above the Pushpin.

4) *ViewByHeightWidth, ViewByBoundingLocations Variable Types*

These types are used for describing two alternative map representations. *ViewByHeightWidth* defines a central point, based on which the map will be shown. There is no guarantee that all the defined Pushpins are being displayed. The map is created so that the center of the provided image is the desired central point (usually user's current location), and the attractions' presence on the map is depended on map's size.

On the other hand, *ViewByBoundingLocations* declares that image map will be created adapted to objects shown on the map. Thus, the map will present all the desired points/attractions.

5) *MapSpecification Variable Type*

This type of variables is used to create maps, based on the provided parameter values. The most significant of them are:

- *DataSource*: the map of MWS.
- *Views*: alternative map representations, using *ViewByHeightWidth*, and *ViewByBoundingLocations* variable types.
- *Options*: stores the size (height and width) of PictureBox control (form element), in which the map will be placed. Map creation is adjusted to the size of the PictureBox. Furthermore, very helpful is the Zoom parameter. Its default value is 1, while the lowest price we can get is 0. It must be noted that Options parameter should be initialized before use.
- *Pushpins*: the potential Pushpins that we are interested to be placed on the map.
- *Route*: in case that we have defined a route (via a specific *Route* MWS variable type), this parameter may cause route design and display on the map.

VII. USING THE LOCATION-BASED SERVICES

In this section, we will present a typical application scenario to demonstrate specific contextual, personalized and user-friendly features of our approach.

Let us suppose that user Alice wishes to see on her mobile the route from her current location to the Airport. Figure 4 depicts the navigation chart of the application, so Alice has to follow the selections' path:

'Map'--> 'Find Route' --> 'To General Attraction'

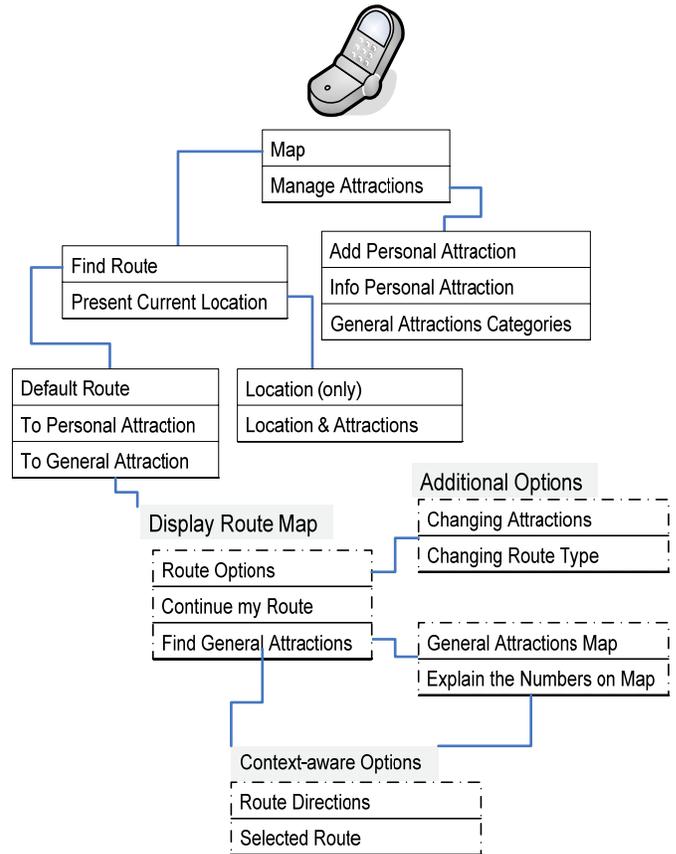


Figure 4. The GUI navigation chart of the application.

Certainly, all attractions belonging to the general categories available from MWS (which are mentioned previously in Section VI), may be used as the destination location of Alice's route. But not only that: Alice may easily choose Airport destination because all categories of attractions that are not of interest for her, are not provided as possible selections on her mobile device. This can be done, because in the past Alice has used the following option to declare her interests:

'Manage Attractions' --> 'General Attractions Categories'

Or, Alice in the past had too many times asked about Airport, and consequently her profile was dynamically updated by AdvPersonalization class, to include this attraction. It must be noticed that, this was possible because Alice in her profile has explicitly allowed the dynamic content-based alteration of her preferences.

Alice, as shown in Figure 5, may see now a map which presents a route according to her preferences, created by MakeRoute class. 'Eiste edo' pin declares her current location, while 'Arxi' (Start) and 'Goneis' (Parents) pins are personal attractions that are close to this route.



Figure 5. Route map display. “Είστε Εδώ” stands for “You are here”, “Arxi” for “Start” and “Goneis” for “Parents”

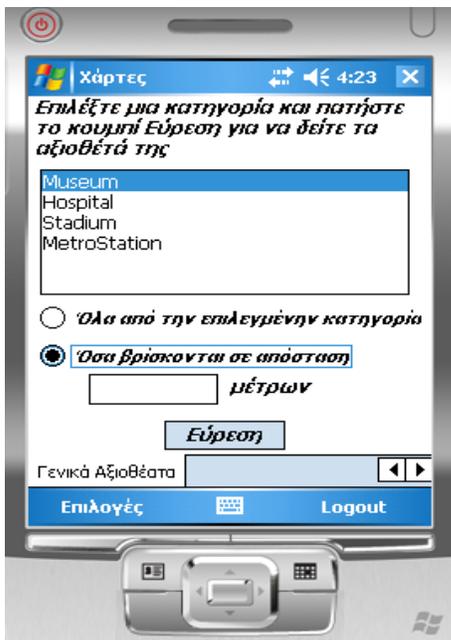


Figure 6. Find general attractions option.

As she moves, Alice decides to stop get informed about the closest personal attractions, or even to change the distance parameter that decides if a personal attraction is close to the route. She just has to select:

‘Route Options’ --> ‘Changing Attractions’

The map will reflect the changes regarding the presence of her personal attractions on the route. If Alice meets accidentally a friend with a car, willing to take her to the airport, she may select:

‘Route Options’ --> ‘Changing Route Type’

In this way, she receives a different route on the map, taking into account the potential one-way streets in this area.

She also may enrich her map, by selecting the *‘Find General Attractions’* submenu. In this way, she may further elaborate her preferences on her stored profile: she may choose which general attractions of her profile (based on their category or on their distance from the specific route) to be displayed on the map (see Figure 6). All these attractions are depicted as numbered pins on the map, but there is a proper menu option to get the explanations about these numbers on the map.



Figure 7. Presenting route instructions.

If Alice decides to get more detailed information on route directions (see Figure 7), she has to select:

‘Context-aware Options’ --> ‘Route Directions’

Also, Alice may ask to see the whole route on her mobile device, using the *‘Selected Route’* option of the *Context-aware Options* menu.



Figure 8. Presenting selected route. “Archeologiko Mousio” stands for “Archaeological museum” and “Eiste edo” stands for “You are here”

In Figure 8, the route to the ‘Archeologiko Mousio’ (Archaeological museum), an attraction of the museum category is presented. Note the difference between Figures 5 and 8: user’s current location (‘Eiste Edo’) is no longer at the center of the display to depict the whole selected route.

It must be noticed also that Alice had not declared this attraction or its category (museum) in her profile and also she had never asked information about it. However, other users with similar declared attributes in their profiles had chosen this attraction and thus her profile was dynamically updated by AdvPersonalization class. This was done because Alice in her profile has explicitly allowed the dynamic collaborative-filtering alteration of her preferences.

The application has similar functionality when users requesting route maps with destinations being personal attractions. It must be also clarified, that the ‘Default Route’ is making use of a specific destination location, namely ‘Telos’ (end), which has taken its value when user made the registration. In this way, every user has a quick way to ask a route map from his current location to a specific (frequently used) destination.

If Alice has no interest to be informed about route maps, she may select:

‘Map’--> ‘Present Current Location’

With this submenu she has access to the most simple location-based services, that is to display on the map her current location with or without nearby attractions (general and/or personal).



Figure 9. Managing personal attractions.

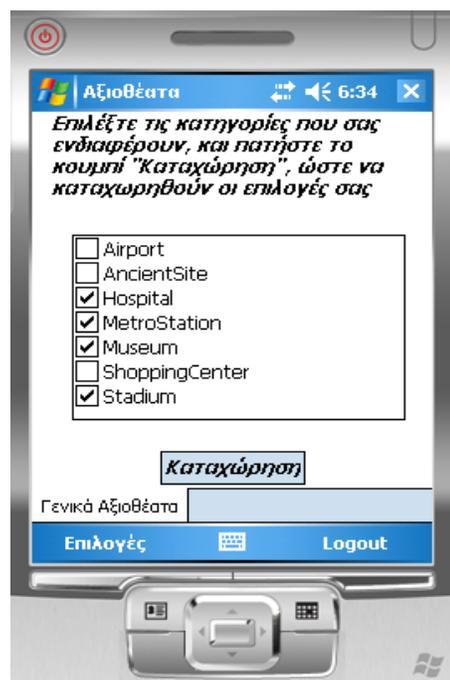


Figure 10. Managing general attractions.

Finally, it must be noticed that ‘Manage Attractions’ option provides powerful administrative functions to Alice, to add, edit, or delete personal attractions (see Figure 9), and/or categories of general attractions (see Figure 10). In this way, Alice may update her profile and enjoy

personalized location-based services. An additional administrative function is the allowance (or not) of the advanced personalization mechanisms, and consequently the dynamic alteration of her profile.

VIII. CONCLUSION AND FUTURE WORK

In this paper we discussed the mobile setting issues, m-commerce adoption concerns, location-based services, context-awareness, and its significance to personalize mobile applications. The design of an indicative end user application is discussed, which makes use of context parameters (such as user's location and preferences), and demonstrates how location-aware context is a powerful enabling factor for any m-commerce application.

In order to demonstrate certain personalized and user-friendly features of our approach, a typical LBS scenario is analyzed and presented. Future research efforts will be focused on investigating a more detailed categorization of users' requirements for LBS, as well as on formulating user's behavior in a flexible context model. Through the use of this context model and the development of more applications, we hope to further increase our understanding of personalization and context-awareness.

ACKNOWLEDGMENT

The author would like to thank Stavros Gitsioudis and Apostolos Provatidis for their contribution in the development of the application case study in this paper.

REFERENCES

- [1] C.K. Georgiadis, "Mobile Commerce Application Development: Implementing Location-aware Information Services", in Proc. of the 5th Advanced International Conference on Telecommunications (AICT 2009), Venice/Mestre, Italy, IEEE CS, pp. 333-338, 2009.
- [2] S.Y. Ho and S.H. Kwok, "The Attraction of Personalized Service for Users in Mobile Commerce: An Empirical Study", ACM SIG eCOM, http://www.sigecom.org/exchanges/volume_3/3_4-Ho.pdf, 22.6.2009, 2003.
- [3] S. Koukia, M. Rigou, and S. Sirmakessis, "The Role of Context in m-Commerce and the Personalization Dimension", in Proc. of the 2006 IEEE/WIC/ACM Intern. Conference on Web Intelligence and Intelligent Agent Technology (WI-IAT), IEEE CS, Washington, DC, pp. 267-276, 2006.
- [4] G. Elliot and N. Phillips, *Mobile Commerce and Wireless Computing Systems*, Addison Wesley – Pearson Education. Harlow, England, 2004.
- [5] M. Chae and J. Kim, "What's so different about the mobile Internet?", *Communications of the ACM*, 46(12), pp. 240-247, 2003.
- [6] C. Ververidis, G.C. Polyzos, and K.-P. Mehdi, "Location-Based Services in the Mobile Communications Industry", *Encyclopedia of E-Commerce, E-Government and Mobile Commerce*, Idea Group Reference, Hershey, USA, 2006
- [7] M. Kakihara and C. Sorensen, "Expanding the "mobility" concept", *ACM SIGGROUP Bulletin*, 22(3), pp. 33-37, 2001.
- [8] M. Kakihara and C. Sorensen, "Mobility: An Extended Perspective", in Proc. of the 35th Hawaii International Conference on System Sciences, 2002.
- [9] Y.E. Lee and I. Benbasat, "A framework for the study of customer interface design for mobile commerce", *International Journal of Electronic Commerce* (1086-4415/2004), 8(3), pp. 79-102, 2004.
- [10] C.K. Georgiadis, "Adaptation and Personalization of User Interface and Content", Chapter in "HANDBOOK OF RESEARCH ON MOBILE MULTIMEDIA", I.K. Ibrahim (Ed.), Information Science Reference Inc. (IGI Group-Idea), ISBN 1-59140-866-0, pp. 266-277, May 2006.
- [11] M. Bina, D. Karaiskos, and G.M. Giaglis, "Motives and Barriers Affecting the Use of Mobile Data Services", in Proc. of the IEEE International Conference on Mobile Business (ICMB 2007), 2007.
- [12] J.H. Wu and S.C. Wang, "What drives mobile commerce? An empirical evaluation of the revised technology acceptance model", *Information & Management*, 42(5), pp. 719-729, 2005
- [13] F.D. Davis, "Perceived usefulness, perceived ease of use, and user acceptance of information technologies", *MIS Quarterly* 13(3), pp. 319-340, 1989.
- [14] V. Venkatesh and F.D. Davis, "A theoretical extension of the technology acceptance model: four longitudinal field studies", *Management Science* 46(2), pp. 186-204, 2000.
- [15] ISO/IEC 9126-1, "Software product evaluation –quality characteristics and guidelines for the user", Geneva: International Organization for Standardization, 2001.
- [16] J. Garofalakis, A. Stefani, V. Stefanis, and M. Xenos, "Quality attributes of consumer-based m-commerce systems", *IEEE Int. Conf. on E-Business and Telecommunication Networks, ICETE, ICE-B, Barcelona, 2007*, pp. 130-136.
- [17] C.K. Georgiadis, I. Mavridis, and A. Manitsaris, "Context-based Humanized and Authorized Personalization in Mobile Commerce Applications", *IJCIS*, Vol. 3, No. 2, pp. 1-9, 2005.
- [18] K. Virrantaus, J. Markkula, A. Garmash, V. Terziyan, J. Veijalainen, A. Katanosov, and H. Tirri, "Developing GIS-Supported Location-Based Services, in Proc. of WGIS'2001 – First International Workshop on Web Geographical Information Systems, Kyoto, Japan, pp. 66-75, 2001.
- [19] S. Steiniger, M. Neun, and A. Edwardes, "Foundations of Location Based Services", *Lecture Notes on LBS*, Department of Geography, University of Zürich, http://www.geo.unizh.ch/publications/cartouche/lbs_lecturenotes_steinigeretal2006.pdf, 25.6.2009, 2006.
- [20] K. t. Hagen, M. Modsching and R. Kramer, "A location-aware mobile tourist guide selecting and interpreting sights and services by context matching", in Proc. of the 2nd Annual Int. Conf. on Mobile and Ubiquitous Systems: Networking and Services (MobiQutous 2005), San Diego, California, 2005.
- [21] S. Boll, J. Krosche, and A. Scherp, "Personalized Mobile Multimedia meets Location-Based Services", in *Dadam, P., Reichert, M., eds.: Workshop "Multimedia - Information systems" at the 34th Annual meeting of the German Computer Society (INFORMATIK 2004)*, Vol. 51 of LNI., Ulm, Germany, GI, pp. 64-69, 2004.
- [22] C.K. Georgiadis and S.H. Stergiopoulou, "Mobile Commerce Application Development: Implementing Personalized Services", in Proc. of the International Conference on Mobile Business 2008 (ICMB 2008), Barcelona, Spain, July 2008, IEEE CS, pp.201-210, 2008.
- [23] MapPoint Web Service, Microsoft Corp., <http://www.microsoft.com/MapPoint/en-us/default.aspx>, 20.7.2009, 2008.