Pers Ubiquit Comput (2006) 10: 227–240
DOI 10.1007/s00779-005-0044-5

ORIGINAL ARTICLE

Stan Kurkovsky · Karthik Harihar

Using ubiquitous computing in interactive mobile marketing

Received: 5 October 2004 / Accepted: 15 April 2005 / Published online: 29 September 2005

Abstract Unique features of handheld devices, including their mobility, personalization and location-awareness engender new types of applications for mobile commerce, such as mobile advertising. Mobile marketing and advertising applications deliver promotional information to consumers based on their preferences and location. In this paper, we present SMMART, a context-aware, adaptive and personalized m-commerce application designed to deliver targeted promotions to the users of mobile devices about the products they like while guarding the users' identity and protecting them from any unsolicited messages. Promotions distributed by SMMART are personalized by performing intelligent matching of the user's shopping interests to current promotions available at a retail site. SMMART can adapt to changing preferences of its user by inconspicuously monitoring his or her shopping habits. We describe a fully functional prototype of SMMART built for Pocket PCs running Windows CE with .NET Compact Framework. This paper also presents a study demonstrating end-user usability and economic viability of SMMART.

Keywords M-commerce · Ubiquitous computing · Context-aware applications · Personalization · Wireless networks

1 Introduction

The process of buying, selling and promotion of information, services and products via computer networks is commonly known as electronic commerce, or e-commerce. Mobile commerce, or m-commerce, is a special area of e-commerce, where mobile devices are used for buying, selling or advertising. Wireless mobile devices have a range of unique features, which impact the nature and specifics of applications designed for them. Such features of mobile devices include their ubiquity (they are affordable and portable), personalization (a typical PDA belongs to and can be identified with a single individual) and location-awareness (a connection established in the wireless medium can be used to determine the physical location of the device) [1]. Many existing e-commerce applications have been adapted to run on wireless mobile devices. At the same time, there is a new class of m-commerce applications, which is becoming possible due to the unique features of hardware, such as mobility and location-awareness inherent to small personal computing devices, such as PDAs and smart phones. A similar trend may be observed not only in m-commerce, but also in many other areas of applications for mobile devices. While some old applications are adapted for mobile platforms, the features inherent to their mobility and wireless communication medium create a unique class of emerging novel applications striving to achieve the anytime, anywhere paradigm of ubiquitous computing. Thus, many believe that m-commerce will serve as one of the major forces driving the development of wireless Internet and related services [2, 3].

This paper focuses on a prototype of SMMART framework—a *S*ystem for *M*obile *M*arketing: *A*daptive, Pe*R*sonalized and *T*argeted, whose early implementation is described in Ref. [4]. Our goal is to build a personalized and context-aware mobile marketing application for wireless devices, such as a PDA, delivering narrowly targeted promotional information to their users. The main objective of SMMART is to help the shopper navigate through many offers or promotions available at a given retail store by directing the user's attention to those products that match their interests. At the same time, SMMART ensures the shopper's privacy: it does not transmit any personally identifiable information and guarantees that the user will receive information about products relevant to their shopping preferences. SMM-

S. Kurkovsky (✉) · K. Harihar
Computer Science Department, Central Connecticut State Uni-versity, New Britain, CT, USA
E-mail: kurkovskysta@mail.ccsu.edu
E-mail: harihar_karthik@colstate.edu

ART is a context-aware application, delivering promotions to the shoppers when they are in a close proximity or inside a participating retail site distributing promotions. SMMART can adapt to the interests of its user by unobtrusively monitoring his/her shopping habits and learning the user's personal preferences as they change over time, thereby maintaining a continually changing user model.

This paper is organized as follows: Section 2 describes the related work in the area of context-aware and location-based applications, as well as some current research issues in mobile user interfaces. Section 3 describes a short-term and a long-term scenario of using SMMART. Sections 4 and 5 describe the architecture of SMMART framework and the implementation of its fully functional prototype. Section 6 discusses the end-user experiences with using our prototype of SMMART framework and presents a study of its economic viability. Section 7 concludes the paper and presents possible directions for future work.

2 Related work

In this section, we describe some existing work in several areas that are closely related to our work. First, we discuss current research and open issues in location-based applications and mobile advertising and then we present some important considerations for designing mobile user interfaces for m-commerce applications. This section is concluded with a brief overview of the current research in personalized recommendation systems.

2.1 Location-based mobile advertising

A comprehensive classification of existing and future m-commerce applications is presented by Varshney and Vetter [5]. As a category within such a classification, mobile advertising applications generally take advantage of demographic or other data provided by consumers, which are used for the targeted delivery of promotional information [6]. Some of these applications are location-based; they deliver advertising messages only to those users who are physically located near the retailer whose products or services are being advertised [7]. A problem inherent to most location-based applications stems from the fact that their coverage area depends on the precision with which the user location is determined; this, in turn, is determined by the network technology used for wireless connectivity or the type of sensors used for navigation and identification of the user location. In order to avoid congestion of a wireless network, each mobile advertising application should cover a small geographic area. At the same time, such applications should avoid overwhelming consumers with a large number of irrelevant advertising messages by limiting their broadcast coverage and carefully selecting promotional messages and their intended recipients.

Current research and applications in location-based mobile advertisement have been surveyed by Ratsimor et al. [8] and Varshney [7]. All applications of this class use one of the two modes of information delivery: push (promotional information is automatically sent to the consumers) or pull technology (promotions are sent at the consumer's request). A large number of current mobile advertising applications use short message service (SMS) to push advertisements to consumers. As a rule, consumers must subscribe to such a service by specifying what kind of information or promotions they would be interested in. i-Mode supported by NTT DoCoMo is widely used in Japan for similar purposes. Mobile Marketing Association has been formed in 2002 by merging the Wireless Advertising Association and the Wireless Marketing Association to promote the creation of uniform standards for mobile advertising.

Most of the existing location-based services are enabled by determining absolute physical geographic location; some of these services are used to provide routing, mapping and related information. Several technologies can be used to determine the location of a mobile device and its user. In particular, as mandated by the US government, by the end of 2005, all mobile phones must be able to support E911 requirements so that they can be pinpointed within a small radius. Most CDMA phones will be equipped with Assisted or Differential Global Positioning System technology (A-GPS and D-GPS), while GSM networks will be upgraded with Uplink Time Difference of Arrival technology (U-TDOA).

A context-aware system operates and adapts itself based on the knowledge about its user's state and physical surroundings [9]. As a special case of location-based ubiquitous applications, context-aware systems are enabled by the same location-sensing technologies, such as wireless radio frequency (RF) and GPS sensors, Bluetooth and Wi-Fi wireless LANs. One of the methods to obtain location information without gathering precise geospatial data is to position a provider and a consumer of a service (in client/server architectures) or two or more peers (in peer-to-peer architectures) in physical proximity of each other. Due to their limited transmission range, context-aware services enabled by wireless LAN technologies can only reach customers located within a close physical proximity of the wireless service provider and therefore solve the problem of potential network congestion, which was mentioned earlier.

eNcentive framework described by Ratsimor et al. [8] is a context-aware m-commerce application used to distribute electronic coupons in a peer-to-peer environment. eNcentive works in both infrastructure and ad hoc wireless networks. However, it pushes all available coupons to its users regardless of their preferences. In order to be effective, eNcentive must be deployed at a large number of retail sites and requires an even larger number of participating customers carrying wireless PDAs. Ad-me [10] is a context-aware advertising system built on top of a mobile tourist guide that utilizes user profiles and location information to deliver personalized advertisements. However, Ad-me requires a GPS receiver for positioning. B-MAD [11] is a location-based mobile advertising system that delivers permission-based advertisements as WAP messages and uses Bluetooth technology for positioning. These frameworks generally do not consider the user's interests and push all available advertising information to them.

2.2 Mobile user interfaces

The success of an m-commerce application can be measured by the change in shopping patterns (increased number of purchases), which is determined by constraints of the mobile device and current mobile environment of the user. Hardware capabilities of the mobile device directly influence the features and quality of mobile applications. One of the greatest challenges in developing successful applications for mobile devices lies in designing a suitable user interface. Small screen size does not permit including rich graphic elements or large amounts of text information in the application interface. As pointed out by Jones et al. [12], everyday user tasks, such as accessing a search engine on a mobile device versus on a wired desktop may result in completely different user experiences. Wireless network connectivity on mobile devices may be limited in its availability and bandwidth, which imposes significant restrictions on the amount and speed of interactions in networked applications.

The mobile user environment may be viewed as consisting of the following three aspects described by Kannan et al. [1]. Spatial mobility refers to the ability of the user to roam anywhere with their mobile devices. When considering networked applications, spatial mobility is limited to the areas of wireless network connectivity. Temporal mobility implies that people typically use mobile devices while engaged in other simultaneous activities. Contextual mobility describes the fact that the users may use their mobile devices differently and mobile applications may behave differently depending on the current context. Here, application context refers to current location, time, user identity, other users and objects [13–15].

The three dimensions of mobile user environment demand a special consideration in the area of m-commerce applications. As the most crucial part of such an application, interface design must support the limited attention of the user, who is often distracted by other events, applications and objects within the three dimensions of mobility. For example, shoppers using m-commerce applications while at a retail store may be distracted by other shoppers, print and multimedia advertisements, all of which limit the amount of attention given to the information on the user's mobile device. Ergonomic factors may make this situation even more complex [16]: for example, how will a shopper hold a PDA and use stylus to navigate through the application while pushing a shopping cart and possibly looking after a child? Lee and Benbazat [17] suggest focusing on the seven design elements (context, content, community, customization, communication, connection, commerce) as a guide to implementing a successful customer interface for an m-commerce application. By following these guidelines, m-commerce user interfaces should have a streamlined navigational structure, no complex menus, simple and descriptive pages with only highly relevant information and an ability to connect on-screen information with the physical world.

One of the goals of many m-commerce applications is to increase the number of buying opportunities by connecting a shopper with a product (or product information) of potential interest to the shopper, which can only be successful if the shopper is truly interested in that product. Two major factors are responsible for establishing the shopper's interest: the product must be relevant to the shopper's preferences and the format in which the product information is presented must catch the shopper's attention given the distracting mobile environment [3]. First, a correct match between the buyer's shopping preferences and the products in the inventory must be established. Here, one of the most obvious choices is to present the shopper with a list of products available at the store and let him navigate through it. Alternatively, the buyer could construct his or her shopping list at home and use it by checking off the products while at a store. In fact, shopping list applications were one of the first existing m-commerce applications for PDAs. However, these solutions are very far from perfect because the product choice is left to the buyer and many possible buying opportunities may be overlooked [16]. One way to overcome this problem is to use an m-commerce system to match products to the buyer's shopping preferences by using keywords or key phrases in the product description [12]. Such an approach can also be viewed in the light of context-aware information filtering and retrieval, where corresponding product information is matched to the current shopping preferences of the user [18]. Secondly, product information should be presented to the shopper in such a way that the user can clearly see it on the small screen while possibly being distracted by many environmental factors. For example, for a system making product suggestions, large amounts of information about many products should be limited and only the highly relevant information must be displayed with the most important data (such as the product name and pricing) clearly highlighted. Related products can also be grouped into categories to conserve screen space and help the user efficiently navigate through the application interface [12].

This paper describes SMMART framework, which belongs to the same class of applications as eNcentive, Ad-me and B-MAD applications mentioned earlier in

Sect. 2.1. SMMART can be used by retailers to distribute targeted marketing information to those customers whose preferences match the descriptions of products that are currently on sale. The user interface of SMMART is designed keeping in mind many considerations discussed in Sect. 2.2. The navigation structure of SMMART is simple and straightforward; the user is always presented with concise and highly relevant information. The framework is designed to deliver the information that its users would like to receive, and therefore, by design, it cannot be used as a tool to flood consumers with spam. SMMART guarantees a high level of privacy because it does not transmit any personally identifiable information; the framework cannot be used by retailers to track customers and their buying habits.

2.3 Recommendation technologies

Recommendation (or recommender) systems are typically used to predict what items (such as retail goods, movies, news stories, Web sites, vacation packages) a given user may be interested in, based on some information about that user, as well as, possibly, other users. The objective of any recommendation system may be viewed as the assignment of a certain user-specific rating value to each item of interest. Methods used by recommendation systems can be classified as collaborative, content-based and hybrid systems, which combine the former two methods. A good overview of these techniques can be found in Ref. [19].

In content-based recommendation systems, the rating of each item is established based on the similarity of the given item's content to the content of other items, whose ranking is already known. One of the most frequently used techniques of describing content is by using keywords. Each keyword is usually associated with a weighting measure that can be determined by one of the information retrieval techniques, such as TF-IDF. Each user of such a system is associated with a profile consisting of a vector of weighed keywords. Cosine similarity measure can be used if each item of interest is also described as a vector of weighed keywords. Other techniques used in content-based recommendation systems include Bayesian classifiers, clustering, decision trees and artificial neural networks. Relevance feedback method is often used to improve the quality of item rating by adding keywords and adjusting their weights; this also improves the ratings of relevant items.

Collaborative recommendation systems use collaborative filtering algorithms to determine the rating of an item for the given user based on how the same item was rated by other users who are similar to the given user. There are two classes of algorithms used by collaborative recommendation systems. Memory-based algorithms predict ratings of items based on the entire collection of items that were previously rated by all users. Model-based algorithms use the entire collection of ratings to learn a certain model, which is used to determine the rating of new items. Collaborative recommendation systems often suffer from the "new item problem," which means that the system will be unable to recommend a new item until that item is rated by a number of users. Another common problem arises due to the "sparsity of rating," which happens when the number of ratings for some items is small compared to the number of ratings needed for those items to be recommended.

SMMART belongs to the class of content-based recommendation systems. It employs user profiles consisting of weighed keywords. The prototype of SMMART presented in this article uses a simple rating algorithm based on the Boolean model, which is discussed in detail further.

The next section presents two scenarios illustrating a typical use of SMMART by a hypothetical shopper named Bob.

3 Scenarios of using SMMART framework

Bob recently installed SMMART Client software onto his wireless PDA. Bob entered his musical preferences into SMMART (they include *Cranberries*, *Dire Straits* and *Police*). As Bob enters his favorite music store, the SMMART Client on his PDA establishes a connection with the store's SMMART Server and tells the server about his preferences. The server responds with a list of products that may be of interest to Bob. This list includes only those products that match Bob's preferences and are currently on sale. Thus, Bob is spared from having to browse through the weekly advertising brochure with no guarantee that he will find anything interesting. As illustrated in the left part of Fig. 1, of all the products that are on sale that day, only two match Bob's interests: *Every Breath You Take*, a compilation of greatest hits by *Police*, and *On Every Street*, a CD by *Dire Straits*.

Bob selects *Every Breath You Take* on the screen of his PDA to view more information about the promotion. As Bob clicks on this product, his SMMART Client assumes that Bob may be interested in other products by *Police* and its musicians. In this case, keyword *Sting* (the lead singer of *Police*) is automatically added into the list of Bob's preferences. Bob also decides to purchase *On Every Street* by *Dire Straits*. Clicking on this product listed in his SMMART Client has two consequences: Bob's interest in *Dire Straits* is confirmed and the keyword *Mark Knopfler* (the founder of *Dire Straits*) is added to his preferences.

After visiting the music store, Bob decides to visit a nearby video store to look for some DVDs. At this moment, Bob's preferences include five keywords (as illustrated in the right part of Fig. 1. Upon entering the store, his SMMART Client receives all current promotions matching his interests, presenting Bob with three products. Bob is most interested in *Lock, Stock and Two*

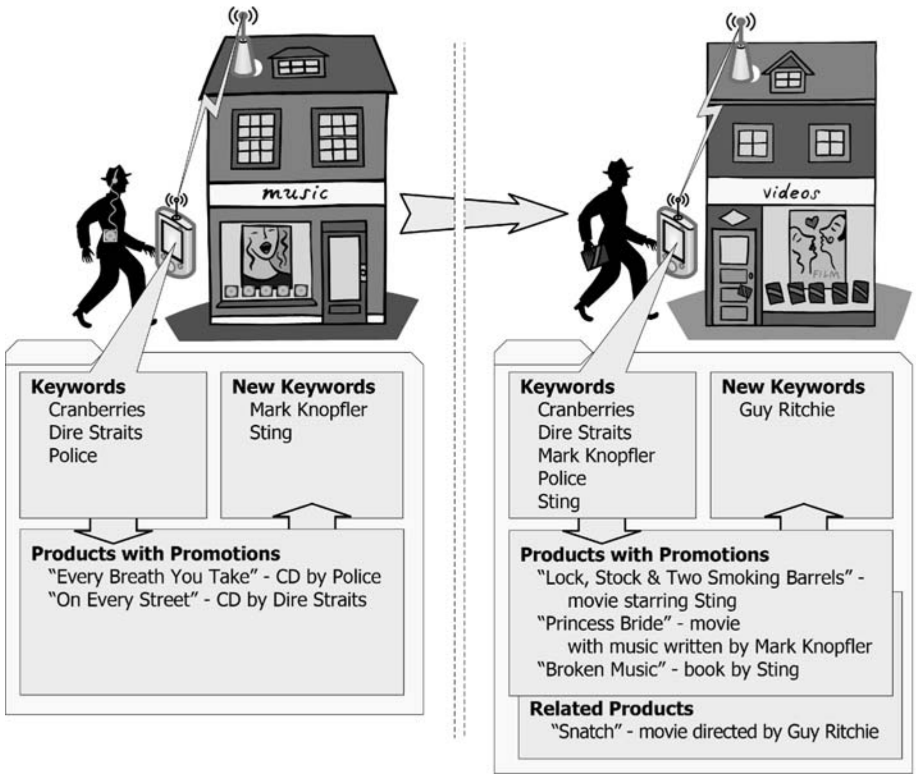

Fig. 1 A typical short-term scenario of using SMMART in two retail stores

Smoking Barrels, a movie starring *Sting*. As Bob selects this product using his SMMART Client, Bob's interest in *Sting* is confirmed and his preferences are updated with a new keyword corresponding to the displayed product, *Guy Ritchie* (director of this movie). Bob's SMMART Client also offers a list of related products, which includes *Snatch*, a movie directed by *Guy Ritchie*.

As the days pass while Bob continues using SMMART during his shopping, his SMMART Client continuously monitors Bob's product browsing patterns. Throughout the entire process of Bob's interaction with the software, SMMART Client analyzes the products that Bob selects and assumes that such an action indicates Bob's interest in that product. Since each product is associated with one or more keywords pulled from the store's inventory, SMMART Client uses them to continuously update Bob's preferences and therefore adapt itself to his evolving shopping interests. Eventually, keywords that occur most frequently will have a higher influence in the algorithm employed by SMMART to match products to Bob's preferences. On the other hand, keywords, which are not encountered frequently in the products that Bob browses, will eventually be removed from his preferences and will play no role in the way products are chosen by SMMART. For example, if Bob has a sustained interest in *Sting*, it is very likely that his preferences stored in his SMMART Client will reflect this artist, all of his work and some other artists with whom *Sting* collaborated. Having such a representation of Bob's shopping interests, SMMART will continue suggesting products that are one way or another related to *Sting* (assuming that there are offers for these products in the context of a given retail store). On the other hand, if Bob's interest in *Guy Ritchie* is superficial and not permanent, SMMART initially may find some products related to this director, but they will always be ranked lower in the list of products suggested by SMMART. After some time, if Bob stops paying attention to these products, SMMART will eventually stop suggesting *Guy Ritchie*'s and related products. Finally, the keyword, *Guy Ritchie*, will be eliminated from the list of Bob's shopping preferences.

4 Architecture of SMMART framework

SMMART framework consists of a server installed at every participating retail location and a client running on a mobile wireless device, as shown in Fig. 2.

4.1 SMMART Client

As shown in Fig. 2, the SMMART Client consists of two main components: *SMMART Browser* and *Preferences* module.

SMMART Browser enables user interaction with the SMMART framework. Typically, after SMMART Client connects to and communicates with a store's server, the user is presented with a list of products matching the user's preferences that currently have some promotions. The user may explore each product in detail by viewing

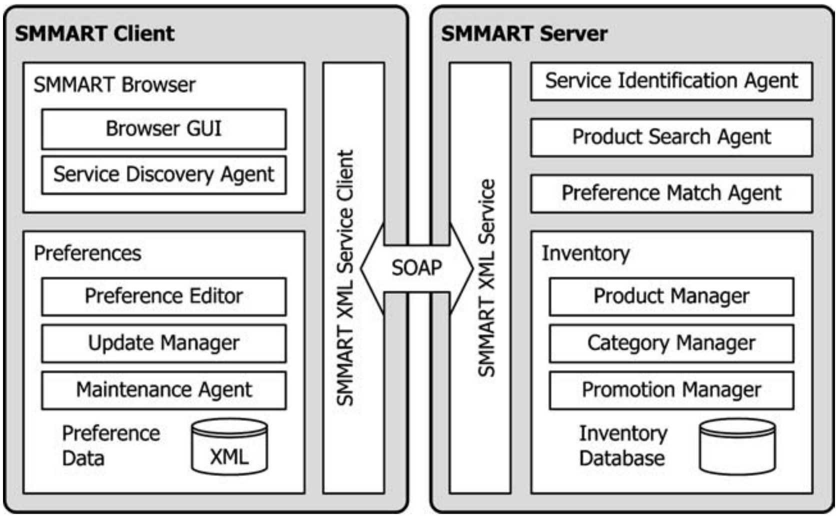

Fig. 2 Architecture of SMMART framework

the product information, offer details and related products using *SMMART Browser*.

Preferences module contains and manipulates the list of keywords stored in XML format in *Preference Data*, as shown in Fig. 3. Each keyword is associated with a numeric weight representing its relevancy to the user and a date stamp indicating the last time this keyword or its weight was updated. As described below, keyword weights and date stamps enable learning capabilities of SMMART framework.

When the user runs SMMART Client, the *Preference Data* is scanned for keywords. If no keywords are found, the user is prompted to populate *Preference Data* with new keywords by using *Preference Editor*. The main responsibilities of this module include adding new, editing and deleting existing keywords and, if needed, manually changing the weights of existing keywords.

All product information displayed by *SMMART Browser* is received from the SMMART Server, to which the client is currently connected. Each product is associated with one or more keywords, which are used by SMMART Client in the process of inference of user preferences. When the user indicates an interest in a particular product by viewing its full description in *SMMART Browser*, *Preference Data* is automatically updated by the *Update Manager*. If the corresponding keywords associated with a product are already present in *Preference Data*, their weights are incremented. Otherwise, they are added into the *Preference Data* with a default weight. Fig. 4 illustrates the algorithm used to update the SMMART user *Preference Data*. By dynamically incrementing the keyword weights each time the user clicks on a product in the browser, SMMART Client adapts itself to the continually changing preferences of the user. The implicit feedback approach used by SMMART is simple, yet it provides a viable mechanism to prove the concept of such an application as SMMART. This approach can be further improved by allowing the user to provide relevant feedback by ranking the returned results by their relevance. Furthermore, this method can be strengthened by adding negative feedback, i.e. allowing the user to specify the keywords or product matches that he or she does not like. By implementing dynamic algorithms of keyword inference and keyword decay (described below), SMMART maintains its user model based on the analysis of the use behavior patterns in the context of location and user history.

Keyword choice is very important for enabling the proper representation of the user's preferences. Each product must be associated with a keyword that not only describes that particular product, but also some other products related to it. For example, all musical CDs by a particular artist should be described by the same keyword—the artist's name. This way, if the user indicates a potential interest in one album, other albums by the same artist may be suggested. Eventually, this artist may become one of the user's favorites. To break out of a cluster of products connected through the user's preferences, SMMART always makes recommendations on

```
<?xml version="1.0" encoding="utf-8" ?>
<preferences>
   <keyword name="Sting"       weight="9" date="9/22/2004" />
   <keyword name="Guy Ritchie" weight="3" date="8/30/2004" />
</preferences>
```

Fig. 3 Sample user *Preference Data* stored in SMMART Client

```
function UpdateKeywords
  in Keywords : keywords to be added/updated in the user preferences
  in/out Preferences : current preference data
begin
  KeywordEntry : keyword record in the preference data format
                 (includes keyword itself, its weight and date stamp)
  for each Kword in Keywords do
    if Kword in Preferences then
      KeywordEntry.Weight = KeywordEntry.Weight + 1
      KeywordEntry.Date = today
    else
      KeywordEntry.Keyword = Kword
      KeywordEntry.Weight = default weight for a new keyword
      KeywordEntry.Date = today
      Insert KeywordEntry into Preferences
    end if
  next for
end function
```

Fig. 4 Algorithm for updating the *Preference Data* with new or existing keywords

other related products. Such relationships among different products are stored in the product inventory, which is described in the next section. Many popular e-commerce Web sites, such as Amazon.com, make recommendations based on what products other shoppers bought. Such relationships among products can be established through market basket analysis, but the specifics are beyond the scope of this work.

It is possible for keywords not representing true user interests to be automatically added to *Preference Data*. SMMART assumes viewing the full product description indicates the user's interest in the product. However, it is possible that the user could click on a product by mistake or determine that the product is of no interest upon viewing its full description. In this case, *Maintenance Agent* has a chance to decrease the weight of such keywords. *Maintenance Agent* runs at periodic intervals, scans *Preference Data* for keywords that have not been updated within a specified period and decreases their weight. Such keywords are usually erroneous or represent past interests of the user. *Maintenance Agent* also takes into account some other factors, such as how frequently SMMART is being used for shopping and how many interactions the user has with the system each time. Generally, as the frequency of use increases, so does the ability of SMMART to maintain the most up-to-date set of preferences that closely match the true interests of the user. The weights of keywords that are never used gradually decrease and reach the value of zero, at which time they are placed in the *Recycle Bin* and are no longer used for matching. When the user runs *Preference Editor*, he can review the contents of the *Recycle Bin* and restore or purge its contents in part or in full. Keywords restored from the *Recycle Bin* go back to the *Preference Data* with the current default or user-specified weight.

SMMART Browser also allows the user to search the store's inventory directly without matching the products against shopping preferences. In basic search, the user specifies one or more keywords, which will be used by a conjunctive query to match product descriptions in the inventory. Advanced search is modeled after a similar feature of Google and contains four search fields that allow searching products "with all of the words," "with the exact phrase," "with at least one of the words" and "without the words." Implementation of these search features is described in Sect. 5.

4.2 SMMART Server

An *Inventory Database* available at every typical retail store provides the basis for all data available to the SMMART framework. Three functional modules work directly with the *Inventory Database*: *Product Manager, Category Manager* and *Promotion Manager*. *Product Manager* retrieves all relevant information about a specific product, which is then used by the SMMART Client when the user chooses to view the details about a particular offering. Similarly, *Promotion Manager* retrieves all promotion information for a given product. *Category Manager* helps organize products into hierarchical categories and retrieve them accordingly.

Preference Match and *Product Search Agents* form the core of the SMMART Server. *Preference Match Agent* receives a list of user preferences from the SMMART Client. This is a list of keywords ordered by their relevancy to the user's interests. For each keyword in the list, SMMART Server finds all the matching products that currently have a promotion and adds them to the list of results, as illustrated in Fig. 5. While it is possible that the same product can be found using more

```
function MatchPreferences
  in Preferences : ordered list of preferences
  out Result : ordered list of products
begin
  Kword : keyword
  for each Kword in Preferences do
    NewResult : list of products
    NewResult = result of inventory search for all products matching Kword
    NewProduct : single product from the inventory
    for each NewProduct in NewResult do
      if (NewProduct not in Result) and
         (NewProduct has a current promotion) then
        add NewProduct to end of Result
      end if
    next for
  next for
  return Result
end function
```

Fig. 5 Algorithm for matching user preferences against current offers and products in the store's inventory

than one keyword, only one entry is added to the result using the maximum weight of all matching keywords. Due to the nature of the matching algorithm, the list of results consisting of matching products is automatically sorted in the order of their relevance to the user preferences. Finally, this resulting list of products is returned to the SMMART Client. SMMART employs a simple matching process that is based on the Boolean model. Such a choice is determined by the fact that there is a relatively small amount of text information in product descriptions against which the keywords are matched. Using other information retrieval models, such as TF-IDF weighting and probabilistic model, could prove more effective and result in a more precise matching especially when dealing with large product descriptions. However, discussing these approaches falls beyond the scope of this article.

Product Search Agent has two primary functions: it assists the *Preference Match Agent* in finding products in the inventory that match individual keywords and also serves as a mechanism for querying the store inventory using the Search and Advanced search features of the SMMART Client.

4.3 SMMART Client/Server communication

SMMART framework can be deployed at one or several retail sites. However, from the viewpoint of SMMART framework, each site is considered independently. To enable SMMART functionality, each site must be equipped with one or more wireless access points connecting SMMART clients to the corresponding store servers. These access points must route data traffic only to the SMMART Server's Web services, as illustrated in Fig. 6. Such a routing must be strictly enforced; otherwise, these access points would enable wireless clients to hijack the provided bandwidth. Fig. 6a shows the framework configuration in a stand-alone store, where the inventory database is hosted on the same network as SMMART Web services. Fig. 6b shows a typical configuration for a chain of stores equipped with SMMART, all of which connect to a single inventory database.

We chose XML Web Services [20] for communication between SMMART clients and servers for several reasons. First, the architecture of a typical Web service fits well in the general philosophy of SMMART. A number of functionally and semantically related methods are united under the umbrella of a single service, all of which work with the same data (store inventory). Secondly, Universal Description, Discovery and Integration (UDDI) entries facilitate the discovery of each individual store's service by their clients and provide an easy way to identify existing Web services. We assume that each SMMART site routes all network traffic from the access point only to

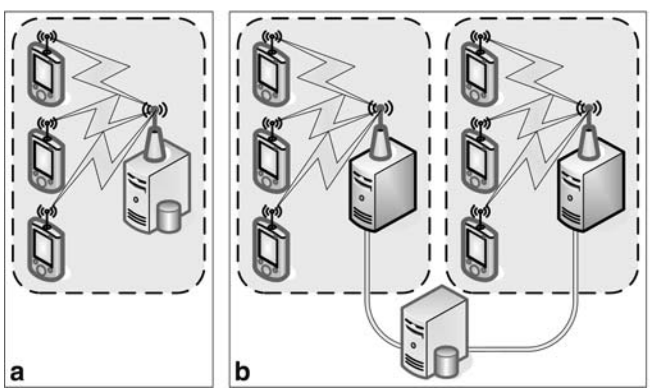

Fig. 6 Possible network configurations for stores equipped with SMMART

the Web server hosting SMMART Web services. This enables an unambiguous identification of the store, to which the SMMART Client is connected, and the availability of the corresponding SMMART services. Thirdly, using XML Web Services helps SMMART framework overcome the burden of possible network disconnections. This is because XML Web Services follow the stateless connection paradigm, i.e. the server does not maintain the record of, nor does it require, any particular sequence of the service invocations.

4.4 SMMART service discovery

Service Discovery Agent, a part of *SMMART Browser*, establishes communication with the corresponding store's server. Whenever a physical network connection is established, *Service Discovery Agent* attempts to call a specialized discovery method within the SMMART Server's Web service. The *Service Identification Agent*, residing on the SMMART Server, monitors all such incoming requests and confirms the connection to the server by returning the name of the store and/or any additional data used to uniquely identify each SMMART site.

The ranges of two or more wireless access points belonging to different SMMART sites may overlap. In this case, *Service Discovery Agent* of the SMMART Client requests information from all available *Service Identification Agents* of stores within the connection range and displays the corresponding store names to the user. The user identifies the store where he is currently located, after which the SMMART Client communicates only with that store's SMMART Server.

5 SMMART framework implementation

To prove the functional feasibility of the SMMART framework, we implemented a fully operational prototype. We chose C# and ASP.NET to implement SMMART Server running on Microsoft .NET framework because of its streamlined support for Web services and easy database connectivity. SMMART Client is also implemented in C# running on Microsoft .NET Compact framework. Our inventory database runs on SQL Server 2000. The remainder of this section illustrates our current implementation of the SMMART Client prototype for a Pocket PC running Microsoft Windows CE.

Once the user is in the vicinity of a store with a SMMART Server and starts the client application, the SMMART Client checks for any stored preferences of the user and if it does not find any, it gives the user an option to manually add preferences, as shown in Fig. 7. Offers displayed by the SMMART Client are always reflective of the current state of preferences in the *Preference Data*, which can be manipulated using the *Preference Editor*; it allows the users to add, modify or delete their preferences. New preferences are added with a default weight (unless specified otherwise by the user) and the current date. The user can delete any unwanted preferences or update the keyword and/or the weight of an existing preference. By default, when presented on the screen, the preferences are sorted alphabetically. The user also has an option to sort them by their weights.

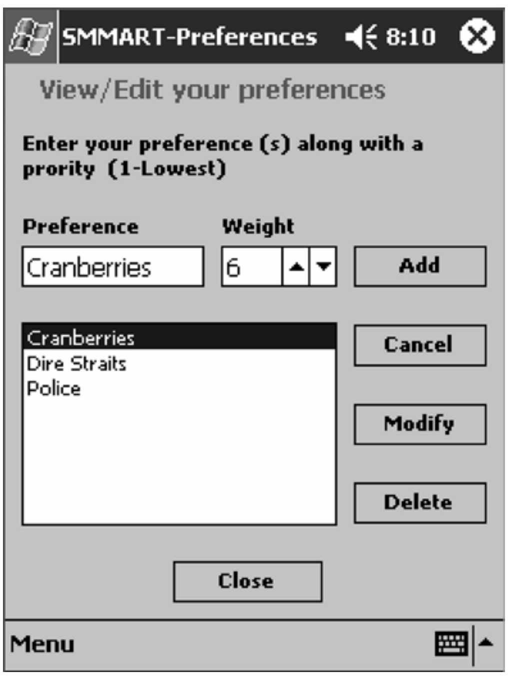

Fig. 7 User preferences

Fig. 8 Current offers

While sorting by weights, preferences with the same weight are also sorted alphabetically.

User preferences are sent to the SMMART Server when the client establishes a connection with a participating store. The *Product Match Agent* in collaboration with the *Product Manager* and the *Category Manager* searches the inventory database for products with offers that match the current user preferences and sends them back to the SMMART Client. The client then displays the offers according to their degree of relevancy to the user's shopping preferences measured by the keyword weights, as shown in Fig. 8. Product details displayed by the SMMART Client, such as the product type, offer end date and the savings potential, are useful indicators for the user to make an informed decision about which products should be explored in detail further.

If the user is interested in exploring a product with an offer, the SMMART Client displays a screen with more useful information about the product, as shown in Fig. 9 This product screen has been designed taking into consideration the limited space available on a PDA screen for including all the above information in a compact manner. Fig. 10 shows the product screen with offer information, while Fig. 11 shows the product screen with related products. The user can also browse through the products related to the current product on display.

In addition to assisting the users in finding products with offers, SMMART also gives the users an option of querying the entire store inventory for products with or without offers using one of the two SMMART Search features as shown in Figs. 12 and 13. In the advanced search screen, the users can search for products whose descriptions contain all keywords, the exact phrase, at least one search keyword, as well as those products that do not contain the given list of keywords. These four options can be used in any combination.

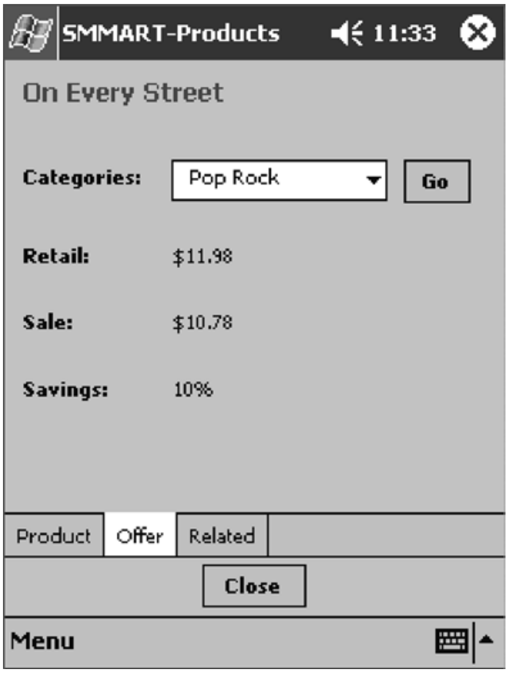

Fig. 10 Offer details

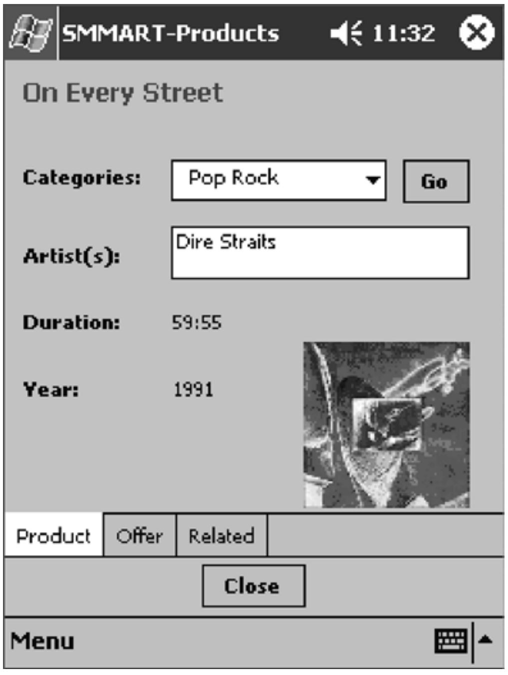

Fig. 9 Basic product details

Fig. 11 Related products

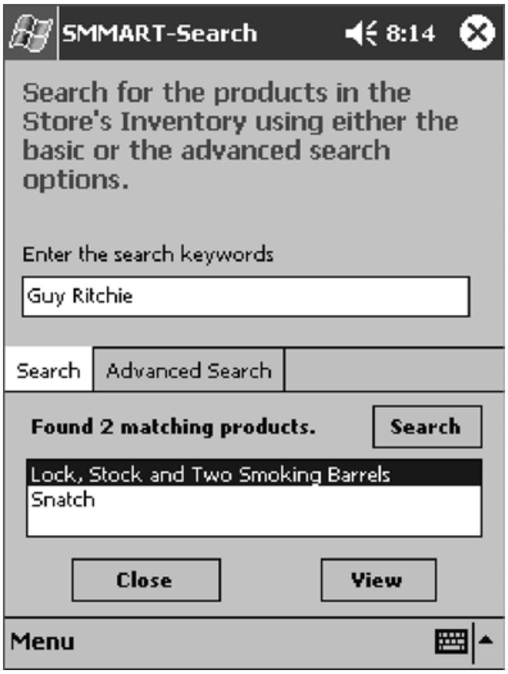

Fig. 12 Basic search feature

Fig. 13 Advanced search

6 Viability and economic feasibility of SMMART

Future success of ubiquitous systems such as SMMART framework depends on two crucial factors. First, the system's client application and its interface must be accepted by the user. If the end-user does not find the system intuitive and convenient to use, it does not matter how beneficial it can be in terms of savings. Secondly, in order for such a system to function, a number of stores must install corresponding servers. The management of each store must be able to see some tangible economic benefits that would outweigh the initial investment in purchasing and installing the required hardware and software and the subsequent upkeep and maintenance.

6.1 Usability of SMMART

We conducted a study to gain more insight into the end-user acceptance of SMMART framework. We created a "test store" whose inventory contained close to a hundred product descriptions, mainly popular musical CDs and movies on DVDs. Thirty-four people participated in the survey, their age ranging from 17 to 43 (with the median age of 21). All members of the group had a definite interest in movies and CDs and visited retailers selling these products at least once a month (some as frequently as every other day). We demonstrated the functionality of SMMART to the survey participants and explained its key features. Then we provided each person with a Dell Axim PDA running a SMMART Client and asked him or her to enter some keywords describing their individual shopping preferences. After that, participants of the survey connected to our "test store" and were able to see whether SMMART could find any products matching their interests. Participants of our survey also experimented with the search feature of SMMART, using both basic and advanced modes. Each time a product was matched to the user preferences or found as a search result, survey participants were able to navigate through the search results and related products. We asked the participants of our survey to "revisit" our "test store" several times and navigate through products to see how the reflection of their shopping preferences might have changed in the SMMART Client. Finally, at the end of the experiment, we asked them to examine the weighed keywords representing their evolved shopping preferences. Upon the completion of the experiment, each survey participant filled out an anonymous questionnaire, cumulative results of which are presented in Table 1.

As can be seen in Table 1, the vast majority of people participating in our survey found SMMART helpful and easy to use. More specifically, based on their own experience with SMMART, 80% of the members of our survey team agreed or strongly agreed that the system is capable of making good matches between their shopping preferences and products in the test store's inventory (question 6 in Table 1). Given a chance to browse through the products found as a result of matching of preferences or searching for keywords, navigate through the different screens of the user interface and system options, 80% of the respondents agreed or strongly agreed that the interface of SMMART Client is intuitive

Table 1 Questions and answers of the end-user survey

	Question		1	2	3	4	5	
1	Using keywords to express my shopping preferences is...	Easy	16	15	2	1		Difficult
2	Entering initial and editing existing preferences is...	Complex		2	1	21	10	Straightforward
3	SMMART can learn and automatically update my shopping preferences	Definitely	12	16	4	2		Not at all
4	Irrelevant product keywords...	Are detected and removed	11	15	7	1		Plague my product matches
5	Products with offers found by SMMART are...	Irrelevant			5	18	11	Good matches
6	My preferences play a role in prioritizing the list of products returned by SMMART...	A major role	12	18	4			No influence
7	Searching for a product using SMMART is...	Cumbersome			2	18	14	Clear
8	Advanced search feature of SMMART helps in narrowing the search results	Greatly	12	16	6			It doesn't help
9	The interface of SMMART Client is...	Intuitive	12	16	5	1		Cluttered
10	I feel that in SMMART my personal information is...	Exposed	1	2	11	12	8	Protected
11	If owned a mobile device with SMMART Client, I would use it in my everyday shopping...	Never		2	1	20	11	Always
12	If owned a mobile device with SMMART Client, I would be inclined to purchase products suggested by SMMART if the price is right	Every time	11	20	3			Never

and easy to use (question 9 in Table 1). Finally, 14 out of 15 or 93% of the respondents said that if they owned a mobile device running a SMMART Client, they would be willing to use the system in their everyday shopping (question 11 in Table 1). As the results of our survey show, it is clear that its participants have a very favorable opinion about SMMART and they find it easy to use.

Generally, consumers are always very concerned about their privacy and they would prefer not to give away any personally identifiable information about themselves or their shopping habits or preferences. From the user's privacy point of view, using SMMART framework is equivalent to the process of searching the inventory of an online store through its Web interface. During the search process, the online store can collect information that would identify the shopper's interests in buying certain products. This information is usually derived from the keywords that the shoppers use when they search for products. Usually, it is also very easy for online stores to detect whether a given search resulted in a successful purchase. SMMART allows its users to make similar product searches fully anonymous because it does not permit the stores to make a logical connection between a product search and a resulting purchase. Furthermore, it would be very difficult for retailers to track return customers by comparing their sets of keywords because when shoppers continually use SMMART, their keyword sets evolve over time.

6.2 Economic aspects

Deploying SMMART framework at a single retail store or at a chain of affiliated stores must be economically justified. The costs of the framework, its supporting infrastructure, data upkeep and maintenance must be less than the revenue from additional sales generated by the customers using SMMART Clients on their mobile devices. To illustrate the economic feasibility of SMMART framework, we developed a simulation model described below.

We consider a scenario, in which a retail store equipped with SMMART Server is visited by n customers over a certain period of time. Several independent variables directly influence the experimental outcome. The probability that customer c_j will make a *purchase* is determined by the value of $P(c_j)$. If customer c_j decides to make a purchase, its *amount* is determined by $A_1(P(c_j))$, which is distributed normally. The probability that the customer c_j is using a mobile device equipped with *SMMART* Client is determined by $S(c_j)$. The probability that this customer c_j will receive a promotion from SMMART and make a purchase *influenced* by it is determined by $I(c_j)$. This measure is also

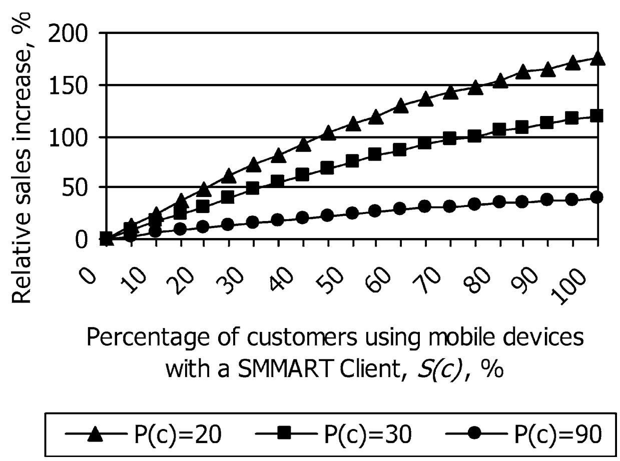

Fig. 14 Expected economic impact of using SMMART framework

dependent on the number of keywords in the user's preferences, the number of items in the store's inventory and the number of received offers. The *amount* of the respective purchase is determined by $A_2(S(c_j)I(c_j))$, which is also distributed normally. We measure $F(P,S,I)$, the impact of SMMART on sales at the given retail site, as the ratio of the total sales to the sales generated only by customers who do not use SMMART Client and thus are uninfluenced by promotions received through them. Therefore, $F(P,S,I)$ is determined as follows:

$$F(P,S,I) = \frac{\sum_{j=1}^{n} \{A_1(P(c_j)) + A_2(S(c_j)I(c_j))\}}{\sum_{j=1}^{n} A_1(P(c_j))}.$$

We conducted a number of experiments with the described simulation model with different values of independent variables. The graphs shown in Fig. 14 portray the results of our experiments for the value of $I(c) = 50\%$, the values of $P(c)$ varying from 20 to 90% and the number of customers $n = 100{,}000$. Because we calculate a ratio of the total sales, higher values of n help improve the quality of obtained data. As demonstrated in Fig. 14, the highest expected economic impact of SMMART can be observed when $P(c)$ is low. This is typical for upscale stores in shopping malls, stores that sell large ticket items or stores where people come to socialize, as well as to shop. Additionally, in such stores, shoppers may have more freedom to use this application—typically, they do not push a shopping cart in front of them and therefore it may be more convenient for them to interact with their PDAs. Furthermore, in such stores buyers usually do not have a shopping list and are inclined to spend more time simply looking around, which may make them more attentive to the product advice offered by SMMART. For example, according to our data, when the probability of a customer to make a purchase is 20% ($P(c) = 20\%$) and when only 5% of all customers are carrying mobile devices with SMMART Client ($S(c) = 5\%$), using SMMART could yield an almost 13% increase in sales.

Alternatively, with higher values of $P(c)$, which are typical for stores where customers are determined to make a purchase and stores where customers make routine purchases guided by a shopping list prepared in advance, such as grocery stores and supermarkets, the expected impact of SMMART is more modest. With $P(c) = 90\%$ and $S(c) = 5\%$, SMMART yields slightly less than a 3% increase in sales.

7 Conclusion and future work

In this paper, we presented a ubiquitous system for interactive mobile marketing, the SMMART framework. SMMART helps a shopper in finding correct matches between his or her buying preferences and products at a retail store that are currently on sale. From the shopper's perspective, SMMART helps the buyer to identify and find interesting products without having to browse through the many pages of promotional brochures often available at retail stores. From the seller's perspective, SMMART creates more buying opportunities, thus potentially increasing the profits.

We built a fully functional prototype of SMMART framework implementing our approach to interactive mobile marketing. We conducted an end-user survey, which demonstrated that a shopper with a wireless PDA equipped with SMMART can receive promotions from retail stores on those products that match his or her interests. The results of this survey also conclude that the end-users find the interface and features of SMMART intuitive and easy to use and that they are ready to embrace SMMART in their everyday shopping. We also studied the economic feasibility of SMMART, which indicates that it will be more effective in stores where customers need additional incentives to make purchases. Possible examples include stores in shopping malls, bookstores, consumer electronics warehouses and any other retailers where consumers come not only to shop, but also to socialize.

SMMART framework can be extended by providing more features that would enhance its usability. Product information pages of the SMMART Browser could be enhanced with a map schematically showing the location of the selected product in the store. SMMART framework could also provide the user with the ability to reserve an item, which is on back order at the current sale price, or the option of ordering an in-stock item to be picked up and purchased on a designated date and time. This feature could also work well with large items such as big screen TVs or other products where the inventory is not kept on the display floor.

Acknowledgements This project was supported in part by the National Science Foundation award CNS-0354144. The authors would like to thank John Hudson for his help in implementing a portion of the SMMART Client. We are grateful to Steve Jones for his encouragement to submit this paper. We are also grateful to the anonymous reviewers for their helpful comments. Finally, we would also like to thank a group of students from Columbus State University who participated in our end-user survey.

References

1. Kannan P, Chang A, Whinston A (2001) Wireless commerce: marketing issues and possibilities. In: Proceedings of the 34th Hawaii international conference on system sciences, Maui, Hawaii, 3–6 January 2001
2. Maamar Z (2003) Commerce, e-commerce, and m-commerce: what comes next? Comm ACM 46(12):251–257
3. Tamminen S, Oulasvitra A, Toiskallio K, Kankainen A (2004) Understanding mobile contexts. Pers Ubiquit Comput 8(2):135–143
4. Kurkovsky S, Harihar K (2004) Intelligent pervasive framework for consumer-supplier interaction. In: Proceedings of the 2004 international conference on pervasive computing and communications (PCC-04), Las Vegas, NV, 21–24 June 2004

5. Varshney U, Vetter R (2002) Mobile commerce: framework, applications and networking support. Mobile Network Appl 7(3):185–198
6. Yunos H, Gao J, Shim S (2003) Wireless advertising's challenges and opportunities. IEEE Comput 36(5):30–37
7. Varshney U (2003) Location management for mobile commerce: applications in wireless internet environment. ACM Trans Internet Technol 3(3):236–255
8. Ratsimor O, Finin T, Joshi A, Yesha Y (2003) eNcentive: a framework for intelligent marketing in mobile peer-to-peer environments. In: Proceedings of the 5th international conference on electronic commerce (ICEC-03), Pittsburg, PA, 30 September–3 October 2003
9. Satyanarayanan M (2002) Challenges in implementing a context-aware system. IEEE Pervasive Comput 1(3):2
10. Hristova N, O'Hare G (2004) Ad-me: wireless advertising adapted to the user location, device and emotions. In: Proceedings of the 37th annual Hawaii international conference on system sciences (HICSS'04), Big Island, Hawaii, 5–8 January 2004
11. Aalto L, Göthlin N, Korhonen J, Ojala T (2004) Bluetooth and WAP Push-based location-aware mobile advertising system. In: Proceedings of the 2nd international conference on Mobile Systems, Applications, and Services, Boston, MA, 6–9 June 2004
12. Jones S, Jones M, Deo S (2004) Using keyphrases as search result surrogates on small screen devices. Pers Ubiquit Comput 8(1):44–68
13. Dey A (2001) Understanding and using context. Pers Ubuquit Comput 5(1):4–7
14. Dix A, Rodden T, Davies N, Trevor J, Friday A, Palfreyman K (2000) Exploiting space and location as a design framework for interactive mobile systems. ACM Trans Comput Human Interact 7(3):185–321
15. Dourish P (2004) What we talk about when we talk about context. Pers Ubiquit Comput 8(1):19–30
16. Newcomb E, Pashley T, Stasko J (2003) Mobile computing in the retail arena. In: Proceedings of the conference on human factors in computing systems, Fort Lauderdale, FL, 5–10 April 2003
17. Lee Y, Benbazat I (2003) Interface design for mobile commerce. Comm ACM 46(12):49–52
18. Brown P, Jones G (2001) Context-aware retrieval: exploring a new environment for information retrieval and information filtering. Pers Ubuquit Comput 5(4):253–263
19. Adomavicius G, Sankaranarayanan R, Sen S, Tuzhilin A (2005) Incorporating contextual information in recommender systems using a multidimensional approach. ACM Trans Inform Syst 23(1):103–145
20. Pilioura T, Tsalgatidou T, Hadjiefthymiades S (2003) Scenarios of using web services in m-commerce. ACM SIGecom Exchange 3(4):28–36
21. Kakihara M, Sørensen C (2002) Mobility: an extended perspective. In: Proceedings of the 35th Hawaii international conference on system sciences (HICSS'02), Big Island, Hawaii, 7–10 January 2002