ISSN: 2319-5967
ISO 9001:2008 Certified
International Journal of Engineering Science and Innovative Technology (IJESIT)
Volume 3, Issue 1, January 2014

Mobile Services on Wi-Fi Proximity Base

Yousef Daradkeh[1], Tariq (Moh'd Taisir) Al Omari[2], Dmitry Namiot[3]
[1]Faculty of Engineering in Wadi Addawasir, Salman Bin Abdul Aziz University, KSA
[2]Faculty of Education Traghen, Department of Computer Science
University of Sebha / Libya
[3]Faculty of Computational Math and Cybernetics, Lomonosov Moscow State University
Moscow, Russia

Abstract— This paper describes a new model for delivering hyper-local data for mobile subscribers. Our model uses Wi-Fi proximity as a service. According to this concept, any existing or even a specially created Wi-Fi hot spot could be used as presence sensor that can trigger access for some user-generated information snippets. In this article we present several examples for deployment: context-aware browser, integration for proximity and social streams and human dynamics models based on proximity data. As the possible use-cases, we can mention, for example, news and deals delivery in malls, news feeds for office centers and campuses, social curation, Smart City projects, personal classifieds, etc.

Key words: **Wi-Fi, proximity, collaborative location, reality mining, social curation, context-aware browsing.**

I. INTRODUCTION

This paper describes how the measurements collected from the Wi-Fi sensors on the mobile phone could be used for new mobile services, based on context-aware computing ideas. It presents an extended and reworked version of conference paper [1]. Technically, proximity sensor is the sensor able to detect the presence of nearby objects without any physical contact [2]. In this article we will describe a model where the existing or even especially created wireless networks could be used as proximity sensors. The detected proximity will trigger either hyper-local news data delivery to mobile subscribers or data selection from the social streams.

Many mobile applications can be characterized as collaborative in the sense that mobile nodes use the wireless network to interact with other mobile nodes that have come together at some common location.

Collaborative nodes typically come together in some area, establish associations with other collaborative nodes dynamically, and make use of common services already available in that locality or provided by members of the group. The members of such a group may migrate together, like visitors in a mall, some group of pedestrians, etc. [3]. Although these applications may use infrastructure networks, they will often use ad hoc networks since they are immediately deployable in arbitrary environments and support communication without the need for a separate infrastructure. This collaborative style of application may be useful in the ubiquitous computing [4]. Context awareness is defined as complementary element to location awareness. Whereas location may serve as a determinant for resident processes, context may be applied more flexibly with mobile computing with any moving entities, especially with bearers of smart communicators. Context awareness originated as a term from ubiquitous computing or as so-called pervasive computing and deals with linking changes in the environment with computer systems, which are otherwise static [5].

Modern applications adopt a context-aware perspective to manage:

a) Communication among users and among systems, or between the system and the user,

b) Situation-awareness, like modeling location and environment aspects (physical situation) or the current user activity (personal situation)

c) Knowledge chunks: determining the set of situation-relevant information, services or behaviors [6].

In our approach, we are dealing with context-aware knowledge chunks. Let us start with the base element – location. There are many different approaches for getting location info for mobile subscribes. In general, it could be pretty standard nowadays (GPS, cell-id, assisted GPS [7]),but everything is getting more complicated as soon as we need indoor positioning. Due to the signal attenuation caused by construction materials, the Global Positioning System (GPS) loses significant accuracy indoors. Instead of satellites, an indoor positioning system (IPS) relies on nearby anchors (nodes with a known position), which either actively locate tags or provide environmental context for devices to sense. The localized nature of an IPS has resulted in design fragmentation, with systems making use of various optical, radio, or even acoustic technologies [8]. However, all of them require

ISSN: 2319-5967
ISO 9001:2008 Certified
International Journal of Engineering Science and Innovative Technology (IJESIT)
Volume 3, Issue 1, January 2014
the utilization of their own API with their own protocols. This can be a big challenge for developing heterogeneous scenarios where different localization systems have to be used for a location service.

One of the most used approaches to indoor location is Wi-Fi based positioning. A standard Wi-Fi based positioning system, such as the one offered by Cisco, is completely software-based and utilizes existing Wi-Fi access points installed in a facility and radio cards already present in the user devices. The IEEE 802.11 Wi-Fi wireless network infrastructure consists of wireless clients and several access points (AP) that act as bridges between the wireless network and the local area network. Client primarily uses some selected access point. This selection could be done manually or automatically for AP that provides clients with the strongest RSSI (received signal strength). As a clients roams, it periodically does a site survey of signal quality measures to determine the best AP to associate. Wi-Fi based location systems generally work in two phases. Phase 1 is so called calibration. On this phase a human operator performs a site survey by measuring the RSSI from different APs at some selected points in the environment. As a result, this phase creates a radio map for the environment. Phase 2 is the real location estimation phase. It matches sampled points on the radio map with the closest RSSI values to the target. This calibration phase could be very costly. For example, the Ekahau location system requires 80 RSSI samples to be taken every 3 meters to attain an average positioning accuracy of 3 meters in a 1000m2 environment [9]. Also we should mention the instability in Wi-Fi based IPS. Some environmental factors that affect RSSI such as people presence and movement, humidity level may change the “static” settings used for radio-map.

Lets us mention also one more interesting approach: collaborative location (CL). The most interesting approach for our future development is Cooperative Location-sensing (CLS). As per this approach, some small number of devices, called reference nodes, obtains their coordinates and the rest, unknown-location nodes, must determine their own coordinates. Simply saying, hosts cooperate and share positioning information. CLS uses a grid representation that allows an easy incorporation of external information to improve the accuracy of the position estimation [10]. The motivation for CL and CLS is very transparent. In many situations, due to environmental, cost, maintenance, and other obstacles, the deployment of a dense infrastructure for location sensing is not feasible. It is exactly what we wrote about infrastructure-less system. In CLS, hosts estimate their distance from their neighboring peers. This can take place with any distance estimation method available (e.g., using signal strength). They can refine their estimations iteratively as they incorporate new positioning information. The accuracy of cooperative localization increases with the density of devices. Another interesting aspect for our approach is dynamic location based services. For example, AROUND [11] architecture is proposed as an approach for supporting location-based services in the Internet environment. AROUND provides a service location infrastructure that allows applications to select services that are specifically associated with their current location. The architecture includes a flexible scope model that defines the association between services and location, and a service location infrastructure organized by spatial criteria and optimized for location-based queries. It deals with two proximity models presented on the Figure 1.

We can select things (events, data) either based on the distance or in scope and at this point we are ready to make the last proposition before switching to our SpotEx model. Of course, the acronym LBS (Location Based Systems) contains the word “location”. But do we really need the location for the most of the services? As seems to us the final goal (at least for the majority of services) is to get data related to the location, rather than location itself. Location in the classical form (latitude, longitude) here is just an intermediate result we can use as key in our requests for obtaining data (our final goal). So, why do not request data directly, if we can estimate location? And the word “estimate” here means actually the proximity. Two objects are either in the same scope or close each other. NFC has a very short range (measured in inches), Bluetooth typically has a range of 20-30 feet, and Wi-Fi is often 50-100 feet in range. Anything that is “visible” using these wireless technologies can be declared to be “in proximity”. But here are the obvious obstacles. The location services have got a wide support on mobile world for example. We should make proximity as easy to use for developers as location has now become. Then we can expect a similar explosion of innovative services. In our development we investigate the proximity statements, associated with Wi-Fi nodes. The explanation for this choice is actually very transparent. There is only one magic word behind this selection. It is a penetration. Mobile phones are probably the most obvious candidate for mass sensing products. As it is stated in [12], the modern phone is a useful tool in social sciences, particularly sociology, social psychology, urban studies, technology assessment, and media studies. The device is willingly carried by a large fraction of people, integrates a number of technologies for automatic observation, can be programmed to interact with the user, and can communicate with remote researchers.

ISSN: 2319-5967
ISO 9001:2008 Certified
International Journal of Engineering Science and Innovative Technology (IJESIT)
Volume 3, Issue 1, January 2014

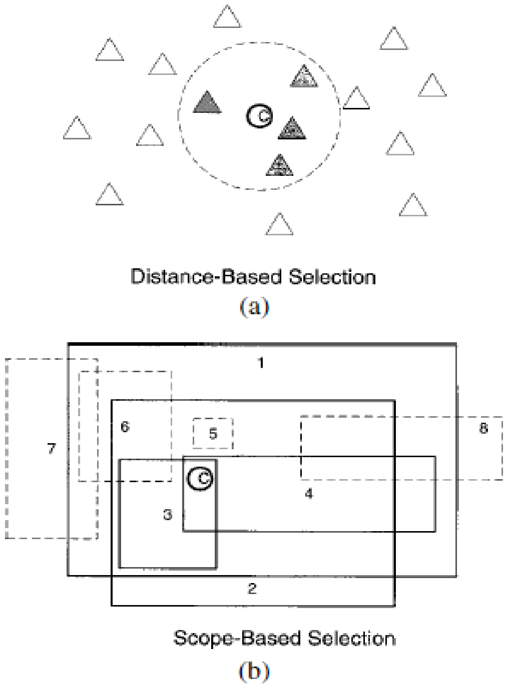

Fig 1. Proximity models

Technically, the modern smart phones could support different build-in sensors. In our research, we choose one particular kind of sensors: wireless network sensors. Practically, we are talking about Wi-Fi modules. The reasons are absolutely the same as the above – this choice let us deal with most widely adopted sensors. All the smart phones nowadays support wireless networks. The rest of the paper is organized as follows. Section II contains an analysis of existing projects that cover proximity services. In Section III, we consider our SpotEx approach as context-aware browser. In Section IV we describe models for integration SpotEx and social streams.

II. RELATED WORK

Let us describe some existing projects. AllJoyn [13] is a peer-to-peer technology that enables ad hoc, proximity-based, device-to-device communication without the use of an intermediary server. Technically, it is a set of APIs for developers. Nokia Instant Community is a new, instant way for communities to socially interact when in close proximity, without the need for WLAN infrastructure or Bluetooth and cellular connections [14]. The clever thing about Nokia Instant Community is that it doesn't need the Internet. It means no searching for a Wi-Fi hotspot to get you in touch, and you do not need infrared or Bluetooth either. It works completely by using the device's ad-hoc Wi-Fi. But we should mention here that the whole idea looks very similar to Wi-Fi direct spec (Figure 2)

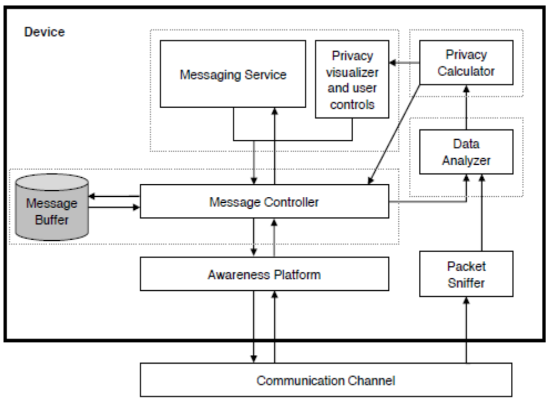

Fig 2. Privacy in Nokia Instant Community

ISSN: 2319-5967
ISO 9001:2008 Certified
International Journal of Engineering Science and Innovative Technology (IJESIT)
Volume 3, Issue 1, January 2014

Local Social by Rococo soft [16] offers a proximity platform for sharing social data. Local Social combines information about people and things close to a user, with information from one or more of their social networks, and makes that information available to a mobile application in a simple way (Figure 3).

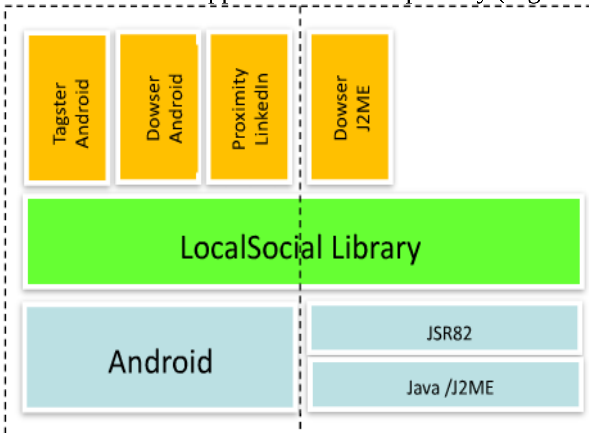

Fig 3. Local Social Platform

Local Social has 2 key components:

1. The Local Social Mobile Library, which is deployed on the handset. This gives developers APIs that allow them to:

a) Use proximity to detect devices nearby using Bluetooth, WiFi, and NFC
b) Connect securely to the Local Social Service
c) Discover information about the devices nearby - including opt-in social tags and information
e) Send messages to devices nearby, store tags about those devices for later

2. The Local Social Cloud Service is a web-hosted service, which mediates between the mobile application and the cloud, and stores and retrieves information about the devices (as tags), tracks analytics, and provides social connections to key social networks. What are the common points to all projects? It is device based API. At the first hand, they are oriented to the device-to-device proximity. It is probably more convenient to the social networks. In our project, we will target device-to-infrastructure objects proximity. It is more convenient to the information providing. For example, deliver some news data for people in Smart City applications; provide dynamic information systems in campuses and office buildings, etc.

III. SPOTEX AS CONTEXT-AWARE BROWSER

Originally, this development started as a special extension for indoor navigation. What if we stop our traditional indoor positioning schema on the first stage: detection of Wi-Fi networks? Obviously, that this detection actually already provides some information about the location – just due to local nature of Wi-Fi network. And as the second step we add the ability to describe some rules (if-then operators, or productions) linked to the Wi-Fi access points. Our rules will simply use the fact that the particular Wi-Fi network (access point) is detected. Based on this conclusion we will open (read – make them visible) some user-defined messages to mobile terminals. Actually, it is a typical example for the context aware computing. The visibility for user-defined text (content) depends on the network context. For the first time this service SpotEx (Spot Expert [17]) was described by the authors in article that has been published in NGMAST-2011 proceedings [18]. It is a working application for Android platform.

Our SpotEx model is based on the ideas of Wi-Fi proximity. Any Wi-Fi hot spot works here just as presence sensor. But we are not going to connect mobile users to the detected networks and our suggestion does not touch security issues. It is not about connectivity at all. We need only SSID for networks and any other public information. So, our service contains the following components:

- Database (store) with productions (rules) associated with Wi-Fi networks. It is web-based data store
- Rule editor. Web application (including mobile web) that lets users add (edit) rule-set, associated with some Wi-Fi network

- Mobile applications, that can detect Wi-Fi networks, check the current conditions against the database and execute productions

With SpotEx we can take any existing Wi-Fi network (or networks especially created for this service – the most interesting case, see below) and add some rules (messages) to that network. Message here is just some text that should be delivered to the end-user's mobile terminal as soon as the above-mentioned network is getting detected via our mobile application. The word "delivered" here is a synonym for "available for reading/downloading". The possible use cases, including commercial deployment are obvious. Some shop can deliver deals/discount/coupons right to mobile terminals as soon as the user is near some predefined point of sale. We can describe this feature as "automatic check-in" for example. Rather than directly (manually or via some API) set own presence at some place (e.g., similar to Foursquare, Face book Places, etc.) and get deals info, with SpotEx mobile subscriber can pickup deals automatically. Campus admin can deliver news and special announces, hyper local news in Smart City projects could be tight (linked) to the public available networks and delivered information via that channel, etc.

Especially, we would like to point attention to the most interesting (by our opinion, of course) use case: Wi-Fi hot spot being opened right on the mobile phone. Most of the modern smart phones let you open Wi-Fi hot spots. We can associate our rules to such hot spot (hot spots) and so our messages (data snippets) become linked to the phones. Practically, we are getting dynamic LBS here: phone itself could be moved and so, the available data will be de-facto moved too. It opens the door for a new class of applications – personal mobile information services. If our rules describe some content and this content could be actually linked to the particular phone, then phone owner can manage content's availability simply by switching on/off his personal hot spot. This use case is probably the most transparent demonstration of SpotEx model. We can open "base" network right on the mobile phone, attach ("stick") rules for the content to that network and our new information channel will be online. There is no infrastructure except the smart phone itself and there is no calibration phase as per indoor positioning systems. By the way, it is the main difference from the centralized location frameworks from Goggle, Nokia or Ericsson, too. All the frameworks for Wi-Fi positioning are using static databases of nodes and note again, that this approach does not touch security and connectivity issues. You do not need to connect mobile subscribers to your hot spot. SpotEx is all about using hot spot attributes for triggers that can discover the content. The term Wi-Fi proximity is used sometimes in connection with Wi-Fi marketing and mean on practice just setting a special splash screen for hot spot that can show some advertising/branded messages for users during the connection to that hot-spot. Unlike this, SpotEx threats Wi-Fi hot spots just as sensors.

How our productions data store (base of rules) looks like? Each rule looks like a production (if-then operator). The conditional part depends on the following objects:

Wi-Fi network SSID and MAC-address,
Signal strength (optionally),
time of the day (optionally),
client ID (MAC-address, see below)

and includes logical functions (predicates) that depend on the above mentioned data and logical operators. The predicates (in the current version) are:

IS_VISIBLE ( )
NOT_VISIBLE ( )
CLOSE_THAN( )
FIRST_VISIT( )
FOLLOW_UP_VISIT( )
TIME()
TIME_WITHIN()

Functions IS_VISIBLE () or NOT_VISIBLE () accept as a parameter network ID (e.g., SSID or MAC-address for access point) and returns a Boolean value which depends on the current networks visibility.

Function CLOSE_THAN() accepts two parameters identified wireless networks (Wi-Fi access points) and returns true if mobile terminal is close to the Wi-Fi access point described in the first parameter. Two functions FIRST_VISIT() and FOLLOW_UP_VISIT( ) are based on the simple fact that in Wi-Fi based system we have MAC-address for mobile terminal. The whole system does not require authorization. So, unlike traditional systems like foursquare, etc., it is possible to discover data anonymously. But in the same time, we have some analogue of UUID, allowing us distinguish the users. It is MAC-address. We keep historical logs for vectors (MAC-address, wireless environment info) and use it for detecting new or retuned "visitors". For example, if for the same MAC-address we have at least two historical records where at least one Wi-Fi access point mentioned twice or more, it is follow-up visitor. The last two functions operate with time. Function TIME() returns current time, and TIME_WITHIN() lets prove that current time is within some predefined interval.

The list of base functions is not fixed finally, of course. It corresponds to the new developments in use cases. Finally, our collection of rules is a set of operators like:

IF IS_VISIBLE('mycafe') AND TIME_WITHIN(2pm, 3pm) THEN {present the special deal info}.

Code block (data snippet) {present the special deal info} contains HTML code delivered to mobile browser for mobile subscribers who can see access point mycafe within 1pm-2pm time frame. Each such snippet has got a title (text) and some HTML content (it could be simply a link to external site for example). Snippets present coupons/discounts info for malls, news data for campuses, etc. Technically, any snippet could be presented as a link to some external web site/mobile portal or as a mobile web page created automatically by the rule editor from SpotEx. For data presented as links to external existing mobile sites (portals), SpotEx works as some universal discovery tool. De facto, it lets mobile subscribers to be aware about context-relevant web resources. Owners for the web resources can describe own sites via network nodes related rules rather then present for them individual QR-codes or NFC-tags, for example.

In case of describing some content right in the SpotEx, the whole system works in this part as a content management system (CMS). For each existing data snippet SpotEx rule editor creates automatically mobile web page. SpotEx middleware hosts that page on the own server. It means, that for describing our data we can use all resources that could be presented on HTML pages. For example, multimedia content is also supported. SpotEx mobile application creates on the fly dynamic HTML page from titles (according to rules that are relevant in the given context). It is what mobile users will see. Technically, it is a classical rule based expert system. It matches existing rules against the existing context and makes the conclusions. Existing content here presents a snapshot for Wi-Fi environment. It is a list of hot spots with attributes and conclusion clause here is a mobile web page assembled from the individual titles. Figure 4 illustrates our rule editor. Because our rules form the standard production rule based system, we can use old and well know algorithms like Rete [19] for the processing. A Rete-based expert system builds a network of nodes, where each node (except the root) corresponds to a pattern occurring in the left-hand-side (the condition part) of a rule. The path from the root node to a leaf node defines a complete rule left-hand-side.

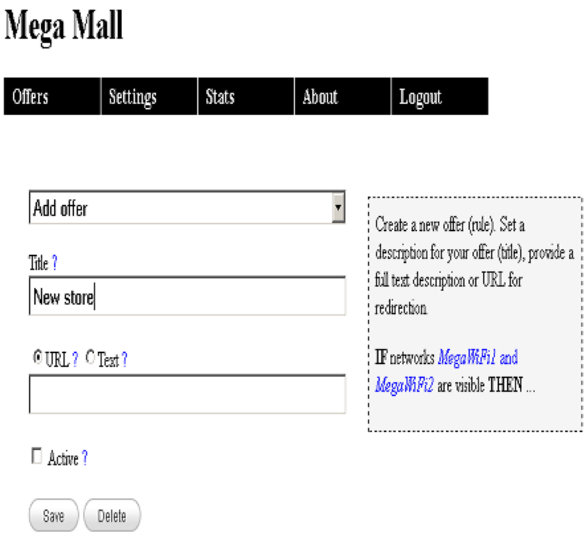

Fig 4. Create a new rule

ISSN: 2319-5967
ISO 9001:2008 Certified
International Journal of Engineering Science and Innovative Technology (IJESIT)
Volume 3, Issue 1, January 2014

Each node has a memory of facts, which satisfy that pattern. This structure presents essentially a generalized tree. As new facts are asserted or modified, they propagate along the network, causing nodes to be annotated when that fact matches that pattern. When a fact or combination of facts causes all of the patterns for a given rule to be satisfied, a leaf node is reached, and the corresponding rule is triggered [20]. It is illustrated on Figure 5.

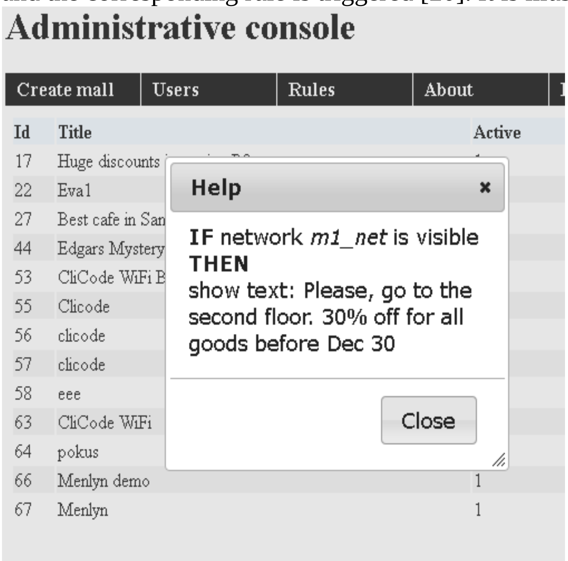

Fig 5. SpotEx rules

The current implementation for mobile client is based on Android OS. This application uses WiFi Manager from Android SDK - the primary API for managing all aspects of Wi-Fi connectivity. This API let us pickup the following information about nearby networks

SSID - the network name.

BSSID - the address of the access point.

Capabilities - describes the authentication, key management, and encryption schemes supported by the access point.

Frequency - the frequency in MHz of the channel over which the client is communicating with the access point.

Level - the detected signal level in dBm.

So, actually all the above-mentioned elements could be used in our productions.

Technically, any snippet could be presented as a link to some external web site/mobile portal or as a mobile webpage created automatically by the rule editor included into SpotEx. Rule editor works in both desktop and mobile web. So, once again just having ordinary smart phone is enough for creating (opening) information channel for delivering hyper-local news data. SpotEx rule editor creates mobile web page for the each provided data snippet. It hosts that page on the own server. It means, by the way, that for presenting our data we can use any resources that could be presented on HTML pages. For example, any multimedia content is also supported.

SpotEx mobile application, being executed, creates dynamic HTML page from titles (according to rules that are relevant in the given context) and presents that mobile web page to the user. It works just as a classical rule based expert system. It matches existing rules against the existing context and makes the conclusions. Existing context here is a description for “Wi-Fi environment”. In other words, it is a list of hot spots with their attributes. And conclusion here is a list of titles that can be presented as a dynamically created mobile web page. On that page, each discovered title could be presented as a hyperlink points to the appropriate data snippet. Any click on the interested title opens the snippet (shows or discovers data to mobile user). So, for the mobile users the whole process looks like browsing, where their browser becomes aware about hyper-local content. It is a typical example of context-aware retrieval. The development of personal networked mobile computing devices and environmental sensors means that personal and context information is potentially available for the retrieval process. Classically, it is referred as context-aware retrieval [21]. The objective of incorporating contextual information into the retrieval process is to attempt to deliver information most relevant to the user within their current context. In general, this process may involve personalization of the retrieval process in combination with context-awareness, but this need not be the case.

As it is highlighted in [21], the ideal retrieval agent in a ubiquitous computing environment could (a) perform actions automatically if there is no need to consult the user; (b) summarize or coalesce documents before presentation; (c) decide when to deliver the retrieved information to the user, e.g. if it is marked as highly relevant, it should be delivered immediately, interrupting whatever they are currently doing; and (d) learn from users how its performance can be improved. The introduction of information delivery not directly controlled by the user introduces the ideas of proactive information retrieval, where a device may automatically poll sensors (in our case) and may trigger information when the user enters a certain context (some like geo-fence described via network proximity).

As per other functionality of our context-aware browser, we can highlight the following notes. At the first hand, we can note that it is the "pull model", versus the "push model" that has been proposed by Bluetooth marketing for example. It could be more convenient (more safe) for the users – there are no automatically downloaded files/messages, etc. But in the same time nothing prevents us from updating that dynamic web page automatically (e.g., by the timer) and simulating "pull model" in the user-safety mode. At the second hand, we can note that because it is browsing, the whole process is anonymous. Indeed, there is no sign-in in the SpotEx. Of course, any data snippet may lead to some business web site/portal, where that site may ask about login, etc. In the same time, SpotEx itself is anonymous. Unlike social networks like Foursquare, there is no need to disclose own identity just for looking new deals in some mall you are currently in. In the same time, we can still collect some meaningful statistics in SpotEx. Because the model requires Wi-Fi to be switched on, we have automatically unique ID for each client. It is MAC-address. It plays a role of global UUID here. So, where we have not login info for our clients, we still can distinguish them. It let us detect, for example, the same person, who did that already twice during the last week, is opening the particular data snippet right now.

Because mobile users in SpotEx model actually work with web pages, we can use pretty standard methods for web server log analysis in order to discover user's activities. A statistical analysis of the server log may be used to examine traffic patterns by time of day, day of week, etc. So, we can detect frequent visitors, usage patterns, etc. And even more – we can use that information in our rules. For example, some mall may offer special things for frequent visitors, etc. Data from real time analytics for our info snippets could be used in conditional parts of our rules too. The next stage of development targets the simplicity of preparing data for SpotEx model. What if instead of the separate database with rules (as it is described above) we add the ability to provide a special markup for existing HTML files? So, rather than writing separate if-then rules we can describe our rules right in HTML code. Technically, we can add for example HTML div blocks with attributes that describe our rules (their conditions). As a model for applied development, we can suggest Web Intents [22].

IV. SpotEx and social networks

Here, we will describe how the above mentioned model for the network proximity could be adopted to the social streams. For pairing sensors data and social streams (e.g. Twitter stream – what is our case), we can describe thematic stream as data (topics) discovered from the tweets and data recorded from our sensors (wireless networks sensors).

Technically, there are three basic parameters (so called where-when-what scheme): location, time and topic. And our stream is a collection of different spatio-temporal-thematic points. Each point has got the following attributes: location, time stamp and topic. Topic here is some content, extracted from the tweets (e.g., hash tags).

Point = { location, time stamp, topic }
Stream = [$Point_1$, $Point_2$ …]

Usually, location is simply a pair of latitude, longitude. But deploying wireless networks sensors let us replace location with proximity. We mean here the relative position regarding the network nodes. E.g., for Wi-Fi networks it could be a proximity to some Wi-Fi access point.

ISSN: 2319-5967
ISO 9001:2008 Certified
International Journal of Engineering Science and Innovative Technology (IJESIT)
Volume 3, Issue 1, January 2014

In other words, this approach let us combine topics (read – text extraction from the posting in the social networks) around the wireless nodes. It could provide more precise location info than the traditional geo places based positioning in social networks. Let us highlight the key points for this approach. At the first hand, it is a special form of check-ins. Check-ins (as objects) could be customized [23]. In general, our approach looks like a special check-in service for social networks. What is a typical check-in record in social networks? It is some message (post, status) linked to the selected location (place). What are the reasons for members in social networks use such special kind of messages? Sometimes it could be stimulated by the business. Practically, user posts advertising for the business in exchange for some benefits. Sometimes it could be used for social connection: let other know where I am and see where my friends are.

But the key point is the special kind of records in the social network – check-in. It could be customized, of course, and business can create own forms for check-in records [24], but they are still some posts in the social networks. In other words, they are always part of the social stream. What if we create a new type of check-in records and separate them from rest of stream? It means that we will provide a separate database that just contains a list of accounts from the social network being concentrated (at this moment!) nearby some place. It is a temporal database, check-in records could be changed constantly and it does not contain the social stream itself – just IDs (e.g. nick names) for accounts.

The second key moment is a new definition for places. Traditionally, for social networks it is a pair from geo coordinates (latitude, longitude) and some description. It already creates problems for indoor (many places within the big mall will be on the same geo position, etc.). What if we remove our "new" places from the social networks too? We will describe our places separately and define them via proximity (network proximity in our case) attributes rather than via geo coordinates. Proximity based definition means that each place should be defined via some metric that we introduce for our network. Let us talk about Wi-Fi. As a base for metric we can use:

- Visible SSID for networks
- counter for access points (our mobile phone can see from the particular place more than one access point for the same network)
- RSSI (signal level) for the each device

As soon as two users (mobile phones) have at least one common visible Wi-Fi access point – they are in the proximity. And obviously, the more similar (close each other) the visible network environments are, the bigger is level of proximity. Now, suppose that we have a mobile application that lets users confirm their social network identity and link that identity with wireless networks info. This application will form (fill) our external temporal database that contains social network identity confirmation and an appropriate network info. Running this application is simply a new form of check-in. This new check-in is an "external" entity for the social network. Our application does not post data back to the social network. It keeps data outside (in the own database). So, in the terms of privacy this check-in does not affect (does not touch, actually) account settings in the social network.

The reasons for acting with new check-ins are absolutely the same as for their old counterparts. Sometimes it could be stimulated by the business. Business entity can use that information for statistics and deliver some benefits in exchange for "check-in". And it could be used of course for the connection: let other know where I am and see who else is here also and now what can we do with this external check-ins database? At the first hand we can list other people at any particular location. Actually, it is always a list of people at "this" location only. It is just because our proximity based system does not support lists of location in the traditional form. Each our "location" is described via Wi-Fi sensors, via visible access points and RSSI. Obviously, all the attributes are dynamical. So, after own check-in, user can see the nearby check-ins only. In this approach user is simply unaware about other "places". He needs to move and check-in again in order to see a new place as well as its check-ins. Note, that the places themselves in this approach are also dynamical.

At the second, we can show (search) social streams nearby. Via public API, we can read data feeds for users. Of course, we can do that if an appropriate data feed is not protected. This feature let us keep the full respect to the existing privacy settings in social networks. Currently we have a proof of concept application that works on Android platform for Twitter. Figure 6 illustrates the main screen from In-Proximity check-in application (currently in beta for Android platform). Application detects the visible Wi-Fi networks (by the same manner as the above described SpotEx does) and asks user for how long his record should be saved in the temp database.

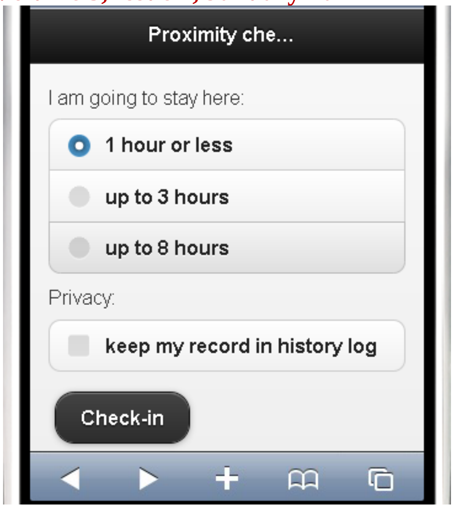

Fig 6. In-proximity check-in

The record could be removed early in case of check-in in any other place. It is up to user also to allow (disallow) saving this check-in record in history log (by default nothing is saved).

V. CONCLUSION

This paper described a new context-aware browsing model for mobile users based on the ideas of Wi-Fi proximity. Developed service can use any existing as well as the especially created (described) Wi-Fi network as presence trigger for discovering user-defined content right to mobile subscribers. Proposed approach is completely software based and does not require additional hardware investments. For using SpotEx you need nothing except the smart phone. Also this approach supports ad-hoc solutions and unlike existing Wi-Fi based indoor-location techniques does not require the upfront space preparations. This service could be used for delivering commercial information (deals, discounts, coupons) in malls, hyper-local news data, data discovery in Smart City projects, personal news sites, etc.

ACKNOWLEDGMENT

The authors would like to thank the anonymous referees. Authors would like to thank the management of Salman Bin Abdul Aziz University for providing financial assistance to do this research a successful manner.

REFERENCES

[1] D. Namiot and M. Sneps-Sneppe “Wi-Fi proximity as a Service”, Think Mind SMART 2012, The First International Conference on Smart Systems, Devices and Technologies, pp.62-68.

[2] C. Seeger, A. Buchmann, and K. Van Laerhoven, “Wireless sensor networks in the wild: three practical issues after a middleware deployment”, Proceedings of the 6th International Workshop on Middleware Tools, Services and Run-time Support for Networked Embedded Systems ACM New York, NY, USA, 2011.

[3] R. Meier, V. Cahill, A. Nedos, and S. Clarke, “Proximity-Based Service Discovery in Mobile Ad Hoc Networks”, Distributed Applications and Interoperable Systems Lecture Notes in Computer Science, 2005, Volume 3543/2005, pp. 1147-1149, DOI: 10.1007/11498094_11.

[4] M. Weiser: "Ubiquitous Computing," IEEE Hot Topics, vol. 26, pp. 71-72, 1993.

[5] S. Poslad “Ubiquitous Computing”, Smart Devices, Environments and Interactions DOI: 10.1002/9780470779446.fmatter, 2009, John Wiley & Sons, Ltd.

[6] C. Bolchini1, G. Orsi, E. Quintarelli, F. A. Schreiber, and Letizia Tanca, “Context modeling and context awareness: steps forward in the context-addict project”. IEEE Data Eng. Bull., 34(2): pp. 47–54.

[7] R. Ahas, S. Silm, O. Järv, E. Saluveer, and M. Tiru, “Using Mobile Positioning Data to Model Locations Meaningful toUsers of Mobile Phones”, Journal of Urban Technology Volume 17, Issue 1, 2010 pp. 3-27.

ISSN: 2319-5967
ISO 9001:2008 Certified
International Journal of Engineering Science and Innovative Technology (IJESIT)
Volume 3, Issue 1, January 2014
[8] Comparison of Wireless Indoor Positioning Technologies http://www.productivet.com/docs-2/Wireless_Comparison.pdf Retrieved: Jul, 2012.

[9] Y.Chen, J.Chiang, H.Chu, P. Huang, and A.Tsui, "Sensor-assisted Wi-Fi indoor location system for adapting to environmental dynamics", MSWiM '05 Proceedings of the 8th ACM international symposium on Modeling, analysis and simulation of wireless and mobile systems, pp. 118 – 125.

[10] J.Ash, S.Kyperountas, A.Hero, R.Moses, and N.Correal, "Locating the nodes: cooperative localization in wireless sensor networks", Signal Processing Magazine, IEEE, vol.22, issue 4, 2005, pp. 54 – 69.

[11] R. José, A. Moreira, H. Rodrigues, and N. Davies, "The AROUND Architecture for Dynamic Location-Based Services", in Mobile Networks and Applications Volume 8, Number 4, pp. 377-387, DOI: 10.1023/A: 1024531629631.

[12] M. Raento, A. Oulasvirta, and N. Eagle, "Smartphone's An Emerging Tool for Social Scientists". Sociological Methods Research February 2009 vol. 37 no. 3 pp. 426-454.

[13] AllJoyn https://www.alljoyn.org/ Retrieved: Aug, 2012.

[14] Nokia Instant Community http://research.nokia.com/news/9391 Retrieved: Jul, 2012.

[15] M. Jadliwala, J. Freudiger, I.Aad, J.Hubaux, and V.Niemi, "Privacy-triggered communications in pervasive social networks", World of Wireless, Mobile and Multimedia Networks (WoWMoM), 2011 IEEE International Symposium on, 2011, pp. 1 – 6.

[16] LocalSocial http://www.rococosoft.com/products/localsocial Retrieved: Feb, 2013.

[17] D. Namiot and M. Sneps-Sneppe, "Proximity as a service". In Future Internet Communications (BCFIC), 2012 2nd Baltic Congress on, pp. 199-205, IEEE, DOI: 10.1109/BCFIC.2012.6217947.

[18] D. Namiot and M. Schneps-Schneppe, "About location-aware mobile messages", International Conference and Exhibition on. Next Generation Mobile Applications, Services and Technologies (NGMAST), 2011, pp.: 48 – 53 ISSN: 2161-2889, DOI: 10.1109/NGMAST.2011.19.

[19] E. Friedman-Hill, "Jess in Action: Java Rule-based Systems", Manning Publications Company, June 2003, ISBN 193011089.

[20] C. L. Forgy "RETE: A fast algorithm for the many pattern/many object pattern match problem", Artificial Intelligence 19(1): pp. 17-37, September 1982.

[21] G. Jones and P. J. Brown "Context-Aware Retrieval for Ubiquitous Computing Environments" Mobile and Ubiquitous Information Access Lecture Notes in Computer Science, 2004, Volume 2954/2004, 371-374, DOI: 10.1007/978-3-540-24641-1_17.

[22] Namiot, Dmitry. "Web Intents as an Extension for M2M Open API", European Journal of Scientific Research 94.3 (2013), pp. 452-462.

[23] D. Namiot and M. Sneps-Sneppe "Advertising in Social Networks: Business-oriented Check-ins", AFIN 2011, The Third International Conference on Advances in Future Internet pp. 20-24.

[24] D. Namiot and M. Sneps-Sneppe "Customized Check-in Procedures", Smart Spaces and Next Generation Wired/Wireless Networking. Lecture Notes in Computer Science, 2011, Volume 6869/2011, pp. 160-164, DOI: 10.1007/978-3-642-22875-9_14.

AUTHOR BIOGRAPHY

Yousef Daradkeh received his B.Sc. degree in Computer Science and Engineering from the Technical University of Moldova (TUM) specializations in Computer`s, Information Systems and Networks in 2000, and his M.S. degree in Software Engineering from the Belarusian National Technical University (BNTU) in 2003. He received his Ph.D. degree in Computer Engineering and Information Technology from the Belarusian State University of Informatics and Radio-Electronics (BSUIR), and He has worked as a Post-Doctoral Research Fellow in the Department of Electrical and Computer Engineering, Schulich School of Engineering, University of Calgary (UC), Canada, in 2009. He has more than 36 published researches in the areas of wireless networking, artificial intelligence, web development, Java programming language, databases, context aware computing, cyber-security, information and economic security, LBS and geo services, knowledge representation and reasoning. He is as a Senior Lecture and Researcher in the faculty of Engineering, Salman Bin Abdul Aziz University (SAU).

Tariq(Moh'd Taisir) Al Omari received his B.Sc. degree in Bachelor of Computer Engineering from the Kharkiv National University of Radio Electronics of Ukraine in 2005, and his M.S. degree in Master of Engineering, Computer Systems and Networks from the Kharkiv National University of Radio Electronics of Ukraine in 2006. He received his Ph.D. degree in Engineering Sciences: Information Technology from the Lviv Polytechnic National University in 2010. He has more than 27 published researches in the areas of Computer-Aided design of MEMS at system level, The formalization of the MEMS automated design process by usage of petri Networks, Mathematical models for stress and displacement detection in the elastic part of the microphone, Development of subsystems for solving optimization problems with the help of genetic algorithms, Development and study of subsystem for solution of tasks of multicriterial optimization.

ISSN: 2319-5967
ISO 9001:2008 Certified
International Journal of Engineering Science and Innovative Technology (IJESIT)
Volume 3, Issue 1, January 2014

Dmitry Namiot received Ph.D. degree from the Lomonsov Moscow State University in 1990. He has been involved in the research and development of expert systems, real time applications, information retrieval system, geo applications and LBS systems. He is currently a senior scientist in the Open Information Technologies Lab, Faculty of Computational Mathematics and Cybernetics at the Lomonosov Moscow State University. Current research interests are mobile computing, context-aware applications and ubiquitous systems.