Using Network Proximity for Context-aware Browsing

SpotEx approach for delivering data to mobile subscribers

Dmitry Namiot
Lomonosov Moscow State University
Faculty of Computational Math and Cybernetics
Moscow, Russia
dnamiot@gmail.com

Manfred Sneps-Sneppe
Ventspils University College
Ventspils International Radio Astronomy Centre
Ventspils, Latvia
manfreds.sneps@gmail.com

Abstract **- This paper describes our SpotEx model for accessing to local data for mobile subscribers. This model is based on ideas of network proximity. In our concept, any existing or even especially created Wi-Fi hot spot or Bluetooth node could be used as presence sensor that can play a role of trigger for opening access for some content (or discovering some content). In our approach we can discover hyper local data as info snippets that are valid (relevant) for mobile subscribers being at this moment nearby some Wi-Fi access point. Info snippets could be either user-generated or collected from the social networks. As the prospect use-cases we can mention for example news and deals delivery in malls, news feeds for office centers and campuses, Smart City projects, personal classifieds and real world games.**

Keywords-Wi-Fi; Bluetooth; network proximity; collaborative location; indoor positioning; context-aware computing.

I. Introduction

This paper describes how the measurements collected from the wireless sensors on the mobile phone could be used for presenting an appropriate content (context-aware information) to mobile subscriber. It is an extended version of our paper from ICDT-2012 conference [1].

In the work that first time introduces the term ‘context-aware’, Schilit and Theimer [2] refer to context as location, identities of nearby people and objects, and changes to those objects. Other authors define context awareness 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, which sought to deal with linking changes in the environment with computer systems, which are otherwise static.

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 [3].

In our article, we are dealing with context-aware knowledge chunks. Let us start with the base element – location. It is the basic element for all the above mentioned definitions for context-aware. In general, getting location info for mobile subscribes could be pretty standard nowadays (GPS, cell-ID, assisted GPS [4]). The picture is much more complicated for indoor positioning. The Global Positioning System (GPS) loses accuracy indoor [5]. The system design for indoor positioning system (IPS) may vary. On practice, IPS systems can use various optical, radio, or even acoustic technologies. However, all of them require the utilization of their own protocols with their own API. The heterogeneous scenarios are typical for IPS.

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 Ekahau [4] is completely software-based and utilizes existing Wi-Fi access points installed in a facility and radio cards already present in the user devices. Companies could deploy also Wi-Fi based radio tags that use industry standard components that adhere to the 802.11 standards. This approach allows for the use of commercial off-the-shelf hardware and drivers to produce a standards-based radio tag that can communicate bi-directionally over the 802.11 networks. For embedded solutions, there is no need for the client to include a specialized tag, transmitter, or receiver.

Because of the entire use of standards-based hardware, such as 802.11b, 802.11g, and 802.11a, a standard Wi-Fi based solution rides the installed based and economies of scale of the networks and end user devices that are proliferating today. Based on the standard and already existing hardware, a company can install the system much faster and significantly reduce initial and long-term support costs. A common infrastructure supports both the data network and the positioning system, something companies strive for. The positioning system works wherever there is Wi-Fi coverage.

Actually, Figure 1 describes the main idea behind the Wi-Fi positioning: signal strength vs. distance. In addition to cost savings in hardware, a standards Wi-Fi based positioning system significantly reduces the potential for RF interference.

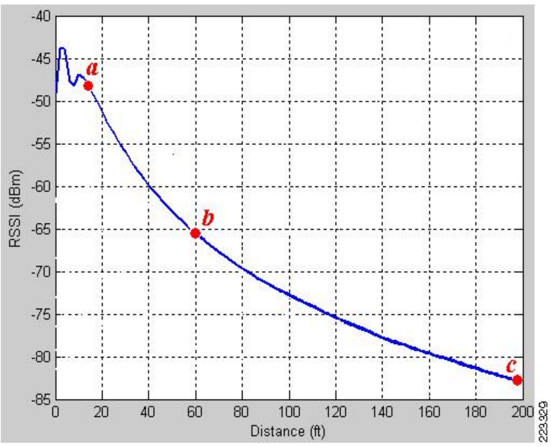

Figure 1. RSSI vs. distance [6]

The total Wi-Fi positioning system shares the same network along with other network clients, so there is no additional installation of a separate wireless networks (as RFID requires) that may cause RF interference with the existing wireless network [7]. The cited article shows that any commodity 802.11's equipment is surprisingly vulnerable to certain patterns of weak or narrow-band interference. This enables to disrupt a link with an interfering signal whose power is 1000 times weaker than the victim's 802.11 signals, or to shut down multiple access points, multiple channel managed network at a location with a single radio interferer.

Wi-Fi location positioning is based on a grid of Wi-Fi hotspots providing, in general, 20–30 meters location accuracy. For more accuracy, there needs to be more access points. There are many articles devoted to Wi-Fi positioning. For example, we can combine a reference point-based approach with a trilateration-based one etc. Several layers of refinement are offered based on the knowledge of the topology and devices deployed. The more data are known, the better adapted to its area the positioning system can be [8].

Figure 1 is very transparent, but it highlights also one of the main weaknesses for this approach. Yes, we can calculate distance to node, but for getting location data we need to know node's location. In other words, for all such systems we need a priori scene preparation. This process could be expensive and it is also almost closes the door for the dynamical systems. What if our basic nodes are moving?

But in the same time Figure 1 shows the way we can use for estimating the relative distance to our wireless node. Let us take at least two mobile devices and replace the distances with the relative values to our basic node. And as the next step we can replace digital values with some literal values that describe proximity (ranges for the relative distances). For example: High (e.g., same room), Medium (e.g., same floor) and Low (e.g., same building). For example, proximity classification for mobile devices using Wi-Fi environment similarity [9] describes an algorithm to compute lists of people and devices that are physically nearby to a mobile user based on the analysis of signals from existing wireless networks. The system evaluates proximity by classifying the degree of similarity of the Wi-Fi scan data through a statistical Gaussian Mixture Model [10].

Lets us mention also one more interesting approach: collaborative location (CL). And the most interesting approach for our future development is Cooperative Location-sensing. Cooperative Location-sensing system (CLS) is an adaptive location-sensing system that enables devices to estimate their position in a self-organizing manner without the need for an extensive infrastructure or training. A node tries to position itself on its local grid through a voting process in which devices participate by sending position information and casting votes on specific cells.

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 [11].

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.

Another area that is interesting for our approach is dynamic location based services. Some authors [12] classify them as application oriented LBS vs. classical content oriented. Content-oriented LBS aim to deliver relevant information depending on users' locations. For example, maps, points of interest, etc. Such LBSs are usually part of applications specialized in content delivery, such as a web browser or a front end for SMS messages. Applications oriented LBSs tailored to the user and delivered services dynamically on the basis of current location and execution context. In contrast to content-oriented LBSs, application-oriented LBSs provide a more powerful and richer interaction model, with autonomic installation and removal of dynamically needed components.

At this point, we are ready to make the last proposition before switching to the 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 for some requests for obtaining data (our final goal). This conclusion opens an opportunity to request data directly using our estimation for the location.

II. SPOTEX

On SpotEx model. What if we stop our traditional indoor positioning schema on the first stage: detection of Wi-Fi networks? 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) related to the Wi-Fi access points. Our rules will simply use the fact that the particularly Wi-Fi network is detected. And 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.

The first time this service SpotEx (Spot Expert [13] developed by Dmitry Namiot) was described by the authors in article published in NGMAST-2011 proceedings [14].

Technical details for SpotEx. SpotEx model does not require calibration phase and based on the ideas of proximity. Proximity based rules replace location information, where Wi-Fi host spots work as presence sensors. SpotEx approach does not require from mobile users to be connected to the detected networks. SpotEx uses only broadcasted SSID for networks and any other public information.

Technically, SpotEx contains the following components:

- Server side infrastructure. It includes a database (store) with productions (rules), rules engine and rules editor. Rule editor is a web application (it supports mobile web too) that lets work with rules database. Rules engine is responsible for runtime calculations. Note, that database is located outside of mobile device (at least, in the current implementation). But it could be positioned on the device too (e.g., for Wi-Fi Direct).

- Mobile application. Application is responsible for getting context info, matching it against database with productions and visualizing the output

SpotEx could be deployed on any existing Wi-Fi network (or networks especially created for this service – the most interesting case, see below) without any changes in the infrastructure. Rule editor lets easily define some rules (data chunks) to that network. Data chunk (message in our early papers) here is just some text that should be opened (delivered) to the end-user's mobile terminal as soon as the appropriate rule is fired. For example, as soon as one of the above-mentioned networks is getting detected via our mobile application. The word "delivered" here is a synonym for "being available for reading/downloading". For end-users the whole process looks like automatic (and anonymous) check-in.

Let us see the proximity marketing use cases. The whole process looks like an "automatic check-in" (by analogue with Foursquare, etc.) Some shop can deliver proximity marketing materials right to mobile terminals as soon as the user is near some selected access point. Rather than directly (manually or via some API) check-in at the particular place (e.g., similar to Foursquare, Facebook Places, etc.) and get back deals info, with SpotEx mobile subscriber can collect deals info automatically. The prospect areas, by our opinion, are information systems for campuses and hyper local news delivery in Smart City projects. Rules could be easily linked to the public available networks.

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. It is a dynamic LBS – services follows to the moved phone.

Figure 2. Wi-Fi hot spot on Android (Samsung Galaxy)

Figure 2 illustrates Wi-Fi hot spot setup on Android phone. Note, that SpotEx does not require open hot spot. All do we need is SSID info.

This use case is probably the most transparent demonstration of SpotEx model. We can open "base" network right on the mobile phone and attach rules for the content to that network. It is all do we need for creating a new information channel. There is no infrastructure except the smart phone itself.

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 includes the following objects:

Wi-Fi network SSID,
signal strength (optionally),
time of the day (optionally),
client ID (see below).

In other words it is a set of operators like:

IF network_SSID *IS* 'mycafe' *AND* time is 1pm – 2pm *THEN* { present the coupon for lunch }

A complete rule-set should be passed to the rule engine for further processing. The rule engine matches each rule (its left part - condition) in the rule set with given facts to decide whether to execute the right part or not. This is called pattern matching process and this process takes place repeatedly. In each cycle the list of facts may be modified: new Wi-Fi network may be detected or access point info may be removed from the list. These changes may cause previously unsatisfied conditions (or clauses) to be satisfied. So, during each cycle the set of rules satisfied must be maintained and updated. In most of the cases, actions of the rules change only a few clauses in the conditions. It is so called temporal redundancy. If a rule engine checks each rule to direct the search for all the facts even if most of them are not modified then it will slow down the process. We can avoid this unnecessary calculation by remembering what has already matched from cycle to cycle and then computing only the changes necessary for the newly added, modified or removed networks data.

Because our rules form the standard production rule based system, we can use old and well know algorithm like Rete [15] 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's left-hand-side. 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 [16].

The main advantage of Rete algorithm is the speed of calculation. Many rules often contain similar clauses or group of clauses. Rete algorithm caches the common components so that they need not be calculated again.

The main drawback of Rete pattern matching is obvious. It is memory intensive. Saving the state of the system using pattern matches and partial matches requires more memory.

The current implementation for mobile client based on Android OS. This application uses *WiFiManager* 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. And now we can prepare rules like this:

IF network_SSID *IS* 'mycafe' *AND* level > -60dBm *AND* time is 1pm – 2pm *AND* network_SSID 'myStore' *is not visible THEN* {present the deals for dinner}

Block {present the deals for dinner} is some data (information) snippet presented in the rule. Each snippet has got a title (text) and some HTML content (it could be simply a link to external site for example). Snippets are presenting coupons/discounts info for malls, news data for campuses, etc. Here is a typical example of user's snapshot:

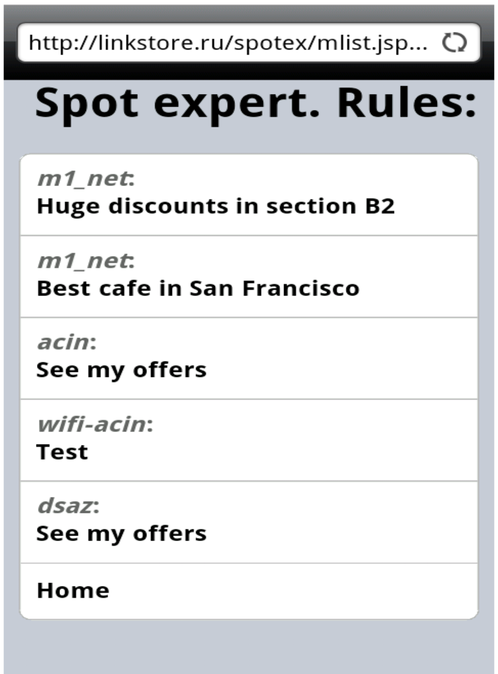

Figure 3. Rules from SpotEx

For Bluetooth mode we can use *BluetoothDevice* class. At least the following data could be used in our productions:

hardware address of Bluetooth device
friendly Bluetooth name of the remote device

Technically any snipped 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 included into SpotEx. Rule editor works in both desktop and mobile web. So, once again, just having an ordinary smart phone is enough for creating (opening) information channel for delivering hyper-local news data. The following snapshot illustrates how our rule editor looks like:

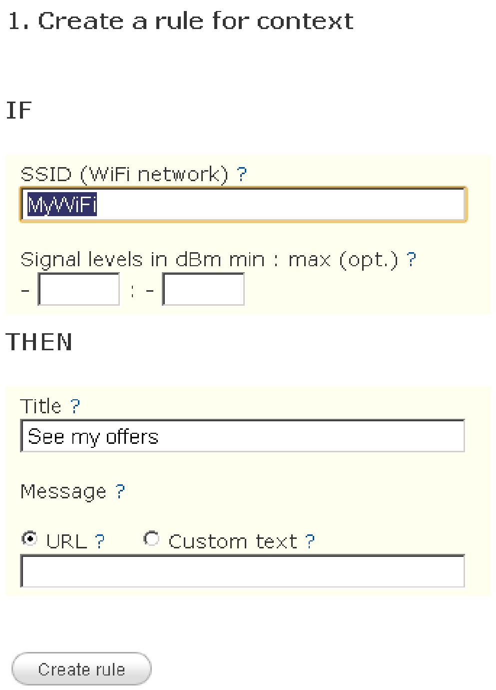

Figure 4. SpotEx rule editor

SpotEx as context-aware retrieval tool. In case of presenting our data as links to some 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 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. SpotEx rule editor creates mobile web page for the each provided data snippet and 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. In particularly, 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: matches existing rules against the existing context and makes the conclusions. Existing content here is a description for "Wi-Fi environment": list of hot spots with 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 that points to the appropriate data snipped. 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. Context-aware retrieval can be described as an extension of classical information retrieval that incorporates the contextual information into the retrieval process, with the aim of delivering information relevant to the users within their current context [17].

Because SpotEx presents data chunks (read – content) as mobile web pages we can follow to the standard security model for web applications. Any content could be public or private (e.g., password protected).

The context-aware retrieval model includes the following elements:

- a collection of discrete documents;
- a set of user's retrieval needs, captured in a query;
- a retrieval task, to deliver the documents that best match the current query, rated on the basis of a relevance measure;
- the user's context, used both in the query formulation and associated with the documents that are candidates for retrieval.

It is obviously, that all the above-mentioned tasks are components of SpotEx. The basic components of Context-Aware Retrieval are: a document collection, which contains the documents that may be retrieved annotated with details of their associated contexts, and the user's current context. In our case the collection of documents is defined by the right parts of our rules. And user's current context is a vector describes visible wireless networks. It is very important also that we can operate not only with snapshots of wireless environments (currently visible networks). Technically the past history is also available. Let us see for example the modern LBS applications based on check-ins. There is at least one important problem – we have not history for any particular check-in. All we can do is just to rely on the history of all previous check-ins. Think, for example, about some check-in in the big mall location. How the person in question reached the check-in point? What was done before?

History can be maintained automatically by the application. For example, a new record of the current context may be added to the history:

- by the time (e.g., every N seconds),
- whenever any retrieval request occurs,
- whenever the current context changes by more than a certain threshold amount,
- whenever user feedback indicates that a particular current context was important.

For example, the route within the building before check-in could provide valuable information for marketing. So we could add some like convoy discovery module to our rule engine. It is really very interesting topic that worth a special investigation.

A crucial property of many context fields is that they are continuous: as the user's context changes new information may need to be retrieved. Such continuous applications normally require fast retrieval, so that the user has the illusion that new information arrives immediately there is any change in their context [18].

Additional features. 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 proposed by Bluetooth marketing for example. And 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., but the SpotEx itself is anonymous. Unlike social networks like Foursquare you do not need to disclose your identity just for looking mall's deals for example.

But in the same time we still can collect some meaningful statistics in SpotEx. Because the model requires Wi-Fi to be switched on, we have automatically unique ID for the each client. It is MAC-address. It is actually global UUID. 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, opens that the particular data snipped. It means also (and it could be a part of our future research and development) that SpotEx model could be used for creating some sort of indoor analytics (like web analytics but for real places). The following figure illustrates a plot for clicks (opened data snippets per month – y axis) versus visits per month (x axis):

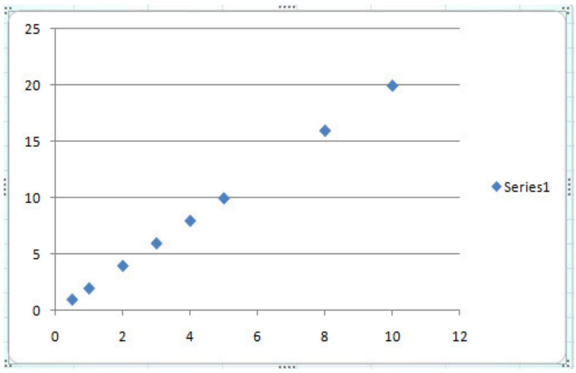

Figure 5. Clicks vs. frequency

Because mobile users in SpotEx model actually work with web pages, we can use pretty standard methods for web server log analysis for discovering 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. E.g., 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.

Note also, that for security reasons it is enough to keep hash-codes instead of the real MAC-addresses.

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). Now, using some JavaScript code we can loop over such div blocks and simply hide non-relevant from them. Custom HTML5 attributes are the best useful candidates here.

In HTML5, we can define custom attributes using the "data" attribute. The exact format is "data-XXXX", where "XXXX" is replaced with the desired custom attribute name, then set to the desired string value. For example:

```
<div id="my_coupon" data-ssid="myNetwork" data-rssi="-60">
Get discount for purchasing 2+ items.
</div>
```

Our attribute name must be prefixed with "data-" in order to validate in HTML 5. So in other words, while HTML 5 supports custom attributes, it does not allow for arbitrary attribute names. But that is enough for our development. For example, the above mentioned example could describe the visibility of div block (HTML text) for those mobile users who can detect the presence of

The appeal of custom attributes is that it lets you easily associate tidbits of information with an element, to be parsed later using JavaScript for example. There are two ways to retrieve the value of "data" attributes using JavaScript: the first is via getAttribute() method of JavaScript, and the second, by accessing the "dataset" property of the element:

```
var mydiv=document.getElementById('my_coupon’')

//Using DOM's getAttribute() property
var ssid=mydiv.getAttribute("data-ssid") //returns "myNetwork"
mydiv.removeAttribute("data-ssid") //removes "data-ssid" attribute entirely

//Using JavaScript's dataset property
var ssid=mydiv.dataset.ssid //returns "myNetwork";
mydiv.dataset.ssid=null; //removes "data-ssid" attribute
```

For doing that we need to make sure that our JavaScript code is aware about the current context. We can achieve that via a special light implementation of local web server. This web server, being hosted right on the mobile phone (on the Android in our case) responds actually only to one type of requests. It returns the current context (Wi-Fi networks) in JSON (JSONP) format.

Why do we need a web server? It lets us stay in the web domain only. There is a simple and clear instruction for web masters:

- add SpotEx script to your page

<script type = "text/javascript" src = http://localhost:8080/spotex.js> </script>

- describe your info snippets as div blocks:

<div rel="spotex" net="WiFi_SSID" levelMin="" levelMax="">
Your HTML code
</div>

Our "old" rules could be presented via collection of attributes.

In this case, JavaScript code loaded from local server will be able to proceed all the div blocks related to SpotEx, and set visibility attributes depending on the context.

Such simple trick let us make any existing HTML page "Wi-Fi context aware". Note that if our script is not available, the page will work as a "standard" HTML page.

This development is placed in line with the modern trends for moving the entirely development into web area. The final goal is to enable rapid development. This embedded web-server could play an important role in the data gathering too.

WiFi Chat. There is also a "side" effect for SpotEx application – WiFiChat service [19]. This mobile application uses the principles described in this article and offers communication tools (web chat and discussions groups) for mobile users nearby the same Wi-Fi access point.

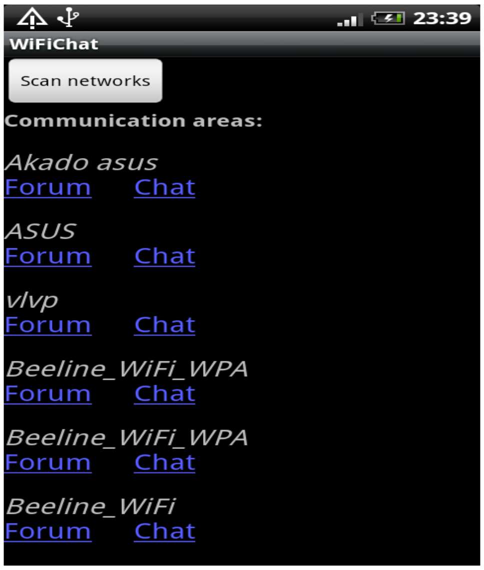

Figure 6. Wi-Fi chat

Think about it as "SpotEx with predefined content". The typical use case – we have Wi-Fi network in the train and this application automatically provides the discussions forum for the passengers. Or, keeping in mind that the "base" Wi-Fi network for this service could be opened right on the phone, this application can present personal forum (classified for example) as well as web chat for phone owner. This Android application is actually a wrapper for web mashup that combines HTML5 web chat engine and cloud based forums from Disqus.

It is the typical tool for the ad-hoc communications on the go. Think for example for chat in the train where Wi-Fi hot spot is provided, etc.

III. THE FUTURE DEVELOPMENT

Here, we see several almost obvious steps. At the first hand, it is open API. In the current implementation SpotEx front-end actually obtains data in JSON (JSONP) format from our server-side database.

As soon as API is going live, the next step is almost mandatory. It should be something that will simplify the development. The good candidates here are web intents [20] Web Intents is a framework for client-side service discovery and inter-application communication. Services register their intention to be able to handle an action on the user's behalf. Applications request to start an action of a certain verb (for example share, edit, view, pick, etc.) and the system will find the appropriate services for the user to use based on the user's preference. It is the basic.

Intents play the very important role in Android Architecture. Three of the four basic OS component types - activities, services, and broadcast receivers - are activated by an asynchronous message called as intent.

Intents bind individual components to each other at runtime (you can think of them as the messengers that request an action from other components), whether the component belongs to your application or another.

Created intent defines a message to activate either a specific component or a specific type of component - an intent can be either explicit or implicit, respectively.

For activities and services, an intent defines the action to perform (for example, to "view" or "send" something) and may specify the URI of the data to act on (among other things that the component being started might need to know). For example, our intent might convey a request for an activity to show an image or to open a web page. In some cases, you can start an activity to receive a result, in which case, the activity also returns the result in an Intent (for example, we can issue an intent to let the user pick a list of nearby images and have it returned to us - the return intent includes data in some format)

Going to our context aware browsing it means that our mobile devices will be able to present local data without low-level programming.

Web Intents puts the user in control of service integrations and makes the developers life simple.

Here is the modified example for web intents integration for the hypothetical web intents example:

1. Register some intent upon loading our HTML document

document.addEventListener("DOMContentLoaded", function() {
var regBtn = document.getElementById("register");

```
regBtn.addEventListener("click", function() {
window.navigator.register("http://webintents.org/spotex"
, undefined);
}, false);
```

2. Start intent's activity and pass it extra data (context info)

```
var startButton =
document.getElementById("startActivity");
startButton.addEventListener("click", function() {
  var intent = new Intent();
  intent.action = "http://webintents.org/spotex";

  intent.putExtra("WiFi_List", List_Of_Networks);
   window.navigator.startActivity(intent);
}, false);
```

3. Get local info snippets (note – in JSON rather than XML) and display them in our application

```
window.navigator.onActivity = function(data) {
  var output = document.getElementById("output");
  output.textContent = JSON.stringify(data);
};
}, false);
```

The key point here is *onActivity* callback that returns JSON formatted data. Additionally, web intents based approach is asynchronous by its nature, so, we do not need to organize asynchronous calls by our own.

Also, we are planning to add Bluetooth measurements too. But, by our vision, we should avoid the typical Bluetooth usage cases and does not use push proxy as per classical Bluetooth marketing. We think that the end users do at least not welcome push approach and it is the source of problems with Bluetooth proximity. Vice versa, in SpotEx Bluetooth nodes will be used the same manner we are using Wi-Fi access points – as presence triggers. In other words, we will add the ability to describe rules for Bluetooth nodes too. One of the biggest disadvantages for Bluetooth comparing to Wi-Fi is discovery times. Wi-Fi nodes could be discovered much more quickly than Bluetooth. As it is mentioned in [21], Bluetooth frame can disseminate much more information at once than WiFi (248 bytes vs. 32 bytes), but Bluetooth's discovery process is much more time-consuming than WiFi's (10.24 s versus 1.2 s).

SpotEx approach could be extended also towards accumulating some ideas from the collaborative locations. We can add trilateration terms (conditions) to our rules, but present them in terms of fuzzy logic (close than, relatively close, etc.). It helps us incorporate grid data in case of many devices without any infrastructure preparation.

The next prospect area we are going to bring network proximity in is social data streams. It worth a separate article, so le us briefly describe an idea. For pairing sensors data and social streams (e.g., Twitter's stream), we can describe thematic stream as data (topics) discovered from the tweets and data recorded from our sensors. Of course, data from sensors depends on the nature of sensors (e.g., proximity sensors, sounds streams, etc.) but in our case, as it was stated above, the set of sensors is limited to wireless networks sensors.

Our idea of integration network proximity and social networks 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 some 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 record in the social network – check-in. It could be customized of course. Business can create own forms for check-in records [22], 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 ID's (e.g., nick names) for accounts.

And how can use 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. Because in out proximity based system there are no lists of location in the traditional form. Each our "location" 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 only nearby check-ins. In this approach user is simply unaware about other "places" unless the movement and new check-in.

At the second – we can show (search) social streams nearby. Via public API we can read data feeds for users (if it is possible, of course, and an appropriate stream is not protected).

As it is already mentioned above, the historical data collected by the SpotEx application could provide a meaningful statistics. History in SpotEx is anonymous set of network environment records (snapshots), collected by the mobile users. It lets is threat SpotEx as a tool for social dynamics research too. The classical example of reality mining [23] collected a lot of information from the mobile phones, includes call logs, Bluetooth devices in proximity, cell tower IDs, application usage, and phone status (such as charging and idle). SpotEx is much more accurate in this tight privacy connected topic and can collect only wireless info data. But it means, on practice, that we can collect some like trajectories where instead of location we can use info about wireless nodes. And this idea technically lets us apply some existing studies to our movement log.

At the first hand adding history movement we can introduce context for user's check-ins. We will be able to analyze for example, the movement patterns for reaching the destination (check-in) point. Such information simply does not exist for the traditional LBS applications. And patterns extracted from the log could be used potentially for creating new business rules. Let us see for example convoy discovery. Given a collection of trajectories, it is of interest to discover groups of objects that travel together for more than some minimum duration of time. A number of applications may be envisioned. The typical applications, mentioned in the scientific papers are: the identification of delivery trucks with coherent trajectory patterns may be used for throughput planning, the discovery of common routes among commuters may be used for the scheduling of collective transport, the identification of cars that follow the same routes at the same time may be used for the organization of carpooling, etc. But one specific task could be interesting in the context of proximity marketing (what is one of the main applied areas for SpotEx): provide some special offers for those reached the final point in the group (group buying, group discount, etc.).

The next area we are going to pay attention to is Wi-Fi Direct specification. Wi-Fi Direct devices can connect directly to one another without access to a traditional network, so mobile phones, cameras, printers, PCs, and gaming devices can connect to each other directly to transfer content and share applications anytime and anywhere. Devices can make a one-to-one connection, or a group of several devices can connect simultaneously. They can connect for a single exchange, or they can retain the memory of the connection and link together each time they are in proximity [24].

As per Wi-Fi Direct spec, a single Wi-Fi Direct device could be in charge of the Group, including controlling which devices are allowed to join and when the Group is started. All Wi-Fi Direct devices must be capable of being in charge of a Group, and must be able to negotiate which device adopts this role when forming a Group with another Wi-Fi Direct device. The device that forms the Group will provide the above described dynamically assembled web page with discovered services. It is how SpotEx could be extended to Wi-Fi Direct. Currently SpotEx requires internet connection in order to match network context against rules served as web based database. In Wi-Fi Direct environment we can keep this database on the one of Wi-Fi enabled devices (read – simply on another mobile phone) and perform matching without internet connection. The drawback here is the need for pairing during the connection setup.

IV. CONCLUSION

This paper describes a new context-aware browsing model for mobile users developed on the ideas of Wi-Fi proximity. Service can use existing as well as the especially created (described) Wi-Fi networks as presence triggers for discovering user-defined content right to mobile subscribers.

The proposed approach is completely software based. It is probably its biggest advantage. For using SpotEx you need nothing except the smart phone. So, there are no prior investments in the hardware. Also this approach supports ad-hoc solutions and 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, etc.

We highlights also our plans for the future research and development that include such prospect directions as analyzing social behavior via network proximity data, integration with social feeds, creating local web server for serving context data, Wi-Fi direct and network proximity, using web intents for context-aware data.

ACKNOWLEDGMENT

Many thanks to the anonymous ICDT-2012 reviewers for the reading and suggested improvements to the paper.

REFERENCES

[1] D. Namiot and M. Sneps-Sneppe “Wi-Fi Proximity and Context-aware Browsing” ICDT 2012, The Seventh International Conference on Digital Telecommunications ISBN: 978-1-61208-193-9, Chamonix, France April 29 - May 4, 2012 pp. 1-7

[2] G. Schilit and B. Theimer Disseminating Active Map Information to Mobile Hosts. IEEE Network, 8(5) (1994) pp. 22-32

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

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

[5] Comparison of Wireless Indoor Positioning Technologies http://www.productivet.com/docs-2/Wireless_Comparison.pdf, Retrieved: Nov, 2012

[6] Y. Daradkeh, D.Namiot, and M Schneps-Schneppe: Context-Aware Browsing for Hyper-Local News Data (CABHLND). International Journal of Interactive Mobile Technologies (iJIM) vol. 6 N.3, 2012 pp. 13-17

[7] R. Gummadi, D. Wetherall, B. Greenstein, and S. Seshan Understanding and mitigating the impact of RF interference on 802.11 networks. SIGCOMM '07 Proceedings of the 2007 conference on Applications, technologies, architectures, and protocols for computer communications ACM New York, NY, USA ©2007 ISBN: 978-1-59593-713-1 DOI=10.1145/1282380.1282424, pp. 385-396

[8] F. Lassabe, P. Canalda, P. Chatonnay, and F. Spies “Indoor Wi-Fi positioning: techniques and systems” Annals of Telecommunications Volume 64, Numbers 9-10, pp. 651-664

[9] A.Carlotto, M. Parodi, C. Bonamico, F.Lavagetto, and M. Valla “Proximity classification for mobile devices using wi-fi environment similarity” MELT '08 Proceedings of the first ACM international workshop on Mobile entity localization and tracking in GPS-less environments 2008 pp. 43-48

[10] M. A. T. Figueiredo and A. K. Jain, "Unsupervised Learning of Finite Mixture Models", IEEE Transactions on Pattern Analysis and Machine Intelligence 24, 2002 pp. 381–396

[11] C. Fretzagias and M. Papadopouli Cooperative Location-Sensing for Wireless Networks, Proceedings of the Second

IEEE International Conference on Pervasive Computing and Communications (PerCom'04), p.121, March 14-17, 2004.

[12] P.Bellavista, A.Kupper , and S.Helal Location-Based Services: Back to the Future, Pervasive Computing, IEEE, Vol.7, issue 2, 2008, pp. 85-89

[13] Y. Daradkeh, D. Namiot, amd M. Sneps-Sneppe Spot Expert as Context-Aware Browsing. Journal of Wireless Networking and Communications 2012, vol. 2, N.3 pp. 23-28 DOI: 10.5923/j.jwnc.20120203.03

[14] 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 14-16 Sept. 2011 pp. 48-53 doi: 10.1109/NGMAST.2011.19

[15] E. Friedman-Hill Jess in action: rule-based systems in Java. Manning Publications Co. Greenwich, CT, USA 2003 ISBN: 9781930110892

[16] 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

[17] P. J. Brown and G. J. F. Jones. Context-aware retrieval: Exploring a new environment for information retrieval and information filtering. Personal Ubiquitous Computing, 5(4): pp. 253–263, 2001.

[18] P. J. Brown and G. J. F. Jones. Exploiting contextual change in context-aware retrieval SAC '02 Proceedings of the 2002 ACM symposium on Applied computing pp. 650 – 656, ISBN:1-58113-445-2 doi:10.1145/508791.508917

[19] D.Namiot Context-Aware Browsing -- A Practical Approach. Next Generation Mobile Applications, Services and Technologies (NGMAST), 2012 6th International Conference on, pp.18-23 DOI: 10.1109/NGMAST.2012.13

[20] Web Intents: http://webintents.org/, Retrieved: Nov, 2012

[21] A. C. Champion, X. Li, Q. Zhai, J. Teng, and D. XuanEnclave: Promoting Unobtrusive and Secure Mobile Communications with a Ubiquitous Electronic World Department of Computer Science and Engineering, The Ohio State University 2012

[22] 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

[23] N. Eagle and A. Pentland "Reality mining: sensing complex social systems" Personal and Ubiquitous Computing archive, Vol. 10 Issue 4, March 2006 pp. 255 - 268 doi:10.1007/s00779-005-0046-3

[24] Wi-Fi CERTIFIED Wi-Fi Direct http://www.cnetworksolution.com/uploads/wp_Wi-Fi_Direct_20101025_Industry.pdf, Retrieved: Nov, 2012