Context-aware Browsing – a practical approach

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

Abstract— **This paper describes one practical approach for creating context-aware browser for mobile subscribers. Our model uses mobile phone as proximity sensor. In our concept, any existing or even especially created Wi-Fi hot spot or Bluetooth node could be used as presence sensor that can open (discover) access for 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 the detected wireless nodes. And an appropriate mobile service (context-aware browser) can present that information to mobile subscribers. As the potential use-cases for the proposed approach we can mention for example news and deals delivery in retail, delivering news feeds for office centers and campuses, Smart City projects, personal classifieds and real world games.**

Keywords-context-ware computing; Wi-Fi; proximity; collaborative location; indoor positioning.

I. INTRODUCTION

In the work that first introduces the term 'context-aware', Schilit and Theimer [1] refer to context as location, identities of nearby people and objects, and changes to those objects. 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. Most of 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.

A. Day [2] defines context is any information that can be used to characterize the situation of an entity. An entity is a person, place, or object that is considered relevant to the interaction between a user and an application, including the user and applications themselves. This definition makes it easier for an application developer to enumerate the context for a given application scenario.

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. In our definition each chunk is data fragment, associated with the context.

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 [4]), but everything is getting more complicated as soon as we need indoor positioning. 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 [5].

One of the most used approaches to indoor location is Wi-Fi based positioning. A standard Wi-Fi based positioning system 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.

In addition to cost savings in hardware, a standards Wi-Fi based positioning system significantly reduces the potential for RF interference [6].

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 [7].

Lets us mention also one more interesting approach: collaborative location (CL) [8]. And the most interesting approach for our future development is Collaborative 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.

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. 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 interesting approach is dynamic location based services [9]. The main idea here is to allow applications to select services that are specifically associated with their current location.

Our own system in this area (SpotEx – Spot Expert [10]) uses proximity ideas for discovering a new context to mobile users. 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). So, why do not request data directly if we can estimate location?

II. SPOTEX

What if instead of the location info we will use proximity calculation relating the visible wireless nodes (Wi-Fi access points and Bluetooth nodes)? E.g., the detection of Wi-Fi networks already provides some information about the location – just due to local nature of Wi-Fi network. And as the second step we will 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 [10] developed by Dmitry Namiot) was described by the authors in article published in NGMAST-2011 proceedings [11]. This paper describes the next development in this approach as well as outlines the nearest plans.

Obviously, our SpotEx model is based on the ideas of Wi-Fi proximity. Wi-Fi hotspots work here as presence sensors. But we are not going to connect mobile users to the detected networks and our suggestion does not touch security issues. We need only SSID for networks and any other public information.

So, this service contains the following components:

- database (store) with productions (rules) associated with Wi-Fi networks

- 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

How does it work? We can take any exiting 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, Facebook 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 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. Actually, we are getting dynamic LBS here: phone itself could be moved and so, the available data will be de-facto moved too.

Figure 1. Wi-Fi host spot on Android

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 it is all do we need for creating a new information channel. There is no infrastructure except the smart phone and we do not need a grid of devices as per CLS models.

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 (and/or BlueTooth nodes) 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), history of visits (tracks) device ID (MAC address)

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 }

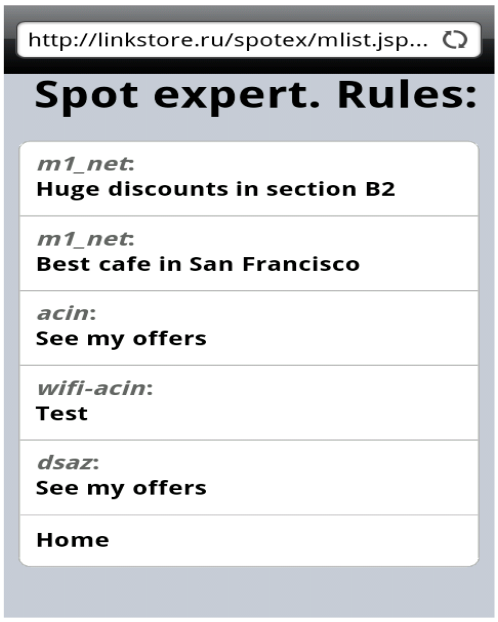

Figure 2. SpotEx rules

Because our rules form the standard production rule based system, we can use old and well know algorithm like Rete [12] 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 [13].

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 information about nearby networks (the network name, the address of the access point, the channel frequency and signal level). All the collected data could be used in our productions. So, we can prepare rules like this:

IF network_SSID IS 'mycafe' AND level > -60db 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.

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.

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 exiting rules against the exiting 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 [14].

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.

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.

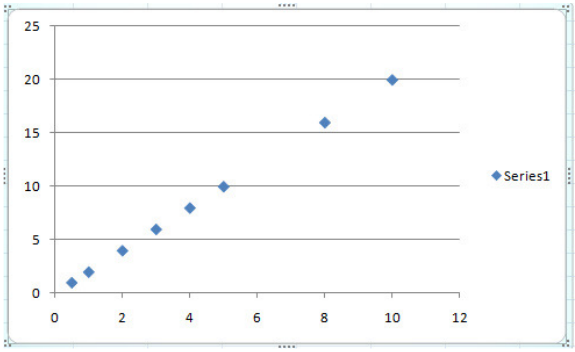

Figure 3. Clicks vs. frequency

For example, Figure 3 illustrates a plot for clicks (opened data snippets per month – y axis) versus visits per month (x axis).

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.

In general, the fact that we could have for the proposed model the standard web-log opens the door for many existing approaches in data analysis. For example, Hidden Markov Models (HMMs) based on local sequence patterns discovered in data. We can try to find interesting sequences whose frequency in the database differs from that predicted by the model [15].

And note also that using statistical data analysis right in the discovering rules is a huge step forward from the model that simply calculates proximity. But this movement obviously requires the real time processing for big data.

Our next play with big data belongs to the history involvement into data discovery. For example, in the modern LBS applications that are mostly circling near the idea of "check-in" we lack the history almost completely. Suppose I have a new check-in in Foursquare. How come I to this place? In the ordinary web browsing any hyperlink click can have a referrer field. Where are referrers for LBS? And context-aware applications can fill this gap. Lets us see the project Funf [16] for example.

Funf Probes are the basic data collection objects used by the Funf framework. Each probe is a contained unit responsible for collecting a specific type of information. These include data collected by on-phone sensors, like accelerometer or GPS location scans, but also many other types of data that can be collected through the phone - from information on the media files stored on the device to call-logs, application usage, or browsing history. Each probe can be remotely configured to be enabled/disabled, what scan intervals or triggers should be used, as well as other parameters. In other words Funf is a rich data logger. And that log could be a source for data discovery too. In the current version SpotEx uses a snapshot for currently "visible" wireless networks. But we can use also a history: a set of wireless networks users saw prior that. Think for example about a big mall. The route (path) used for reaching our current position could be logged. This path could be presented also in terms of proximity: a set on nodes some user was nearby prior to the current place. And "path" info could be used in our discovery rules.

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. 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 (wireless nodes) in JSON (JSONP) format.

Why do we need a web server? It lets us stay in the web domain only. Our "old" rules could be presented via collection of attributes for HTML tags.

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.

There is also a “side” effect for SpotEx application – WiFiChat service [17]. 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. 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:

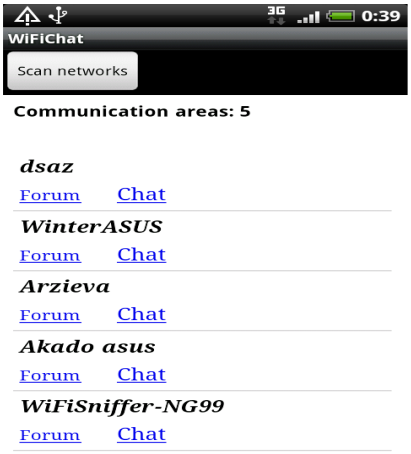

Figure 4. Wi-Fi Chat application

It is the typical tool for the ad-hoc communications on the go.

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 [18] 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:

```
window.navigator.register("http://webintents.org/spotex"
, undefined);
```

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);
```

Obviously, that it is much shorter than the long sequence of individual calls as per any Open API. 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.

The next step could be related to the connecting data from the social networks. Actually, any the above described rule associates data snippets (*then* part in our rules) with the wireless networks context info. We can threat this wireless networks context as some form of the digital signature (or hash code) associated with our data. What if we open the ability to use the similar hash for data feeds from social networks? E.g., some author will be able to associate (temporarily) own Twitter timeline with the current context. After that any other participant will discover this Twitter's stream as hyper-local stream. Author can write tweets without the putting location info into them and without using the special hash tags – the whole stream will be visible as "local" for all users within the same wireless networks context.

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 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 [19].

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.

IV. CONCLUSION

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

The proposed approach is pure software based as well as highly flexible and extendable. For using SpotEx you need nothing except the smart phone and 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.

REFERENCES

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

[2] A. Day, "Understanding and Using Context" (2001). Personal and Ubiquitous Computing Vol. 5, Number 1, pp. 4-7, DOI: 10.1007/s007790170019C.

[3] 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: Jan, 2012>

[6] 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

[7] 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

[8] 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.

[9] Zhang, C., Huang, Y.: Cloaking locations for anonymous location based services: A hybrid approach. GeoInformatica 13(2), 2009, pp. 159-182

[10] SpotEx Project: http://spotex.linkstore.ru, <retrieved: Jan, 2012>

[11] 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

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

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

[14] 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.

[15] S. Jaroszewicz Using interesting sequences to interactively build Hidden Markov Models DATA MINING AND KNOWLEDGE DISCOVERY Volume 21, Number 1, pp. 186-220,

[16] Funf Open Sensing Framework http://funf.media.mit.edu/ <retrieved: Mar 2012>

[17] Wi-Fi chat Project: http://wifichat.linkstore.ru, <retrieved: Jan, 2012>

[18] Web Intents: http://webintents.org/, <retrieved: Feb, 2012>

[19] Wi-Fi CERTIFIED Wi-Fi Direct http://www.cnetworksolution.com/uploads/wp_Wi-Fi_Direct_20101025_Industry.pdf, <retrieved: Jan, 2012>