Proceedings of the 2014 Federated Conference on Computer Science and Information Systems pp. 925–932
DOI: 10.15439/2014F192
ACSIS, Vol. 2

Graph Based Messaging APIs – concept and implementation

Michał Cieszko(1,2)
(1) Orange Labs
ul. Obrzeżna 7
02-691 Warsaw, Poland
(2)Warsaw University of Technology Faculty of Electronics and Information Technology
ul. Nowowiejska 15/19
00-665 Warsaw, Poland
Email:
cieszko.michal@gmail.com

Jarosław Legierski
Orange Labs
CBR
ul. Obrzeżna 7,
02-691 Warsaw, Poland
Email: jaroslaw.legierski@orange.com

***Abstract*—This paper presents an idea of new communication Application Programming Interfaces (APIs) based on graph oriented data stored in database. The use case described in paper allows the sending of traditional telecommunication messages (SMS or USSD) to dedicated people identified as best recipients based on their skills and location. A decision algorithm implemented in API, besides the organizational data takes into consideration geo-location of user' mobile phones.**

I. Introduction

NOWADAYS, with the Internet based on Web 2.0 we can observe a large expansion of social frameworks and services. Social portals such as an Facebook [1], Google+ [2], Twitter [3] and lately Instagram [4] have become very popular, especially among young people. The natural representation of connections between people linked by social networks or network of professionals (e.g. Linked In [5] or Viadeo [6]) is represented by data structures in a graphs form. This data structure in a very good way represents: people (e.g. as a graphs vertices), connection between them (graphs edges) and other parameters such as interests and hobbies (represented by attributes associated with its nodes or edges stored in graph). During the last 20 years in the Internet we can observe a large expansion of data generated by people and for people. This trend called "Big Data" in large part results generation of data sets which have the structure of a graph and can be stored as graphs attributes.

The paper is organized as follows: Section I presents a short introduction. Section II contains an overview of some related work. Section III describes the idea of a graph based messaging communication services. Section IV elaborates on our proof-of-concept prototype and system architecture. Section V presents future work, plans and intentions. The last section VI concludes this paper.

II. Existing Solutions

A. APIs exposed graph oriented data sets

Graph data sets and based on them services are used by many portals and application in the Internet. In this paper we concentrate on graph based data exposed in Application Programming Interfaces (API).

The most interesting API, based on a graph is offered by Facebook: (The Graph API [7]). This API allows: reading publishing, updating and deleting of records correlated with Facebook users activities: accounts, posts, groups, events, location and places.

Another information set (defined as a free, knowledge graph with millions of people, places, and things.) offers Google as an Freebase - Google's Knowledge Graph [8]. The Freebase APIs support a function responsible for searching and manipulating of this open database repository. Google also expose an additional data description language (Search Cookbook [9]), which allows including additional semantic information in text queries.

Another similar API based on Twitter data – GraphEdge API [10],[11] which allows for Twitter data manipulation has lately evolved to GraphEdge Pro service [12]. This solution helps PR marketing agencies track, report, and analyze the activity of their customers Twitter accounts.

The next service GraphMuse API [13] – is a solution dedicated for developers interested in the exploration of Facebook data. This easy RESTfull API returns: friends, groups, family or collective interests data sets in the JSON format.

CicerOOs Semantic Graph API [14] allows semantic searches of Points of Interest (POI) related to geographical objects (data correlated with tourism information in Italy). Apart from simple search, this API

978-83-60810-58-3/$25.00

can return the suggested possible topics of interest (e.g. restaurants, castles, galleries, parks) taken from the semantic graph.

A completely different, from above mentioned area of application is provided in the *RunKeeper's* Health Graph API [15] – which offers developers access to all of health oriented information such as: nutrition information, workouts, sleep and body measurements data, blood glucose levels etc.

The *Viadeo* [16], service which operates professional social networks provides the Graph API that allows developers to integrate professional social context with websites, applications and services. A social portal using this API offers developers access to public information on some objects: members, connections, jobs, articles, news, comments, etc.

Table 1. Exposed in the Internet graph oriented APIs

API	Functionality	Data Source
Facebook The Graph API	Data search and manipulation	Own data
Freebase Google API	Data search and manipulation	Own data
GraphEdge API	Data mining second level API	Twitter API
GraphMuse API	Data sets access simplification	Facebook API
CicerOOs Semantic Graph API	Simple and semantic data search	Own data
RunKeeper's Health Graph API	Data search and manipulation	Own data
Viadeo Graph API	Data search and manipulation	Own data

B. *Messaging applications and services based on graph data*

In the Internet some applications and services can be identified, which offer messaging functions based on information from graph oriented data sets. The *Meetbymaps.com* portal [17] offers the Instant Messaging (chat) communication service for Facebook users based on location and events (realtime geolocated chat) using the Facebook Graph API. Another portal, based on Facebook graph *WishMindr* [18] is the freely accessible virtual wish list. Using this application the end user can prepare wish lists from any site and add preferred gifts using the *WishMindr* search. *WishMindr* also offers an email communication channel used for reminders about wish lists for e.g. friends and family.

The next solution *oGoWoo* www.ogowoo.com [19] is a free advertising system. The *oGoWoo* service uses the Facebook API to post promotional and advertising messages on users Facebook walls on their behalf. As benefits, the application users get gifts and discounts. Another service *Decoda* [20] allows you to share favorite lyrics between Facebook users (e.g. marking favorite lines from a song and posting them to Facebook as a status or posting lyrics to Facebook wall). The service can also post information to a friend's Facebook wall thanks to the Facebook Graph API. Table 2 provides examples of communications applications using Facebook Graph API.

Table 2. Example communication driven applications based on Facebook Graph APIs

Application	Communication channel
Meetbymaps	IM (chat)
WishMindr	e-mail
oGoWoo	Facebook walls
Decoda	Facebook walls

C. *Telco Messaging based on API*

SMS, MMS and USSD based messaging is known in telecommunication for many years. Short Messaging Services and Multimedia Messaging Service can be originated (sent) and terminated (received) by network terminals (mainly phones). Unstructured Supplementary Service Data (USSD) can by initiated by phones (in the form of USSD codes e.g.*100#, *665*1#). Communication Service Providers (CSP) in the last decade of XX century, have started opening their networks to external developers by exposition of the Telco API in the Internet. This concept extends traditional device based messaging model to an API originated or terminated communication function. In the network architecture it appears as an additional exposed API element – Service Delivery Platform (SDP). SDP using API in Web Services form can provide external developers with a large set of communication functions from CSP networks such as send/receive: SMS, MMS or USSD, mobile terminal location, click to call, payment etc. for new innovative applications [21],[22],[23],[24]. Basic Telco APIs set is standardized by GSMA as OneAPI specification [25] based on RESTful architecture style Web Service [26]. Another specification ParlayX [27] uses Service Oriented Architecture and SOAP protocol [28].

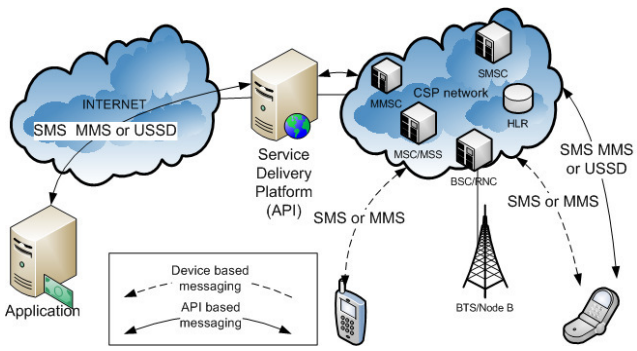

Fig 1.Telco messaging: based on devices and APIs

III. THE IDEA OF GRAPH BASED MESSAGING SERVICES

This chapter presents the idea of Graph Based Messaging (GBM) services exposed in APIs form. There are many solutions used in a single system that allow the communication of groups of people depending on their status, location or additional features e.g. the solutions based on the Unified Communications paradigm [29],[30], [31] or applications described in chapter II. The application based on the Unified Communication are dedicated to enterprise sector and mostly allow end users communicate using: voice calls, voice and web conferences, or chats. In UC, a system's subscribers are addressed using names, e-mails or phone numbers and applications are trusted and private from an enterprises point of view. Companies in the B2B sector have a lot of information which is graph oriented: organizational structures, information about calls and actions from private telecommunication or IT systems. This information can be very useful for the development of new applications and services but should be extracted from many ICT systems, structured and exposed in the form of dedicated Application Programming Interfaces. The second problem is related to the confidentiality of the exposed data. A dedicated API should also isolate sensitive private information which greatly simplifies the development of an external application from a legal point of view. The solution described in chapter 2 solutions use a graph API exposed in the public domain (in the Internet) and exposes a user's identifier (e.g e-mails or stored in Facebook data) based on permissions granted by users.

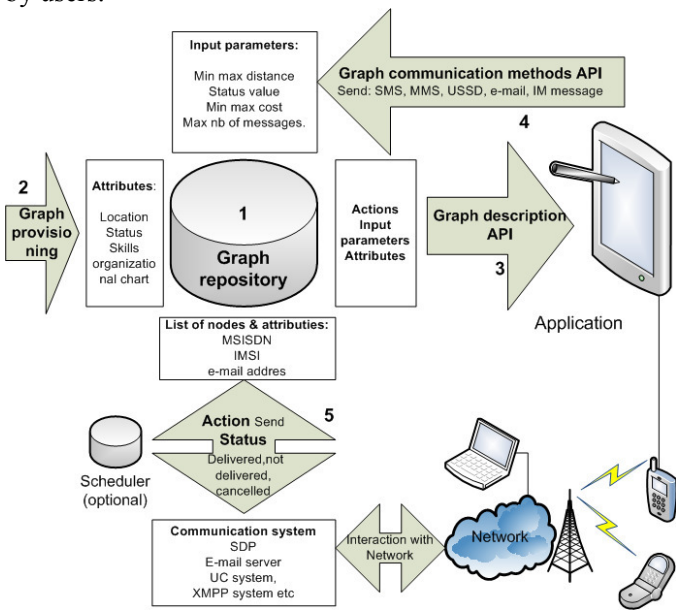

Fig 2. Idea of a Graph Based Messaging APIs

In the solution presented above the main element is **Data Repository** (1) – this is a database which contains the graph oriented data sets. Next, a very important is the **Graph provisioning** (2) – an interface for graph data sets maintenance (add/remove nodes, attributes), this element can be implemented in the form of: a web interface, an API or a batch file.

Graph description API (3) – is a high level API for external application defined as: a graph structure, graph attributes, and possible actions. By using this API an external application can automatically get graph meta data description.

Graph communication methods API (4) – a set of APIs dedicated to invoking some actions (e.g. send messages, display graph data sets etc.)

Actions (communication functions) (5) – executed by a graph messaging service, interacts with communication systems such as e.g.:

- Service Delivery Platform – sending SMS, MMS, USSD, Terminal Location,
- SMPT server – sending and receiving e-mails,
- UC system – read/set presence status , monitor office phone calls, invoke voice and web conferences etc.
- XMPP server – dedicated for sending Instant Messaging information (chats).

The idea of an API based on graph oriented communication services presented in this paper allows developers to think about communication more widely. The presented concept gives external developers a simple programming interface and provides following advantages:

- sensitive user data (e-mails, MSISDN etc.) is not exposed,
- developers do not need to implement graph specific algorithms on the application side (e.g. clustering, shortest path)
- developers are isolated from telecommunication specific protocols

Using the above presented concept allows developers to create applications in a simpler way. The majority of useful and needed (e.g. mobile) applications were invented by people from normal society. Graph oriented data exposed by social portals, communication providers or enterprises has a lot of potential. Developers, companies and people from societies are looking for a new ideas and applications. Graph based messaging API is the answer to computerized society needs.

IV. TEST DATA AND MEASUREMENT

As a test data set, in the presented in this paper proof of concept of a Graph Based Messaging APIs, the

organizational structure of the department of Open Communication Systems and Open Data Department in Orange Poland R&D Center was used. This structure is presented in Fig 3 and contains:

- graph vertices- represented by people (names)
- graph edges – based on ongoing projects and other activities virtual teams
- graph attributes - associated with vertices: employees: e-mails, phone numbers, terminal location and people skills. Skills are defined in following form:

DEV - developer
IN – IN network specialist
API – API expert
SDP – Service Delivery Platform expert
BD – Big Data expert
LOC – mobile terminal location specialist
DM – Data Mininig expert
STAT – Statistics
STD - Student

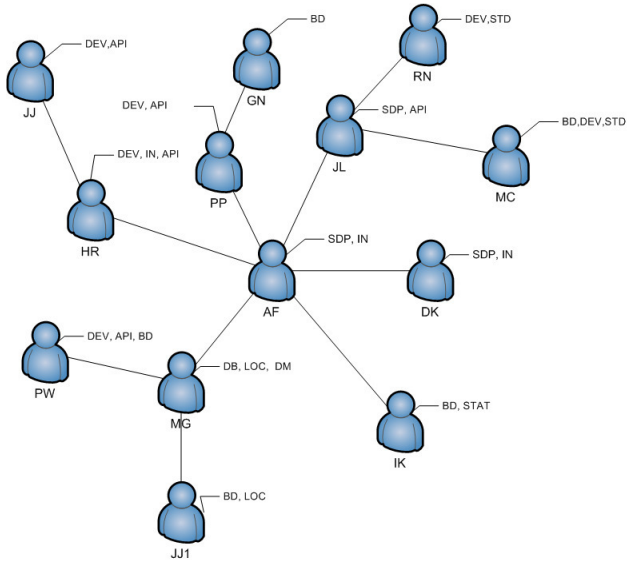

Fig 3. Test data set

Mathematically, a graph G is represented as G(V, E, A) where V are vertices (people), E a set of edges connecting some vertex in V (virtual teams), A set of attributes A={a1,a2, a3 , …an} is defined as list of skills, contacts (phone numbers, e-mails) and mobile terminals geolocation coordinates.

V. SYSTEM ARCHITECTURE

In this section a Graph Based Messaging API system architecture realizing messaging functions is described. The developed system, based on a graph database as a source of information, offers end users messaging functions such as SMS, USSD or e-mail in an Application Programming Interfaces form. The project integrates Gmail, Google Maps and Orange API services and creates a coherent communication system which it's high level architecture is presented in Fig.4.

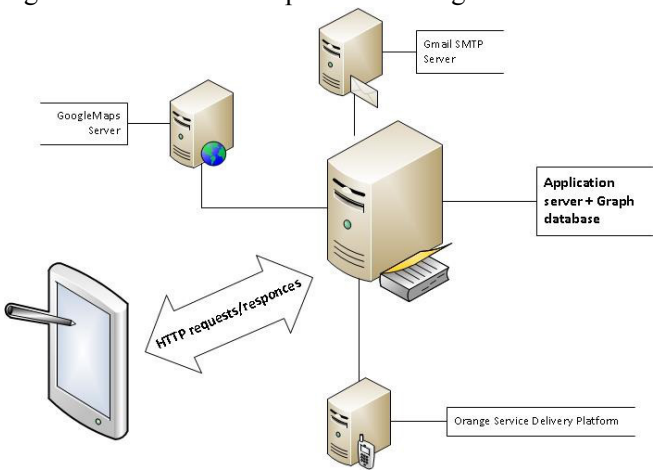

Fig 4. A high level architecture of the graph based messaging API

A. *Programing environment*

The Graph Based Messaging API system was developed in JRuby [33] – a Java implementation of the Ruby programming language and deployed on Linux Ubuntu server. The application server is exposed in the Internet using a public IP address. JRuby provides a set of classes and Standard Libraries for the Ruby language. The JRuby environment also offers a possibility of downloading many additional libraries (External Gems) created by independent developers and external companies. An example of an external gems, used during system implementation, are libraries that connect system with an open-source Neo4j graph database developed by Neo Technology. An embedded version of Neo4j database in High-availability Cluster mode was used in the graph based messaging API system as data repository.

B. *Database*

The system presented in this paper system was developed based on the Neo4j database in community open source version dedicated to the creation and testing of application prototypes. The Open Communication Systems and Open Data Department structure presented in Fig.3 was used as a test data set. A traditional, plain relational data model will not be efficient with these kinds of data sets because of a large number of relations between people which is characteristic for social networks. In comparison with relational databases, graph databases such as Neo4j [34] are faster and generate less processor load. Neo4j in the embedded mode offers an opportunity to store, on the local server, data consisting of users, places and competences entities with basic information like personal data or dynamic parameters defining unique terminal location or its actual status. Each of the users (vertices, nodes) is in relation with others by

assignment to a common workplace or some competencies shown in figure 5. All relationships are stored as graph attributes in both of the related entities, defined as incoming or outgoing. A graph oriented structure allows access to all nodes connected by a relationship with and additional set of parameters. From a data model point of view, operations like adding new relationships or attributes are easy to perform and do not demand a reorganization of data structure.

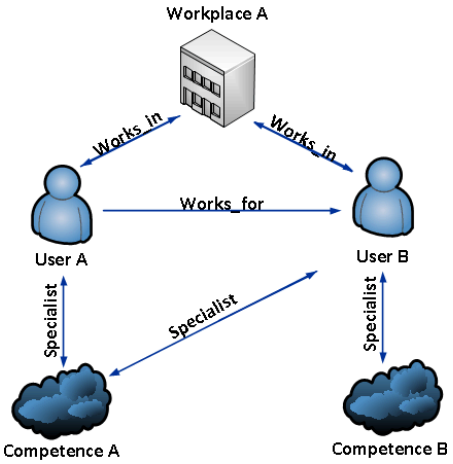

Fig. 5. Model of relationships

C. *Graph data provisioning - Web Application*

A dedicated web application was developed for end-user graph data provisioning and managing. This application allows for the creation of personalized accounts and edition and deleting of user data. Using web forms [Fig. 6] it is possible to set or modify data necessary to create or enlarge an user account. An interesting function is the definition of the location of workplaces/meeting places, which allows the implementation of messaging services based on user location. What is more, the system and proposed data model is extensible and its functionality can evolve depending on user needs.

Fig. 6. Web form interface

D. *Data Flow*

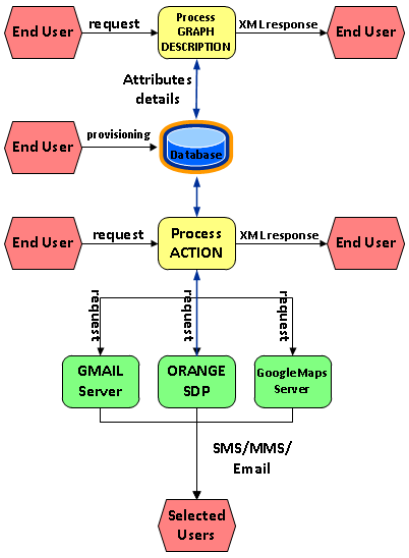

Fig. 7. Data flow in system

The model presented in fig. 7 shows the system data flow used to realize all implemented API messaging methods.

E. *Graph description API*

The Graph description API module exposes in RESTlike Web Service form, information about all the metadata (nodes attributes and communication functions) exposed by the system. The end GBM Application user (mainly the application used GBM API) can request, by using an URL, all actual graph attributes and methods that are possible to perform. The Graph description API return data in the presented below XML format.

```
<GraphDescriptionAPI>
<functions>
        <function>
                <functionname>SendSMS-
        competence</functionname>
                <link>
                http://80.48.104.236:8093/graph_api/SMS?
        competence=%&msg=
                </link>
                <description>
                Sending SMS messages to users with
        specified competencies
                </description>
        </function>
        . . .
        </function>
</functions>
<attributes>
        <competence>
                <value>DEV</value>
                <description>Developer</description>
        </competence>
        . . .
        <location>
                <value>BARBARY</value>
                <description>ul.Św.Barbary2</description>
        </location>
        . . .
</attributes>
</GraphDescriptionAPI>
```

In the presented above structure, the following functions and attributes described in Table 2 are defined as XML elements.

Table 2. Graph description API XML elements

XML element	Description
GraphDescription API	XML root element
functions	List of communication functions exposed by Graph
Function	communication function
functionname	Name of communication function e.g send SSMS or USSD
Link	URL for the server exposed communication function
description	Function description
attributes	List of graph attributes which can be used as parameters values in Graph communication functions
competence	Attribute name in this case – competence (skill)
Value	Attribute value (e.g. BD – Big Data)
description	Attribute value description (e.g. – Big Data)

The existing and exposed in graph database functions and attributes of the API presented in this chapter are similar to the concept of the Web Services Description Language (WSDL) for SOA or WADL (Web Application Description Language) proposed for RESTful Web services, but unlike them functions and attributes are dynamically changed with the change of contents of the database (for example, after adding new competences - employees skills) - they appear in the Graph description API as new attributes.

F. *The Graph communication API methods*

The Graph based messaging offer three communication functions: Send SMS, Send USSD and send e-mail as an easy RESTlike methods, which can be used in external applications for invoking communication between people which data contains an implemented graph database.

Sending SMS or USSD messages can be executed by calling an URL with the following syntax:
http://80.48.104.236:8093/graph_api/SMS?competence=value&msg=value
where:
competence - is the list of people skills (described in chapter 4)
message – is a text message sent to a group of people e.g:
http://80.48.104.236:8093/graph_api/SMS?competence=BD,SDP,LOC,STAT&location=OBRZEZNA&msg=Hello

An application server parses parameters and checks graph database content. A communication action (sending SMS or USSD messages) is processed by SDP (via Orange API) to the set of selected users. After a successful sending process, the API return a XML response with summary information about the number of informed people shown below:

```
<objects type=”array”>
   <object>
      <status>Success</status>
      <method>SMS</method>
      <ammout>5</ ammout>
      <message>Hello</ message >
   </object>
</objects>
```

The email sending process is similar to the SMS and USSD messages. The send email action has its own JRuby controller method which is accessible by the URL bellow.

http://80.48.104.236:8093/graph_api/Email?competence=value&location=value&message=value
where:
competence - is a list people skills (described in chapter 4)
message – is a text message sent to a group of people
location – a list of predefined location where An application will search for people who are located in a particular place
e.g:
http://80.48.104.236:8093/graph_api/Email?competence=ALL&location=OBRZEZNA&message

Calling this method allows the user application to send with the SMTP protocol an email messages to the list of selected people based on employees skills.

Successfully delivered messages are summarized in a XML response shown bellow.

```
<objects type=”array”>
   <object>
      <status>Success</status>
      <method>Email</method>
      <ammout>3</ ammout>
      <message>Hello</ message >
   </object>
</objects>
```

An additional special email API (DANGER e-mail type)

http://80.48.104.236:8093/graph_api/Email?type=DANGER

informs about a danger situation of the person which called the API function (all information about the person who invoked the API is collected from the username by using the HTTP Basic authentication method). In this special case, the graph application sends to the all the people in the nearest location a special e-mail containing a GoogleMap picture with a marker pointing to the location of the person who send requested as shown in Fig.8.

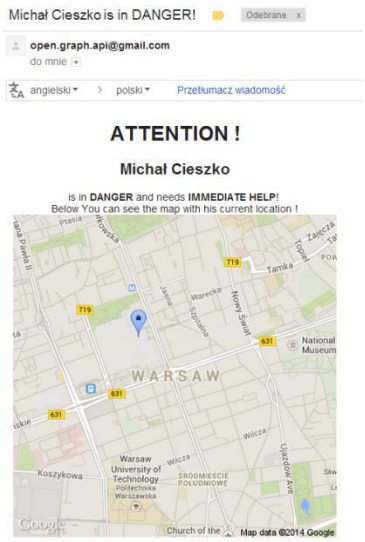

Fig. 8. An example DANGER type e-mail message

The presented above special type of GBM location API can be very useful e.g. in case a health problem with the person sending the request when immediate help is mandatory. In this case the API might be invoked by a mobile application using build in hardware like accelerometer, thermometer etc.

G. *End User applications*

The Graph Based Messaging concept includes functionality dedicated to wide range of applications and services. From the security point of view, API (using Base authentication method) invocations are processed only for correctly authenticated API users. The API was developed in a RESTlike architecture style. For an API exposed by the Graph Based Messaging system, the authors have developed a few example applications. The first one is a Windows application which offers the sending of SMS, USSD or e-mail messages to the users having defined skills or located in dedicated places.

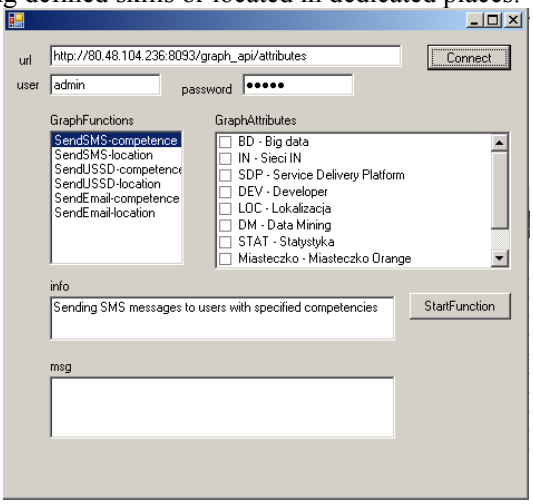

Fig. 9. Windows Client Application

After start, the application end user can choose any of the available *GraphFunctions* and then select some of the allowed *GraphAttributes*. The *GraphFunction* description appears inside the *info* section. Action (sending of messages) can be invoked using the *StartFunction* button which makes a HTTP request and invoke the selected Graph communication API method. A response received from the API is presented in the *info* field as is shown in Fig.11.

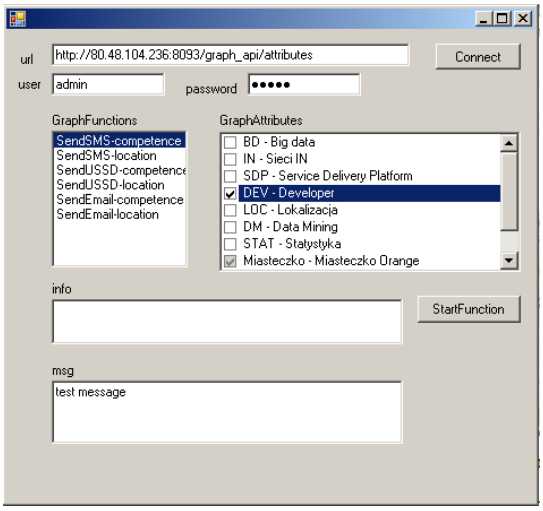

Fig. 10 Windows Client Application – an example function call

The second end user interface is dedicated web application. This subsystem is based on a web graphical user interface and offers additional features which makes it easier to invoke some actions (e.g. sending SMS/USSD/Email), check the system status or visualize a mobile terminal's (and themselves phone user) locating accuracy [Fig.11].

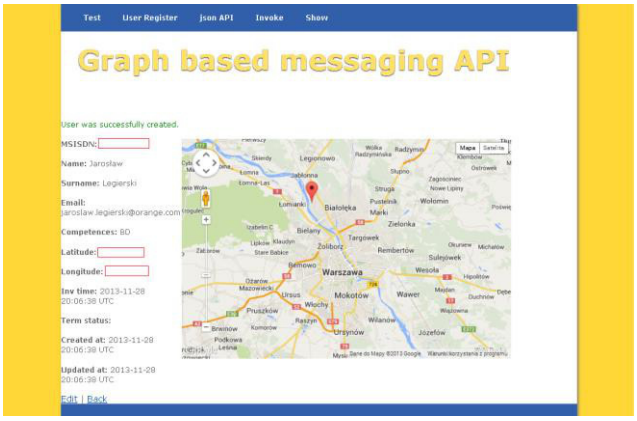

Fig. 11. End user Web GUI

Values of competence or predefined location parameters are store in database and represent the set of users which are related to requested skills and locations.

VI. FUTURE WORK

In the next system version the authors want to integrate

the Graph Based Messaging system with Facebook Graph API as a source of information about users, friends and other attributes, which can be very useful e.g. in e-mail messaging in case of emergency (DANGER e-mail API described in chapter IV F). The next functionality is a provisioning method, which can import and create a graph structure based on data from a text file provided by business clients. In future system versions there are also planned to be implement personal features (as stored in graph additional attributes e.g. users calendars), to extend the area of application usage.

VII. Summary

The Graph Based Messaging system is a prototype which presents the potential area of usage the traditional telecommunication channels connected with social networks technologies such as graph databases. The described communication oriented functionality in the API form allow to solving of many problems:

The system isolates user data (such as MSISDN number, e-mail) from the API requester and themselves thus addressing legal problems (sensitive data are not exposed). All data is processed on the server side, client applications are light and not generate client processor load from a computational point of view.

Client applications are not complicated from a mathematical point of view, graph oriented algorithms (clustering, shortest path) are implemented on the server side and the end result of them is available via the API.

The large advantage of the presented solution is the implementation of Neo4J – dedicated graph database. Relational databases generate many constraints. Programmers have found it difficult to append new objects and relationships to existing solutions while keeping system stability at the same time. The Graph based messaging system is a solution providing good performance while keeping all user data in full secret. What is more, the end user (API client) does not need to know the dataflow inside the system and details about implemented algorithms. The Author's aim was to provide the concept of the solution based on graph data structure with a wide range of functions and easy to implement in different business usage scenarios.

The prototype of the Graph Based Messaging System has been developed under the Orange Labs Open Middleware 2.0 Community program [35].

References

[1] Portal Facebook https://www.facebook.com/ [28.11.2013]
[2] Portal Google + https://plus.google.com/ [28.11.2013]
[3] Portal Twitter https://twitter.com/ [28.11.2013]
[4] Portal Instagam http://instagram.com/ [28.11.2013]
[5] Portal LinkedIn www.linkedin.com [28.11.2013]
[6] Portal Viadeo http://pl.viadeo.com/en/ [28.11.2013]
[7] The Graph API , Facebook https://developers.facebook.com/docs/graph-api/ [27.11.2013]
[8] Freebase API https://developers.google.com/freebase/ [28.11.2013]
[9] Search Cookbook https://developers.google.com/freebase/v1/search-cookbook [28.11.2013]
[10] GraphEdge API http://www.programmableweb.com/api/graphedge
[11] Waldron Faulkner GraphEdge Blog http://waldronfaulkner.wordpress.com/ [28.11.2013]
[12] GraphEdge professional http://agency.graphedge.com/index.php [28.11.2013]
[13] GraphMuse API http://graphmuse.com/api#introduction [28.11.2013]
[14] CicerOOs Semantic Graph API http://www.ciceroos.it/api-documentation [28.11.2013]
[15] Health Graph API http://developer.runkeeper.com/healthgraph [28.11.2013]
[16] Viadeo Graph API http://dev.viadeo.com/documentation/ [28.11.2013]
[17] Portal Meetbymaps.com http://www.meetbymaps.com/en [17.01.2014]
[18] Portal WishMindr http://www.wishmindr.com [17.01.2014]
[19] Portal oGoWoo www.ogowoo.com [17.01.2014]
[20] Portal Decoda www.decoda.com [17.01.2014]
[21] Podziewski, A.; Litwiniuk, K.; Legierski, J., "Emergency button — A Telco 2.0 application in the e-health environment," Computer Science and Information Systems (FedCSIS), 2012 Federated Conference on , vol., no., pp.663,677, 9-12 Sept. 2012
[22] Wawrzyniak, P.; Korbel, P.; Borowska-Terka, A., "Student information delivery platform using telecommunications open middleware APIs," Computer Science and Information Systems (FedCSIS), 2013 Federated Conference on , vol., no., pp.871,874, 8-11 Sept. 2013
[23] Korbel, P.; Wawrzyniak, P.; Grabowski, S.; Krasinska, D., "LocFusion API - Programming interface for accurate multi-source mobile terminal positioning," Computer Science and Information Systems (FedCSIS), 2013 Federated Conference on , vol., no., pp.819,823, 8-11 Sept. 2013
[24] Korbel, P.; Skulimowski, P.; Wasilewski, P.; Wawrzyniak, P., "Mobile applications aiding the visually impaired in travelling with public transport," Computer Science and Information Systems (FedCSIS), 2013 Federated Conference on , vol., no., pp.825,828, 8-11 Sept. 2013
[25] GSMA OneAPI http://www.gsma.com/oneapi/ [17.01.2014]
[26] L. Richardson, Sam Ruby, David Heinemeier Hansson, RESTful Web Services, O'Reilly, 2007
[27] Open Service Access (OSA); Parlay X web services, 3 GPP, http://www.3gpp.org/ftp/Specs/html-info/29199-01.htm, [17.01.2014]
[28] SOAP specification W3C, http://www.w3.org/TR/soap [17.01.2013]
[29] "LifeWorks Die Kommunikation der Zukunft beginnt", Information and Communication Networks A50001-N1-S20-1 bySiemensAG2004
[30] "Siemens ICN Announces LifeWorks Concept for Integrating Personal, Enterprise and Carrier Technologies", http://www.prnewswire.com/news-releases-test/siemens-icn-announces-lifeworks-concept-for-integrating-personal-enterprise-and-carrier-technologies-71170057.html
[31] D.Bogusz, J.Legierski, A. Podziewski, K. Litwiniuk, „Telco 2.0 for UC – an example of integration telecommunications service provider's SDP with enterprise UC system", Wrocław 2012.
[32] A.C, Weaver, B. B. Morrison, Social Networking, Computer, Volume: 41 , Issue: 2, 2008
[33] JRuby portal http://jruby.org/ [24.03.2014]
[34] Ian Robinson, Jim Webber, Emil Eifrem, Graph Databases, O'Reilly Media, 2013
[35] Open Middleware 2.0 Community Program portal www.openmiddleware.pl [30.03.2014]