A Telco Enabled Social Networking and Knowledge Sharing.

Manfred Schneps-Schneppe, Dmitry Namiot, Andrey Ustinov

***Abstract*— Our main goal is to change the paradigm relating to the voice data usage in web applications. The idea is seamlessly integrate telecom data (voice) into web applications for knowledge sharing. And the principles we are describing here could be used in the education process. Technically we merge Telecom and Web programming, adding a new software component (proxy) to open source IP-PBX Asterisk for content providers. The document is exampled by mashup service Gazetta 2.0 for mobile phones and geo tagging for fixed line phones (audio-records collected right from phones being mapped on the Google Maps).**

***Keywords*— Open Source, telephone exchange, Web 2.0, mashup, Google Maps API, geo tagging.**

I. Introduction

Knowledge sharing enables people in virtual communities to access relevant knowledge from broader scope of resources. The performance in such environments is fundamentally based on how effectively the knowledge can be shared across people, and how efficiently the created knowledge can be organized and disseminated to enrich digital content.

A social network service uses software to build online social networks for communities of people who share interests and activities or who are interested in exploring the interests and activities of others. Most services are primarily web based and provide a collection of various ways for users to interact, such as chat, messaging, email, video, voice chat, file sharing, blogging, discussion groups, and so on. The current trend for content management dictates the growing role of multimedia data. Another obvious tendency is integration. Customers (users) should be able do deploy all the available channels for content distribution and delivery.

Our main goal is to change the paradigm relating to the voice data in web applications. We use Web 2.0 technology, namely, support user-generated content by well known techniques: Ajax-based rich Internet application techniques, semantically valid XHTML and HTML markup, ontology (in the form of tags), REST (Representational State Transfer) Web APIs, mashups merging content from different sources, etc. In telecommunication applications, Web 2.0 technology corresponds to Telco 2.0 and Mobile 2.0 [1] containing the essence of online social networks. To attract user-generated content we need open programming interfaces. And it is what we do from the very foundation of AbavaNet company [2].

The idea is seamlessly integrate telecom data (voice) into web applications. Such achievement seriously explodes the possibility for social services. Just one simply explanation: all the mobile phones become the main devices for posting/getting data from new services. We add new software applications to Asterisk using our application server Abava Gateway (Fig. 1). Asterisk provides a central switching core, with four APIs for modular loading of telephony applications, hardware interfaces, file format handling and codecs. It allows here transparent switching between all supported interfaces, allowing it to tie together a diverse mixture of telephony systems into a single switching network. And what is very important in the context – Asterisk is free open source software. Until now Asterisk is used (reviewed) mostly in the traditional telecom sense – as a free replacement for the traditional PBX. But by our opinion, it could be (and should be) used mostly as a software service, integrates voice data into modern Web 2.0 applications.

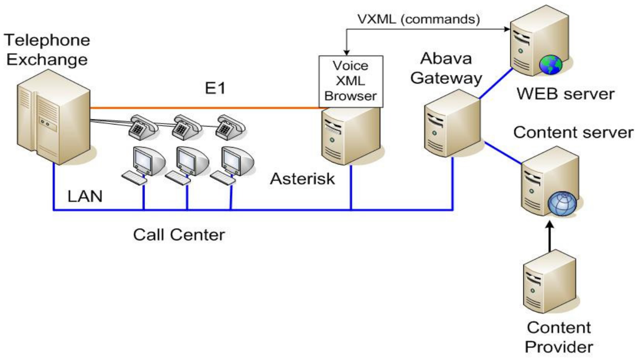

Figure 1. Application server Abava Gateway

Abava Gateway plays a role of middleware between real-time telco world including IP-PBX Asterisk and call center functionality, from one side, and internet world with content providers based on web services, from another.

Our own experience and extensive market study shows the two groups of applications around the open source PBX: completed applications (e.g. call centers, messengers etc.) and embedded solutions on the base of PBX. Nobody

Manuscript received Jul 29, 2013.

M. Schneps-Schneppe is with the Ventspils International Radio Astronomy Centre, Ventspils University College, Ventspils, Latvia (e-mail: manfreds.sneps@gmail.com).

D. Namiot is with the Faculty of Computational Mathematics and Cybernetics Lomonosov Moscow State University, Moscow, Russia (email: dnamiot@gmail.com).

A. Ustinov completed master's research project in Open Information Technologies Lab, Faculty of Computational Mathematics and Cybernetics Lomonosov Moscow State University, Moscow, Russia.

actually targets web developers as potential customers for telecom platform. But most the applications developed nowadays are actually web applications. And corporate development anyway is unable to create all the potentially demanded applications. So the key issue for adding telco to existing and future social networks is to satisfy web development crowd with simply tools, let them deploy telecom enabled applications without any expertise in telecom world.

The main area of interest during our professional activity has been related to telecom applications following Parlay/OSA concept [3]. These applications are carried on in cooperation with Ericsson, Iskratel and other operators and vendors. The following are several Mobile 2.0 services currently available by means of Russian operator AudioTele (Figure 2):

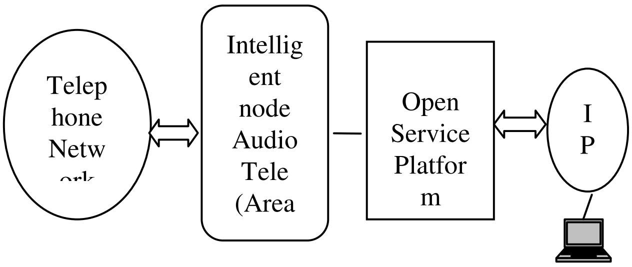

Figure 2. Implementation of Open Service Platform Abava for user generated content delivery (in the framework of the above listed services and Intelligent node AudioTele).

1) Voice mail. Access will be provided via IVR interface and via the web. Voice mail server will provide also an open API for access to voice mail from third party web sites.

2) Automation voice info. User-defined voice fragments.

3) Voice Recorder. Lets you record and publish voice messages.

4) HTTP Gate. Lets you post information about incoming calls to web applications via HTTP protocol.

5) Pay call. This service provides navigation through web site confirmed by the call. The idea is to allow site navigation only after the call to some premium call number. It could be integrated with any web-site via Ajax interface. Let us see some examples.

a) User authentication on a certain site. Along with the name / password, we ask the user to pay for their work by the premium call. It could be implemented as one time password returned by the call service (text-to-speech).

b) Paid publication. For example, at a portal with some user generated content, the publication of advertisement is possible only after the payment. Payment is made through a call to the specified number.

c) Pay wall. We feature free part of the ad / article / response, and so on, and completely content becomes available only after the payment.

In all cases, the payment is a call to the premium number allocated by the operator. Upon connection, the caller reported some code (a sequence of digits) to be entered in the HTML form. Next, the code entered by the user can be verified.

6) Voice SMS. This service combines voice messages and SMS

7) Network microphone. It lets you record voice messages and automatically publish them on the public web site.

8) Voice blog. It lets you publish voice messages to blogs, using the public API's provided by the blog platforms.

II. Telecom Mashups

Let us provide two concrete examples of the applications of the work proposed:

1) Gazetta 2.0 - access to voice mail right into concrete web page,

2) Geo tagging - audio-records collected right from phones being mapped on the Google Maps. The newspaper published service phone number right in the article (E.g.: "Listen the additional voice data by calling XXXX"). A reader calls this number and listen the voice data. Currently the podcast for leading Russian business paper could be reached by this way.)

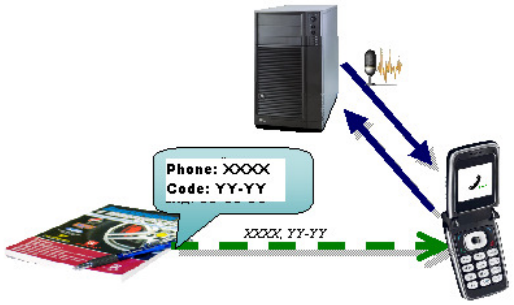

Figure 3. Telecom Gatezza

Geo tagging for fixed line phones. The following describes a mashup service for fixed line phones – audio-records collected right from phones being mapped on the Google Maps [4].

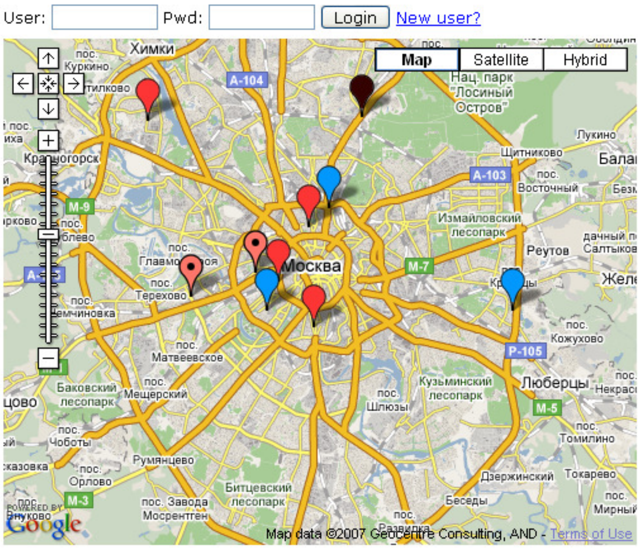

Figure 4. Map with user-generated voice messages

The idea is very simple: for the fixed line phones we know the location of the phone. So voice messages recorded by phone could be geo tagged (connected to the map) for the future access either from Internet. Internet users can collect messages for the selected area or download automatically created podcast for that area. Phone users can call and get voice messages by the geo proximity. Phone users can also reply to the messages without discovering the original (author's) phone. So with this application we can provide Web 2.0 (or Telco 2.0) service (actually – the class of services) that mixes Internet and telephony.

The fixed line phones can actually have one advantage over mobiles (at least from the point of view services) - the location of customer is known. The features that are getting available in the mobile world via Location Based services and API's could be deployed in the fixed line world much more easily. The service (actually as we will describe below – the whole class of services) deploys the very simple idea – as soon as we getting call we know (at that moment) where to find the originating party of this call. And "where" here means the location in geo sense - we can simply get latitude/longitude for the originating phone. So the next step was almost obvious: let us record messages by the phone and map them on the map, using Google Maps API. So we will get user generated voice content (simply mp3 files in our case) attached to the map. See Figure 4 contained user generated voice messages. Marker type and color are indicating the various topics for the messages.

What can we get from such a service: a) internet users can now pickup messages from telco users right from the map (get all the messages for the selected area), b) telco users can call a service number and simply listen the messages, selected by the geo proximity (e.g.: listen all the messages within the 10 km area, etc.). So how does it work? There is a service number mobile users can call. A special Voice XML based application on that number simply records the message. Messages will be stored as MP3 files. Now the service application will request address information for the originating phone.

Here are two options for address information: 1) a public service that returns address information by the phone, 2) some simplifying versions of that service. In both cases after getting the address we can use geo-coding service from Google Maps API (actually we've used in some services Yahoo geo coding too). Geo coding service accepts address information and returns latitude/longitude info for the given address. And <latitude, longitude> pair could be used in Google Maps API for drawing the marker on the map. That is all actually. All the rest is absolutely transparent.

Internet users can see the Google map mashup. So they can narrow the area, listen the records from that area (build-in flash based player right on the site), download the records from the selected area as a podcast (RSS feed will be generated by the service) or even embed the map (selected area from the map) into own web site/blog. Phone users can call a service number and listen back all the records closest to their location. "Closest" here means right the distance calculation on the pap. Also, the telco users can respond to the messages right from the player, so they do not need to know the originating number (and the originating number is not discovered here).

III. ASTERISK AS A SERVICE DEVELOPMENT PLATFORM

Our idea is to bring web developers into PBX programming world. As seems to us it is more than ambitious goal. Web development and Telecom development exist in parallel. For reaching this goal we deploy a set of new Java based tools for open source IP-PBX Asterisk. Using Java let anyone the ability to extend our own tools in the future. But again our primary goal is not add new Java based tools. The idea is to completely change the way telecom services are developed. We are intending to bring telecom stuff right into web development world via scripting languages, typical for web development and via pre-build set of widgets for the mail telco-related tasks.

Until now Asterisk is used (reviewed) mostly in the traditional telecom sense – as a free replacement for the traditional PBX. But by our opinion it could be (and should be) used mostly as a software service integrates voice data into modern Web 2.0 applications. We are planning to introduce new software for service development as well as pre-build services for Asterisk. Service development tools will allow new development for non-telco oriented staff. We are planning to convert telecom development into web development. Telecom domain tasks will be converted right into Web domain tasks.

Our study shows that this area is relatively empty at this moment. And of course the question step after the standard voice connection is how to program services for PBX. So our extensive background in the J2EE standards as well as in the Open telecommunication protocols lets us suggest a simple and useful approach for the developing applied services on the Asterisk. That project includes the following tasks:

- introduce Java libraries for developing Asterisk scripts right in JSP (web scripting technologies)
- add custom JSP tags for Asterisk based development
- prepare manuals and examples for access to Asterisk platform from various popular web frameworks
- prepare widgets for the main telecommunication tasks
- implement Voice XML and CCXML over Asterisk scripts particularly for personalized IVR with semantic Web features

XML based languages are completely independent of programming systems. So it is a way we will bring PBX development into web world.

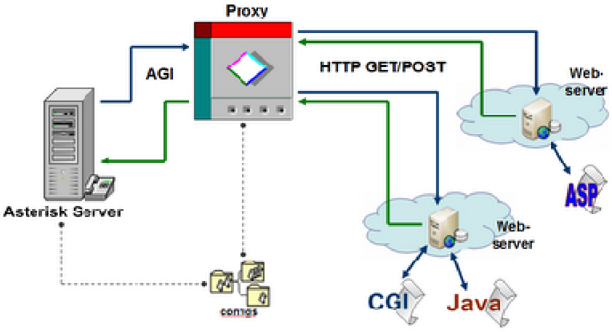

Figure 5. Asterisk Platform

We propose to integrate a new component (proxy) into the Asterisk platform. The main functionality of the proxy is to translate telecommunication calls into HTTP requests to external web services. Telecommunication services are located separately from the PBX, while the information they receive from Asterisk is presented as a HTTP-request. Technically, HTTP GET/POST request is a request, in which external telecommunications service passed information about the subscriber's name - *CallerIdName*, caller's number - *CallerIdNumber* and and called number - *Extension*. Upon receiving necessary parameters, such as (calling/called number) a web service produces and forwards its instructions to the proxy. The latter receives and translates them into Asterisk instructions. The development of such services under the architecture described above is similar to a conventional CGI-script, for which there is a plenty of programming tools. As a result a programmer doesn't need to be familiar with the Asterisk API.

As seen the AGI-proxy component is at the heart of the model (Fig 5). The AGI-proxy is a Java-based application implemented on the basis of FastAGI, an open source library. The AGI-proxy is installed directly on the PBX Asterisk side [5].

In fact, the Asterisk represents the same thing for the AGI-proxy, as the J2EE container does for a Java-servlet. All calling massages produced by the Asterisk come to the proxy within the method service with two parameters: interfaces *AgiRequest* and *AgiChannel*. Through the first parameter one can get the information about the calls (caller name / number, called number, context of the call, feed settings, etc.) and through the second one the interaction with the Asterisk is carried out (call termination, transfer mode and etc.). Subscriber's name (*CallerIdName*), caller's number (*CallerIdNumber*) and called number (Extension) are wrapped within a string-parameter by the AGI-proxy. Then the AGI-proxy executes an HTTP request on its behalf to an external web service, whose URL is set in the configuration. The response from the invoked service is seen as an indication to what to do with the call. The call may be terminated or redirected to a specified number; a media file may be played. The instructions returned by an external service must be presented in an xml-format. On receiving this xml-response, the proxy will play the media file "demo-nogo" (*playMediaFile*).

Under this architecture the sequence of calls is reduced to a very simple diagram (Figure 6).

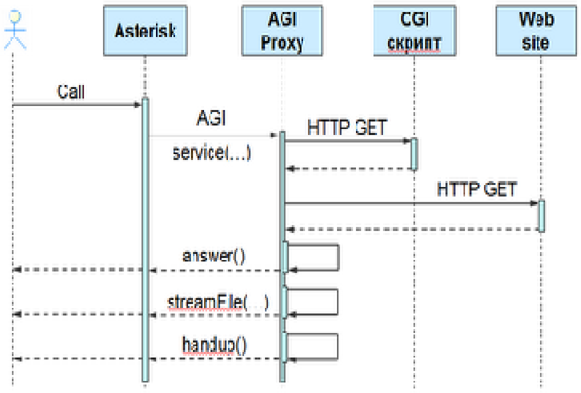

Figure 6. Call Sequence

Web services within this model may have many different purposes - from the statistics (to record a call and terminate it - in voting systems) to the Asterisk dial plan control (auto-answering and call forwarding in communication systems). The above mentioned schema for Asterisk connection was actually widely used across several products [6, 7, 8].

IV. Asterisk as a platform for the education

The idea of the proposed Asterisk proxy converts the task of the telecom services development into web programming. It seriously lowers the entry level for programming skills needed for the telecom development. This feature could be successfully used in the educational process. Must more students nowadays are familiar with the web programming, writing CGI scripts etc. So this proxy was successfully used in the university processes. Also the hosting for the telecom service in the form of CGI script could be very cheap. All do we need now is just an ordinary web hosting with CGI support (e.g. with PHP or Perl).

References

[1] Schneps-Schneppe, M., & Namiot, D. (2008). Telco Enabled Social Networking: Russian Experience. In BALTIC CONFERENCE (p. 33).

[2] AbavaNet http://abavanet.ru Retrieved: Jul, 2013

[3] D. Namiot and D.Gepech (2006). Open Interfaces Parlay X in NGN nodes. Vestnik svyazi, (1), 34-37. (in Russian)

[4] M. Schneps-Schneppe and D.Namiot (2008). Telco enabled social networking exampled by GEO tagging. APPLIED INFORMATION AND COMMUNICATION TECHNOLOGIES, 50.

[5] Asterisk Web Gate code: http://code.google.com/p/asterisk-web-gate/ Retrieved: Jul, 2013

[6] Schneps-Schneppe, M., Maximenko, A., Namiot, D., & Malov, D. (2012, October). Wired Smart Home: energy metering, security, and emergency issues. In Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT), 2012 4th International Congress on (pp. 405-410). IEEE. DOI: 10.1109/ICUMT.2012.6459700

[7] Schneps-Schneppe, M., & Namiot, D. (2013). About Home Gateway Mashups. International Journal of Open Information Technologies, 1(5), 1-5.

[8] Y. Daradkeh, D. Namiot, and M. Sneps-Sneppe. Telecom mashups: a practical example. International Journal of Engineering Research and Applications (IJERA) ISSN: 2248-9622, 2(2):710–714, 2012.

[9] Namiot D., Schneps-Schneppe M. Smart Cities Software from the developer's point of view. International Conference on Applied Information and Communication Technologies (AICT2013), 25.-26. April, 2013, Jelgava, Latvia, pp. 238-245