Augmenting Context Awareness by Combining Body Sensor Networks and Social Networks

Md. Abdur Rahman, *Member IEEE*, Abdulmotaleb El Saddik, *Fellow IEEE and* Wail Gueaieb, *Senior Member IEEE*

Abstract — **Due to recent advancements in social networks, many people can consume diversified services on a daily basis and have developed an association with different community of interest via these services. However, a person only accesses a sub-set of these services at a given time either to consume certain services or to share information with a community of interest (COI). This paper tries to answer two important research questions: "how to dynamically capture user context from heterogeneous sources?" and "which services and COI are related to any given context?". To address these two challenges, we propose a framework called SenseFace, which provides user context from two sources: body sensor network (BSN) and multimedia information contained within social network space. We present the detailed design and implementation of the framework and share our preliminary test results.**

***Index Terms* — social factors, intelligent sensors, multisensory systems, telemetry.**

I. Introduction

Social networks have become popular in terms of information consumption, sharing and communication among groups of similar interest. Moreover, the popularity of social networks has been proliferated due to several technologies offered by Web2.0 and modern smart phones. With the help of these technologies creating, managing, sharing and accessing information is now ubiquitous. As a result, a user can be in touch with his/her community of interest via different social networks and consume diversified Internet-based services such as weather, calendar, medical, professional, sports, trading, consumer, educational etc. any-time and anywhere [1].

Capturing user context is an attractive area of research and numerous techniques have been studied in the past [2], [3], [4], [5]. For example, context has been studied in areas such as personal context data (user profile, user location, time of system access, to-do list, types of end-device used, and contact list), social context data (social ties through social networks and interactions, types of information shared), and application based context data (types of web services used, bandwidth and reliability requirements of each service, types of protocols and access mechanism needed for each service, URL of each service) [1]. Several researches advocate that user context can be captured using both BSN and multimedia information contained within various services ubiquitously [4], [6], [11], [17]. A BSN is a wireless network of sensor nodes that can capture physical and ambient phenomena without any human intervention, can work in an ad-hoc nature, and can share the sensory data with other networks such as ad-hoc networks, Internet and cellular network. Hence, sensors deployed in a human body or one's surrounding environment can provide a rich source of event and ambient information [17]. In a similar fashion, a person's nearby friends and family members' real-time geographic location and activities can be tracked from different social network applications[1]. Thus, analyzing information contained in one's personal social network provides very useful contextual information [7].

Context-awareness deals with the adaptation of computing systems to the user's current context. A framework can be called Context-Aware if: "*it uses context to provide relevant information and/or services to the user, where relevancy depends on the user's task*." [27]. Hence, a context-aware social network framework supports recommendation of a subset of social ties, often called community of interest (COI) and services to a user, automatic execution of a service for a user and tagging of context-to-information to support later retrieval. By combining and inferring on this sensed information, location such as "in a research meeting" or "in a doctor's office," or an activity like "driving a car" and a physiological condition such as "high blood pressure" can be deduced.

Md. Abdur Rahman is researcher at the Multimedia Communications Research Laboratory, School of Information Technology and Engineering, University of Ottawa, 800 King Edward Ave, Ottawa, Ontario, K1N 6N5, Canada (email: rahman@mcrlab.uottawa.ca).

Abdulmotaleb El Saddik is the director of the Multimedia Communications Research Laboratory, School of Information Technology and Engineering, University of Ottawa, 800 King Edward Ave, Ottawa, Ontario, K1N 6N5, Canada (email: abed@mcrlab.uottawa.ca).

Wail Gueaieb is the director of Machine Intelligence, Robotics, and Mechatronics (MIRAM) Laboratory, School of Information Technology and Engineering, University of Ottawa, 800 King Edward Ave, Ottawa, Ontario, K1N 6N5, Canada (email: wgueaieb@site.uottawa.ca).

[1] http://fireeagle.yahoo.net/

User context and relations between a group of co-users have been studied in terms of analyzing email, blogs, co-authored papers, and exchanged documents, where the authors present an algorithm to cluster users based on the content of social network information [8]. Authors in [7], [9] have elaborated how to capture use context in real-time using a smart phone paired with wearable sensors. For example, the author in [10] has developed a prototype BSN for mobile users, which can recognize up to 52 different physical activities by employing only three tri-axial accelerometers. Although similar other works exist in the literature, there have been few efforts to add both sensory data as well as social networks as sources of user context in social network domain.

In our earlier paper [11], we presented an open source framework, named SenseFace, which consists of four-tier network including mobile BSN, cellular network, Internet and an overlay network consisting of social networks, to pass sensory data from the mobile BSN to the overlay network and vice versa. The BSN assumes a smart phone as a personal gateway for upstream sensory data push to members of a COI in multiple formats. SenseFace also provides a wide range of options to share the sensory data coming from a BSN with members of one's COI. In this paper we propose several extensions. First, we elaborate the open stack architecture [12] by detailing the associated protocols and logics within each layer. The open stack is used primarily to find a total set of one's social ties and the services consumed from heterogeneous sources on the Internet. Second, we propose a ubiquity stack that can capture user context from the BSN and some diversified services on the Internet. Finally, we show how using both stacks we can dynamically find context-aware services and COI.

The rest of the paper is organized as follows. The framework design has been elaborated in detail in Section II with respect to two stacks, the open stack and the ubiquity stack. Section III describes the implementation details while section IV provides the framework evaluation results. Finally, section V concludes the paper and proposes some future possible work.

II. System Design

This section provides an in-depth design of the proposed framework in the light of different protocol stacks. We first propose a high level block diagram (see Figure 1) that best illustrates the framework. We assume that a subject carries a body sensor network, which comprises several wearable sensors and a smart phone having several types of built-in and external wearable sensors to monitor different physical and contextual information. The smart phone also acts as personal gateway by pushing sensory data from the BSN to remote users and receiving queries from authorized members of a given COI. The subject consumes diversified services from the Internet and interacts with COI through social network services such as Facebook, Twitter etc.

In order to seamlessly connect the physical BSN with the social networks and collect user context, the proposed framework utilizes two protocol stacks. The framework employs the open stack of Internet [12] to extract the social ties and consumed services of a subject from the Internet, which is called the personal social network (PSN). The ubiquity stack can collect user context in real-time from body sensor network, social network space and previous contextual profile stored in a database, details of which will be discussed later. Finally, the proposed ubiquity stack uses the PSN to extract a subset of relevant services and COI at any given user context and rendered to appropriate GUI. Next, we elaborate different entities of the framework in details.

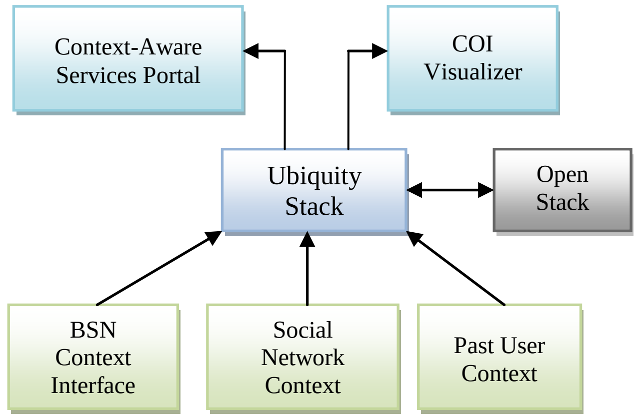


Fig. 1. High-level architecture of the proposed system.

A. *Open Stack*

The open stack facilitates handling online information of a user such as username, password, the personal and shared data, and social ties. In other words, it allows a user to maintain a common digital identity over diversified social networks and web services that one is connected with. Although sharing one's personal profile and extracting information about social ties as well as consumed services poses important privacy concern and requires incorporation of appropriate trust model, we left it out of scope in this paper. We are investigating proper privacy models such as [13] that can be integrated with the framework's future version. The open stack is shown in Figure 2. We now describe each layer of the stack in details.

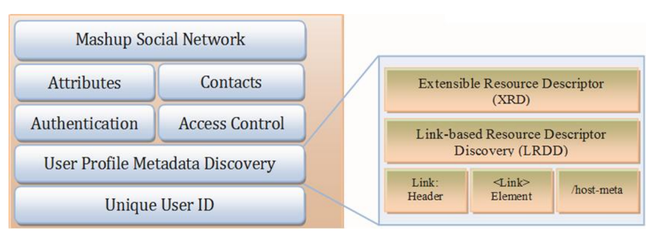


Fig. 2. Open stack to merge different social networks (courtesy: Marc Canter [12]).

Layer 1: Unique User ID

Unique user id is the piece of ID e.g. email address, user profile URI, cell phone number, or MAC address of a smart phone that is used by the layer above as a key to the user profile.

Layer 2: User profile metadata discovery

The user profile metadata discovery stack includes three sub-layers as shown in Figure 2. This layer takes a resource URI as input and provides the whereabouts (using middle and bottom layer) and format of access (top layer) as output. XRD[2] uses an XML schema to describe a resource e.g. the profile of a subject and its relationships to other linked resources or services e.g. address book, calendar, identity provider, type of supported authentication supported, and list of involved social networks. LRDD[3] is a resource discovery protocol that uses the bottom link framework to associate resources using existing protocols such as the HTTP Link header and HTML <LINK> element and /host-meta[4].

Layer 3-A: Access Control

Access control allows a COI to grant access to their private data, such as address book, from one site e.g. the service provider to another site e.g. consumer site without revealing their identity. Popular access control mechanisms for the open stack generally include OAuth[5], which offers two different access control formats: two legged and three legged, where each leg refers to different party involved.

Layer 3-B: Authentication

Authentication layer allows signing in with one's existing social network credentials such as Yahoo, Google, Facebook, Twitter, or any OpenID[6] provider to consume a service. OpenID provides a unique URI, which points to a subject's profile. It allows people to use single login information over multiple web services.

Layer 4-A: Attributes

Using this layer, two sites can exchange attributes of a particular subject such as full name, picture, birthday, country, gender, email, nickname, height etc.

Layer 4-B: Contacts

This layer works as a transport layer by allowing a web service to organize and publish information such as email, address books and lists of friends about every registered user's contacts. Typically existing user information is being kept in various proprietary formats (e.g. Google's GData Contacts API, Yahoo's Address Book API etc.) that are not portable in various social network platforms. Portable contacts provide a common protocol to share address book and other contacts data.

Layer 5: Mashup of Social Networks

Social network services are a cloud of services based on WSDL, REST, XML/HTTP, or database objects. Mashup layer processes Mashable entities that are raw information containers with standard interfaces or URIs that can be invoked to consume data. They can be Google calendar REST interface, XML interface of ECG heart beat sensor, or JSON-RPC interface of Yahoo weather API, to name a few. This layer encapsulates the logic of data processing and manipulation actions including joining, merging, sorting, filtering, constructing, transforming, clipping, and so forth. At the end of Mashup, this layer creates entities that can be shared with COI exposed as a service interface such as in REST, RSS, WSDL services to be consumed on i.e. Wiki pages, blogs, websites, portal servers, emails, faxes, or called directly from a smart phone.

In order to facilitate Mashup process, researchers have proposed many Mashup models [14], [15]. Among others, the Actor, Activity and Asset (3A) interaction model is intended for designing and describing social and collaborative environments [16]. Every service and sensor in a BSN can be mapped to one of the entities of the 3A model as shown in Figure 3. An Actor is the owner of an Asset, which is connected with a member of COI through an Activity. An Asset represents the sharable information source i.e. sensory data, a message etc. Thus, context information can also be modeled as system Assets. An Activity is the formalization of a common objective to be achieved by the subject and his/her social ties. Activity can be among others, information related to acceleration or ECG sensor, or user location/activity tracking from his/her twitter profile page.

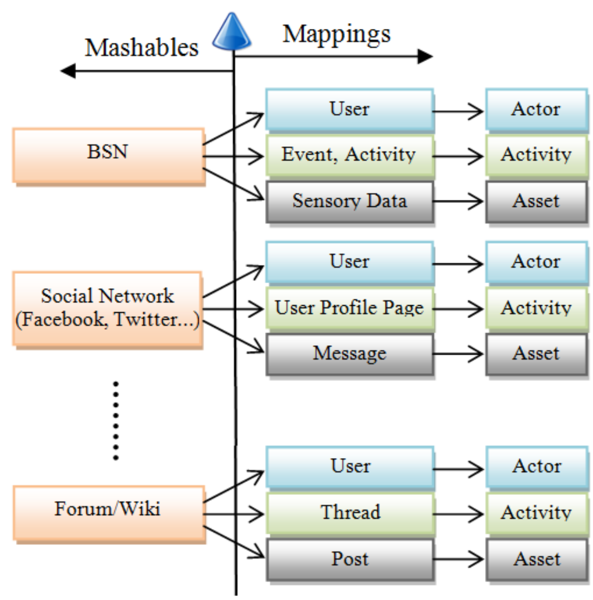


Fig. 3. Mapping individual Mashable entities into corresponding 3A model.

Leveraging the five layers of the open stack described above, a subject can extract information about his/her social ties and consumed services. In this literature, this total set of services and social ties of a subject is referred to as personal social network (PSN). In other words, open stack provides a means of extracting an up-to-date raw PSN database of a subject from the Internet, which is used by the ubiquity stack to provide context-aware services.

B. Ubiquity Stack for Context-Awareness

The role of this stack is to provide a ubiquitous overlay network by providing the logics, protocols and algorithms

[2] http://hueniverse.com/2009/03/the-discovery-protocol-stack/
[3] http://tools.ietf.org/html/draft-hammer-discovery-03
[4] http://tools.ietf.org/html/draft-hammer-hostmeta-05
[5] OAuth: http://oauth.net/
[6] OpenID: http://openid.net/

to extract user context in real-time and map a subset of services and COI with each context. This layer leverages the open stack to maintain a mirror dataset of one's PSN (see Figure 4). The stack facilitates in storing one's PSN in a smart phone or a protected web URI, to ensure user privacy and to facilitate user mobility. Since both the services and ties' information are available to this stack either locally or in a subject's personal web space, the overlay network can operate very fast and independent of the open stack. Also, this layer facilitates a user to specify higher layer concepts such ontologies, fuzzy logic etc. to define the COI and context. For example, definition of context, which service to use, and who should be the COI in each context is completely vague and varies from subject to subject. Next, we briefly describe each layer of this stack.

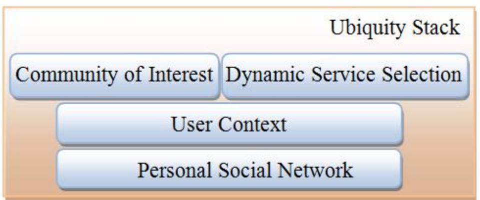


Fig. 4. Context-aware social network stack.

Layer 1: Personal Social Network

This layer plays an important role in the framework by maintain a downstream communication with the open stack. As discussed earlier, the PSN layer provides an updated mirror dataset obtained from the open stack that contains the raw metadata of one's association with all the services and corresponding social ties. For the optimum case, we assume that the PSN layer only connects to the open stack when there is an update in terms of services or social ties or a request from upper layer to look for any contextual information from the social networks e.g. new message in Twitter or events from, i.e. , Google calendar. This layer also plays a salient role for the upper layers by facilitating a way of categorizing the raw social ties of a user into high level categories such as family, health, study, business, and sports, to name a few. There are several benefits to such categorization. Assume a subject wants to define a context called 'alarming blood pressure' and map a COI containing a subset of family members, relatives, friends, family physician, nearby colleagues etc. from the available social tie dataset. In this case, the social ties might be decentralized in Facebook, Twitter, phone book etc. To facilitate the higher layer mapping process, a subject can employ existing social network analysis algorithms to categorize each social tie into a higher-level category as described earlier. For example, in our earlier work [17], we showed how a Markov process based social network analysis model called SIENA can process a subject's raw social network dataset as input and categorize each tie into one or more of the PSN categories.

Layer 2: User Context

This layer tries to capture a user context from three sources: real-time from BSN, social networks and past user contexts. Similar to the recommendations found in [1], [24], [25], [26], we take into consideration the following types of contexts during the modeling process:

- Personal context data including user profile, user location (home, school, gym, shopping mall, airport, mobile etc.), time of day or week (week day, week end, vacation etc.), user activities (walking, sleeping, lying, talking, running, driving etc.), user physiological information (heart beat, blood pressure etc.), to-do list, and types of end-device used, to name a few.
- Social context data including contact list, social ties through social networks and interactions, and types of information shared.
- Event based context such as appointments and meetings.
- Application based context data such as types of web services used, bandwidth and reliability requirements of each service, types of protocols and access mechanism needed for each service, and URI of each service.
- Historic context i.e. a subject's past context information stored in a database similar to user profile or resource profile. For example, Twitter offers a rich source of user context in terms of current and past activities. Another example is last 10 minutes physiological or one's ambient data stored in a smart phone.
- Intra-user context difference, which is a result of change in one particular user's context throughout a day. For example, every user needs to access different services or communicate with different categories of people during different period of a day.

Tiny wearable sensors and sensors that come built-in with modern smart phones facilitate the collection of some personal and intra-user context. In addition, we can collect intra-user context such as location from social networks e.g. a person's location via Yahoo FireEagle[7], announcing current location to family members while traveling via Lufthansa air lines, broadcasting one's current location via Automatic Packet Reporting System (APRS[8]) network where there is no Internet or cellular network, to name a few.

Layer 3A: Community of Interest

This layer semantically maps each social tie from the PSN category with a subject depending on user's context. In order to map the user context with PSN, we incorporate semantic knowledge. In our earlier work [17], we showed how Fuzzy Ontology could be used to model social relations depending on semantics, context, desire and intentions of a person, which is vague and might differ from user to user [18], [19], [20]. For example, in the case of video conferencing context, one might need to map a subject with the research members. The context layer uses a user's existing context profile or dynamically finds user context data from BSN or social networks to locate pertinent context such as temporal, event-based, spatial, or ad-hoc context. In the case of physiological contexts such as "alarming heart rate", a mapping is done among sensor profile (ECG sensor), resource profile (smart phone ECG handler application, ECG handler web services), user profiles (family physician, immediate caregiver family

[7] http://fireeagle.yahoo.net/
[8] http://www.aprs.org/

member, friend, colleagues, and emergency contacts), and context profile (physiological, event-based, spatial, temporal).

Layer 3B: Dynamic Service Selection

This layer is co-located with the community of interest layer and tries to answer the types of services a subject needs to invoke at any given time. This layer is responsible to call the actual service to be rendered to the current user interface e.g. a web portal. Depending on the context information provided by layer 2, a number of services might be involved. For example, assume a subject has access to three different audio/video conferencing tools such as SKYPE[9], ADOBE ConnectPro[10], and TANDBERG[11]. However, he/she has to use only one of them depending on the context i.e. depending on the type of community involved, types of end device available, types of mobility pattern, bandwidth available etc.

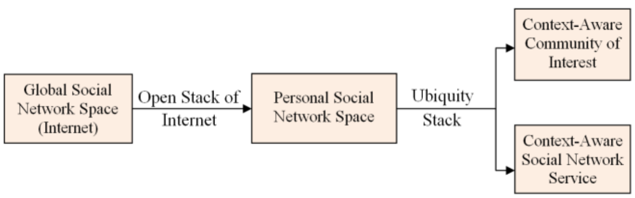


Fig. 5. Using the open stack of Internet and ubiquity stack to find social network services and COI based on user context.

Figure 5 shows a block diagram on how to utilize the above two stacks containing eight layers to find one's COI and necessary services ubiquitously. As shown in the figure, the framework uses open stack to filter a user's personal social network from the global social network space. With the help of Ubiquity stack, anyone can extract his/her PSN, carry ubiquitously through smart phone or store them online and subsequently use to find each service and involved COI at any given context.

Fig. 6. Sensors and smart phone used to evaluate the framework (a) An HTC dream running Android 2.0, (b) Garmin Forerunner heart beat sensor, (c) SHIMMER ECG kit, and (d) Built-in sensors and communication channels comes with the HTC smart phone.

III. Implementation

1) Body Sensor Network

Figure 6 shows the hardware we have used to realize the BSN with details about the built-in sensors and sensory data pushing interfaces that come with the Google phone. To extend the sensory interface we have employed a Lego NXT brick including touch sensor, sound sensor, light sensor and ultrasonic sensor as well. Combining the sensory media, we can differentiate a subject's physiological context such as running, walking, and driving. Figure 7 shows different steps of a scenario where sensory data triggers a user context and the framework dynamically chooses the COI and a set of services.

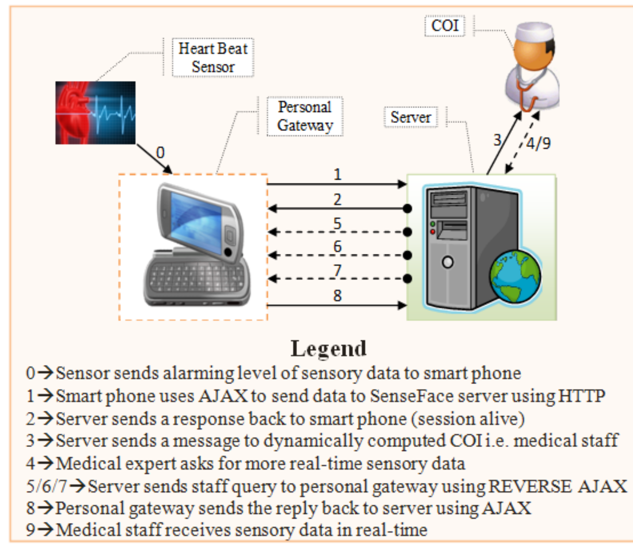


Fig. 7. Two-way sensory data communication between a BSN and COI at a given physiological context.

2) Social Network Stack

The Mashup layer has been implemented using PHP language. We retrieve the attributes of a user using OpenID 2.0 Attribute Exchange[12]. Portablecontacts has been implemented using RPX library[13]. An android OS-based version of PortableContacts has been implemented using jpoco[14]. Access control has been realized by the oauth-php[15] library. Authentication has been provided by OpenID library[16] such that a user can authenticate using any of his existing ID's; e.g. Yahoo ID, Windows Live ID, Facebook ID, Twitter ID, etc. The user profile metadata discovery protocol stack has been implemented using PHP-XRD[17]. To Mashup services we use Enterprise Mashup Markup Language (EMML[18]) and Google Social API implemented using PHP. 3A model has also been translated using a PHP script.

From the experience gained through our earlier research [17], we decided to employ Ant Colony Optimization

[9] http://www.skype.com/
[10] http://www.adobe.com/products/acrobatconnectpro/
[11] http://www.tandberg.com/

[12] https://verify.sxip.com/demorp/
[13] https://rpxnow.com/
[14] http://code.google.com/p/jpoco/
[15] http://code.google.com/p/oauth-php/
[16] http://www.janrain.com/developers
[17] http://github.com/willnorris/php-xrd
[18] http://www.jackbe.com/enterprise-mashup/

(ACO) algorithm to scrape the social ties dataset from different heterogeneous sources. The ACO algorithm works in two steps. During the first step, we employ forward and backward ants to find all the possible routes between a user and any service or a user profile in terms of a URI. Each forward ant looks for possible services a subject is connected to and the backward ants reinforce the found paths. For example, an ant can obtain an XRD document, which provides a list of links, and the relation of these links to the actual resource. As an alternative way is to parse an XRDS file that tells an ant a list of related services, and describes the properties of these related services. The ants also collect several criteria such as type of authentication required, how many hops to finally reach the destination, types of parameters needed, and the URI of the service. During second step, data ants use the information left behind by the forward and backward ants in the first step. Each data ant is dedicated to connect to each URI e.g. one to connect to Facebook, one to connect to an IMAP server, one to connect to calendar services and so on.

We use both AJAX and Reverse-AJAX for automatic client-server communication during different phases of ant algorithm. In order to collect social network information we have implemented open source Sphider[19] as search engine where we store information regarding information from different social network crude information by indexing URL's related to a user. Later on we use different queries to extract relationship to be used. Details about different social networks we have successfully mashed up can be found in [17].

3) Context-Aware Services

In order to design a proof of concept prototype we have implemented the following services.

- Health Services (Google Electronic Health Record, Online Prescription, Calorie Burnt/Wight Loss)
- Location Services (Home or outdoor location tracking using Wi-Fi/Cell triangulation and GPS)
- Sensor-assisted Services (Heart beat sensing, Fall detection, In Conversation)
- Communication Services (Email, SMS, Fax, MMS, Voice mail, RSS)
- Audio/Video Conferencing Services (Skype, Google Talk, ADOBE ConnectPro)
- IM Services (MSN, Yahoo Messenger, Google Talk)
- Social Networks (Facebook, YouTube, Flickr, Twitter)
- E-learning Services (Moodle, WebCT)
- Collaboration Services (Google Docs)
- Event Services (Google calendar, Yahoo Weather)

4) Web Portal

To test our system we implemented a web portal using open source iMoogle [21], which renders each service within a portlet. iMoogle has been configured to dynamically read a JSON[20] file containing the spatial layout of each portlet, where the spatial layout is dynamically updated depending on user context. The server side portlet programming is done using PHP, Perl and Python scripts. JavaScript framework has been implemented using Yahoo YUI[21] and for REST framework, we use Konstrukt[22]. Figure 8 shows the web portal updated dynamically as the user context is changed.

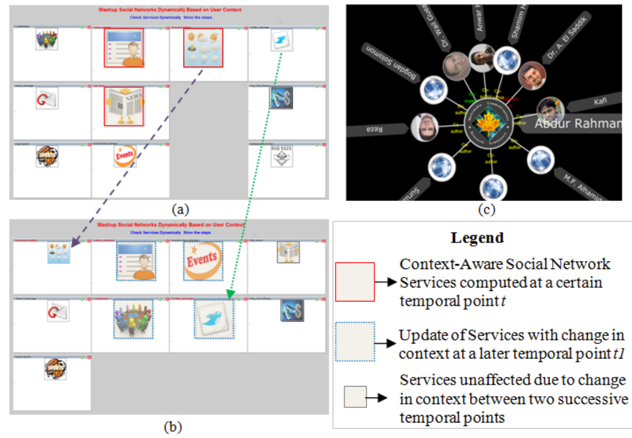


Fig. 8. A web portal showing social network services (a) services with large rectangular icons are services that need immediate attention from the user at a given context, (b) updated user context dynamically updates the portal content as well as the layout, and (c) an instance of context-aware retrieval of COI visualized in a flash interface.

IV. Framework Evaluation

1) Body Sensor Network

We designed the BSN using the software architecture called the Representational State Transfer (REST[23]). REST is an architectural style of accessing any resource from the web. REST has three unique design advantages: using REST we can represent each sensor by a unique URI; each URI can be accessed using HTTP protocol; it is very light weight compared to SOAP messages. REST architecture is used to define the access pattern of each sensor. For example, the REST interface to access ECG data using HTTP GET method can be as following:

http://example.com/SenseFace/people/arahman/sensors/ECG

In total, 15 participants took part in evaluating different features of the proposed system over 11 months, from September 2008 to July 2009. The participants are chosen from different ethnicities, ages, sex, geographic locations, and professions. Tests have been conducted at different times of the day to accommodate varying Internet and cellular network traffic conditions. We have tested the BSN in both upstream and downstream links. In order to test the upstream sensory data communication, we have defined some predefined activities and associated sensory data threshold that should trigger data push to remote server. We have tried to simulate some events to find out

[19] http://www.sphider.eu/
[20] http://www.json.org/
[21] http://developer.yahoo.com/yui/3/
[22] http://konstrukt.dk/
[23] http://www.ics.uci.edu/~fielding/pubs/dissertation/top.htm

the performance of the sensory data push. The events that we considered in our case are abrupt sit down from standing position and suddenly applying hard brake to a high-speed car. In the former case, the motion and the accelerometer sensors trigger the sensory data push while in the latter case, the combination of GPS, motion and accelerometer sensors take part in initiating the sensory data push. To test the heart beat related event, we relied on the fact that when a person runs, his heart beat rate increases significantly, which is considered an alarming situation (e.g., heartbeat beyond 180 per minute).

Table–I Sensory data access round-trip delay between a BSN and members of COI

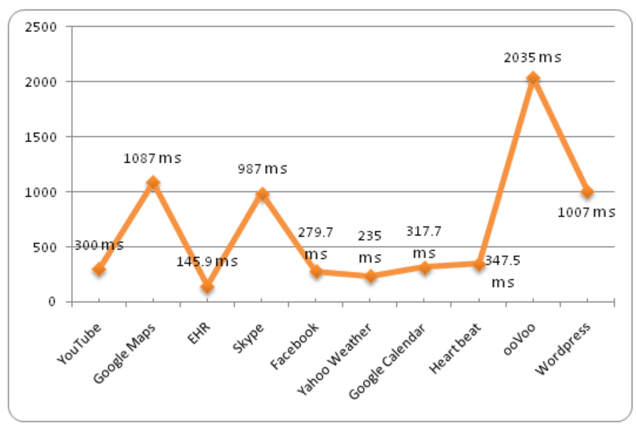
Sensor Type	#RESTful Request	#Successful Response	Success Ratio (%)	Avg. Delay (s)
Temperature	78	74	94.9	1.55
GPS	55	52	94.5	3.25
Motion	67	65	97	1.34
Accelerometer	70	68	97.1	1.65
Heart Rate	35	32	91.4	1.23

As for the upstream communication, up to 98% of the time, the framework was able to push sensory data to the SenseFace[24] server. We attribute this 2% times of failure to the poor data coverage by the mobile networks while a person is roaming. To avoid state loss of a mobile user where there is hardly any Wi-Fi or cellular network coverage, one might resort to alternative approaches. For example, we can use APRS network if a user is outdoor. In the worst case, a delay tolerant scheme might be adopted such that in such state, the system caches the request or response and as soon as the network coverage is available, it finishes the transaction at the cost of additional delay. To test the downstream data communication, we tried to inquire different sensory data in a RESTful way. Table-I shows the total number of RESTful requests sent and total number of correct responses received and the average delay of successful responses. While testing each sensory data access, we employ a round trip timer to calculate the associated delay. Because unsuccessful requests produced null value, we could only receive delay values for the successful responses. The delay shown in Table-I refers to round trip delay and calculated as follows:

Average round trip delay = downstream processing delay (at web server) + downstream network delay (Wi-Fi/cellular) + processing delay (at smart phone to communicate with the sensor(s)) + upstream network delay (Wi-Fi/cellular) + upstream processing delay (at web server).

Among the sensors, the average delay in the case of GPS is high in comparison to other sensors. This is due to the fact that establishing connection with the satellites takes more time than with Wi-Fi or cellular based location tracking techniques. We have identified several reasons of the miss ratio, as elaborated in [22], [23]. First, how the smart phone and the sensors are aligned and the distance between them has impact on channel loss ratio because the sensors are wirelessly connected with the smart phone using Bluetooth. Second, the mobile OS puts several restrictions on user written applications running at the OS background while the smart phone is in sleeping or conversation mode. Third, in many cases, the pairing between a sensor and the smart phone got distorted and thus lost the I/O request. Fourth, a subject was outside of coverage of both Wi-Fi and cellular data network while he/she was mobile. Finally, in the case of location tracking we used cell triangulation, Wi-Fi triangulation and GPS signals. Frequent mobility of a subject from indoor to outdoor or vice versa is the major cause of this location-related delay.

We have tried to relate the miss ratio shown in Table-I with the above mentioned 5 causes. Out of 14 missed I/O incidences 5 instances of failure were identified as University VPN issue, 3 were reported as Bluetooth connection loss between the sensor and the smart phone, 2 were caused by Wi-Fi or cellular data coverage problem, 2 were lost while the person was in conversation mode, and the rest were caused by other unidentifiable causes that yet need to be analyzed. We thus attribute the above reasons to these external causes that are not part of the proposed framework and can be improved by taking appropriate measures. A noticeable reason that we have discovered is that every user has his/her own style of organizing sensors and the smart phone spatially. These diversified characteristics can cause one or more source of I/O misses. This requires thorough investigation and research to mitigate these effects and make the BSN more stable, which will be addressed in our future works.



Legend
EHR[25]: Electronic Health Record

Fig. 9. Average round trip delay in milliseconds in rendering a typical instance of several services using the proposed system.

2) Service Rendering Performance

Figure 9 shows the average delay of rendering the services to the user interface taking into account the following factors:

[24] http://137.122.89.108/SenseFace/demo.php

[25] http://www.auditor.on.ca/en/reports_en/ehealth_en.pdf

Average round trip delay of an individual service = (The delay posed for creating the JSON containing the skeleton and content of each portal, performing database operation, and calculating credits in the web server + rendering the web services in a portal) / N

where N is the total number of test instances for each service, which is 200 in this case. The round trip delay showed in Figure 9 takes into consideration various traffic conditions such as variations due to day and night time, Wi-Fi and cellular data network switching, and access from desktop machine and smart phone. Among all the services, the audio/video conferencing client ooVoo[26] costs the highest delay because of its complex authentication mechanism and java plug-in response delay.

V. Conclusion and Future Work

In this paper, we have presented a framework to augment user context awareness with the social networks domain. We have tried to exploit the open stack in order to extract a user's personal social network from heterogeneous sources of Internet. We then elaborated the design of a ubiquity stack that can capture user context in real-time using sensors and Internet-based services and semantically map a subset of social networks ties and relevant services. Using off-the-shelf hardware, social network APIs, and open source tools, we have implemented both stacks where different sensory data input from a personal body area network is pushed to the appropriate community of interest using dynamically selected services. As a possible future research avenue, we are planning to carry out a usability test to find out the quality of user experience. In addition, we are studying several possibilities of addressing the privacy model of sharing private information over the web.

References

[1] M. Baldauf, S. Dustdar, and F. Rosenberg. A survey on context-aware systems. *Int. Journal of Ad Hoc and Ubiquitous Computing*, vol. 2, no. 4, pp. 263-277, Jun. 2007.

[2] D. Phung, B. Adams, K. Tran, S. Venkatesh and M. Kumar, "High accuracy context recovery using clustering mechanisms," in *Proc. IEEE International Conference on Pervasive Computing and Communications*, pp.1-13, Mar. 2009.

[3] J. An, S. Pack, S. An, M. Kim and Y. Jeon, "A novel service architecture for personalized context aware services," in *Proc. 11th International Conference on Advanced Communication Technology*, vol. 1, pp. 554 – 559, Feb. 15-18, 2009.

[4] Y. Kawahara, H. Kurasawa and H. Morikawa, "Recognizing User Context Using Mobile Handsets with Acceleration Sensors," in *Proc. IEEE International Conference on Portable Information Devices*, pp. 1-5, May 25-29, 2007.

[5] R.Y. Shtykh and J. Qun, "Capturing User Contexts: Dynamic Profiling for Information Seeking Tasks," in *Proc. 3rd International Conference on Systems and Networks Communications*, pp. 365 - 370, Oct. 26-31, 2008.

[6] L. H. Leong, S. Kobayashi, N. Koshizuka and K. Sakamura, "CASIS: a context-aware speech interface system," in *Proc. of the 10th international conference on Intelligent user interfaces*, pp. 231-238, 2005.

[7] A. Joly, P. Maret and J. Daigremont, "Context-Awareness, the Missing Block of Social Networking", *International Journal of Computer Science and Applications*, Technomathematics Research Foundation, vol. 6, no. 2, pp. 50-65, 2009.

[8] P. Velardi, R. Navigli, A. Cucchiarelli and F. D. Antonio, "A New Content-Based Model for Social Network Analysis," *IEEE International Conference on Semantic Computing*, pp. 18-25, 2008.

[9] D. Bottazzi, R. Montanari and A. Toninelli, "Context-Aware Middleware for Anytime, Anywhere Social Networks," *IEEE Intelligent Systems*. vol. 22, no. 5, pp. 23-32, 2007.

[10] E. M. Tapia. (2008). Using machine learning for real-time activity recognition and estimation of energy expenditure. Thesis (Ph. D.)--MIT, School of Architecture and Planning. [Online]. Available: http://dspace.mit.edu/handle/1721.1/44913

[11] Md. A. Rahman, A El Saddik, and W. Gueaieb, "Data Visualization: From Body Sensor Network to Social Networks," in *Proc. of the IEEE International Workshop on RObotic and Sensors Environments*, pp. 157 – 162, 2009.

[12] Open Stack. [Online]. Available: http://blog.broadbandmechanics.com/2009/05/02/open-stack-v2/

[13] A. Kapadia, N. Triandopoulos, C. Cornelius, D. Peebles and D. Kotz, "AnonySense: Opportunistic and Privacy-Preserving Context Collection," in *Proc. of the 6th International Conference on Pervasive Computing*, pp. 280–297, May 19–22, 2008.

[14] J. Zhang, M. Karim, K. Akula and R. Ariga, "Design and Development of a University-Oriented Personalizable Web 2.0 Mashup Portal," in *Proc. IEEE International Conference on Web Services*, pp. 417-424, Sep. 23-26, 2008.

[15] S.J. Miah, J. Gammack and G. Greenfield, "Mashup technologies for building End-user enabled business portal," in *Proc. 3rd IEEE International Conference on Digital Ecosystems and Technologies*, pp.121-126, Jun. 1-3, 2009.

[16] S. El Helou, C. Salzmann, S. Sire and D. Gillet, "The 3A Contextual Ranking System: Simultaneously Recommending Actors, Assets, and Group Activities," in *Proc. of the 3rd ACM Conference on Recommender Systems*, pp. 373-376, Oct. 22-25, 2009.

[17] Md. A. Rahman, A. El Saddik and W. Gueaieb, "Building Dynamic Social Network From Sensory Data Feed," *IEEE Transactions on Instrumentation and Measurement*, vol. 59, no. 5, pp. 1327 – 1341, 2010.

[18] A. Bahri, R. Bouaziz and F. Gargouri, "Fuzzy ontology implementation and query answering on databases", *28th NAFIPS2009*, pp.1-6, Jun. 2009.

[19] S. Gallova, "Fuzzy Ontology and Information Access on the Web", *International Journal of Computer Science*, vol. 34, no. 2, 2007.

[20] X. Hu et al., "A Mobile Service Application Platform with Chinese Short Message Based on Fuzzy Ontology for Employment of University Graduates," *8th International Conference on Mobile Business*, pp.164-169, Jun. 27-28, 2009.

[21] iMoogle [online]. http://www.moonkiki.com/moonkiki/imoogle/

[22] M. A. Hanson et al., "Body Area Sensor Networks: Challenges and Opportunities," *IEEE Computer*, vol. 42, no. 1, pp. 58-65, 2009.

[23] B. De Silva, A. Natarajan and M. Motani, "Inter-User Interference in Body Sensor Networks: Preliminary Investigation and an Infrastructure-Based Solution," *6th International Workshop on Wearable and Implantable Body Sensor Networks*, pp. 35-40, 2009.

[24] C. Bolchini, C. Curino, E. Quintarelli, F. A. Schreiber and L. Tanca, "A data-oriented survey of context models," *SIGMOD Record*, vol. 36, no. 4, pp. 19-26, 2007.

[25] D. Preuveneers et al., "Towards an extensible context ontology for ambient intelligence," in *Proc. 2nd European Symp. Ambient Intelligence*, LNCS 3295, pp. 148-159, 2004.

[26] M. Kaenampornpan and E. O'Neill, "An integrated context model: Bringing activity to context," in *Proc. Workshop on Advanced Context Modeling, Reasoning and Management*, 2004.

[27] A. K. Dey, "Understanding and Using Context," *Personal Ubiquitous Comput.*, Springer-Verlag, vol. 5, no. 1, pp. 4-7, 2001.

[26] http://www.oovoo.com/