



uOttawa

L'Université canadienne
Canada's university

**FACULTÉ DES ÉTUDES SUPÉRIEURES
ET POSTDOCTORALES**



**FACULTY OF GRADUATE AND
POSTDOCTORAL STUDIES**

Md. Abdur Rahman

AUTEUR DE LA THÈSE / AUTHOR OF THESIS

Ph.D. (Electrical and Computer Engineering)

GRADE / DEGREE

School of Information Technology and Engineering

FACULTÉ, ÉCOLE, DÉPARTEMENT / FACULTY, SCHOOL, DEPARTMENT

SenseFace: Towards a Context-Aware Social Network Framework

TITRE DE LA THÈSE / TITLE OF THESIS

Abdulmotaleb El Saddik

DIRECTEUR (DIRECTRICE) DE LA THÈSE / THESIS SUPERVISOR

Wail Gueaieb

CO-DIRECTEUR (CO-DIRECTRICE) DE LA THÈSE / THESIS CO-SUPERVISOR

Peter Liu

Shervin Shirmohammadi

**Jason Gu
Dalhousie University**

Jiying Zhao

Gary W. Slater

Le Doyen de la Faculté des études supérieures et postdoctorales / Dean of the Faculty of Graduate and Postdoctoral Studies

SenseFace: Towards a Context-Aware Social Network Framework

Md. Abdur Rahman

Thesis Submitted to the Faculty of Graduate and Postdoctoral Studies
In partial fulfillment of the requirements
for the degree of
Doctor of Philosophy in Electrical and Computer Engineering

Ottawa-Carleton Institute for Electrical and Computer Engineering
School of Information Technology and Engineering
Faculty of Engineering
University of Ottawa



uOttawa



Library and Archives
Canada

Published Heritage
Branch

395 Wellington Street
Ottawa ON K1A 0N4
Canada

Bibliothèque et
Archives Canada

Direction du
Patrimoine de l'édition

395, rue Wellington
Ottawa ON K1A 0N4
Canada

Your file *Votre référence*
ISBN: 978-0-494-74248-8
Our file *Notre référence*
ISBN: 978-0-494-74248-8

NOTICE:

The author has granted a non-exclusive license allowing Library and Archives Canada to reproduce, publish, archive, preserve, conserve, communicate to the public by telecommunication or on the Internet, loan, distribute and sell theses worldwide, for commercial or non-commercial purposes, in microform, paper, electronic and/or any other formats.

The author retains copyright ownership and moral rights in this thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without the author's permission.

In compliance with the Canadian Privacy Act some supporting forms may have been removed from this thesis.

While these forms may be included in the document page count, their removal does not represent any loss of content from the thesis.

AVIS:

L'auteur a accordé une licence non exclusive permettant à la Bibliothèque et Archives Canada de reproduire, publier, archiver, sauvegarder, conserver, transmettre au public par télécommunication ou par l'Internet, prêter, distribuer et vendre des thèses partout dans le monde, à des fins commerciales ou autres, sur support microforme, papier, électronique et/ou autres formats.

L'auteur conserve la propriété du droit d'auteur et des droits moraux qui protègent cette thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

Conformément à la loi canadienne sur la protection de la vie privée, quelques formulaires secondaires ont été enlevés de cette thèse.

Bien que ces formulaires aient inclus dans la pagination, il n'y aura aucun contenu manquant.


Canada

© Md. Abdur Rahman, Ottawa, Canada, 2011

ABSTRACT

Social networks offer multimedia information consumption and sharing with millions of people around the world. The backbone of social networks stems from the advancements of different Web 2.0 technologies. Moreover, smartphones equipped with high speed Internet access have added ubiquitousness to Internet-based services. This gives rise to the fact that many people consume diversified services on a daily basis and have developed an association with different communities of interest via these services. However, a person only accesses a subset of these services at a given time either to consume certain information or to share information with a community of interest (COI), depending on the context. This poses three important research questions that need to be addressed: 1) “How do we dynamically capture user context from heterogeneous sources?” 2) “Which services are found to be interesting by a subject in a given context?” and 3) “How do we define the COI in a given context?”

In this thesis, we propose detailed models and algorithms to answer the above three questions in the social network perspective. First, we propose algorithms to dynamically extract the services and social ties of an individual from the global heterogeneous social network space, for example, the Internet, which is called a personal social network (PSN). Second, we propose a framework named SenseFace, which employs a novel ubiquitous stack to extract user context from two sources: body sensor network (BSN) and the PSN. The BSN helps us in capturing different vital physical information, body activity and ambient information to infer user context while the PSN offers user context in the form of events, real-time messages, to-do lists, location and so on. Third, we present detailed models and algorithms to dynamically map a user context with a subset of relevant services and a subset of social network members. Finally, we present a prototype addressing the detailed

design and implementation of the framework. The effectiveness of the framework has been validated by conducting both usability and run time performance testing.

ACKNOWLEDGMENTS

I am thankful to my dear LORD, ALLAH, who has created me and given me intelligence and potential for doing research. Then, my sincere gratitude goes to my academic supervisor Dr. Abdulmotaleb El Saddik and co-supervisor Dr. Wail Gueaieb, who helped, encouraged, and guided me towards my academic as well as my personal success. The thesis would not be successful without enormous help from them. They provided me a substantial amount of help and suggestions throughout my work. A special gratitude also goes to Dr. Heung-Nam Kim and fellow members of the Multimedia Communication Research Laboratory team for many helpful discussions and their constant encouragement and support. I would like to give my sincere thanks to my beloved parents, wife and children for their support throughout my graduate studies. I am also grateful to all my friends and family who helped me either with suggestions or with positive affirmation. I would like to acknowledge the financial assistance of the Ontario Government for its Ontario Graduate Scholarship (OGS), the University of Ottawa for Admission Scholarship and Excellence Scholarship and the Natural Sciences and Engineering Research Council of Canada (NSERC).

TABLE OF CONTENTS

ABSTRACT	III
ACKNOWLEDGMENTS.....	V
TABLE OF CONTENTS.....	VI
LIST OF FIGURES.....	X
LIST OF TABLES	XIII
GLOSSARY OF TERMS	XIV
CHAPTER 1: INTRODUCTION	1
1.1 Context-awareness in Social Networks.....	1
1.2 Motivation Scenario: Ubiquitous Social Networks.....	4
1.3 Problem Statement and Challenges.....	7
1.4 Contributions.....	10
1.5 Scholarly Activities and Publications	10
1.6 Thesis Organization	13
CHAPTER 2: BACKGROUND AND RELATED WORK.....	14
2.1 Open Stack of Internet	14
2.1.1 Layer 1: Unique User ID	14

2.1.2 Layer 2: User Profile Metadata Discovery.....	14
2.1.3 Layer 3-A: Access Control.....	15
2.1.4 Layer 3-B: Authentication.....	15
2.1.5 Layer 4-A: Attributes	16
2.1.6 Layer 4-B: Contacts	16
2.1.7 Layer 5: Mashup of Social Networks.....	16
2.2 Community of Interest	17
2.3 Personal Social Networks	18
2.4 Context-Aware Services in Social Networks.....	20
2.5 Context-Aware Sensor Frameworks	23
2.6 Comparison with Related Works	28
2.7 Summary	31

CHAPTER 3: MODELING UBIQUITY AND CONTEXT-AWARENESS IN

PERSONAL SOCIAL NETOWRKS	32
3.1 Personalized Social Networks and Services	33
3.1.1 Deriving Personal Social Networks from Heterogeneous Sources.....	34
3.1.2 Extracting Internet-based Services from Heterogeneous Sources	36
3.1.3 Composite Services and Social Tie Modeling	37
3.2 Modeling User Context.....	38
3.2.1 Extracting User Context from Body Sensor Networks.....	45

3.2.2 Extracting User Context from Heterogeneous Services.....	45
3.3 Mapping User Context to Community of Interest and Services	46
3.3.1 Dynamic Context-Aware Services	46
3.3.2 Context-based Communities of Interest.....	47
3.3.3 Service Priority Modeling	47
3.4 Summary	50

CHAPTER 4: DESIGN OF CONTEXT-AWARE SOCIAL NETWORK

FRAMEWORK.....	51
4.1 Extracting Personal Social Network and Services Information	56
4.1.1 Aggregation of Heterogeneous Services	59
4.1.2 Extracting Social Ties and Services Information.....	60
4.1.3 Social Network Analysis.....	64
4.2 Collecting User Context from BSN and Internet-based Services	66
4.2.1 Context Extraction from Body Sensor Network	68
4.2.2 Gathering Context from Internet-based Services	72
4.2.3 Alignment of Heterogeneous Context Data	77
4.2.4 Context Management	78
4.3 Context-Aware Services and COI.....	81
4.3.1 Ubiquity Stack.....	81
4.3.2 Service Layout Manager	88

4.4 Summary	88
CHAPTER 5: FRAMEWORK IMPLEMENTATION.....	89
5.1 Personal Social Network Extraction Subsystem.....	89
5.1.1 Open Stack for Personal Social Network Discovery	89
5.1.2 Personal Social Network Extraction	92
5.2 Context Extraction Subsystem	95
5.3 Dynamic Service and COI Adaptation Subsystem	99
5.4 Summary	101
CHAPTER 6: TEST RESULTS.....	102
6.1 Run-time Performance Study	103
6.1.1 Service and Social Tie Access	103
6.1.2 Sensory Data Access	107
6.1.3 Dynamic Services and COI Rendering	110
6.2 Concurrency	112
6.3 Usability	117
6.4 Summary	134
CHAPTER 7: CONCLUSIONS AND FUTURE WORK.....	135
Future Plan	138
REFERENCES.....	140

LIST OF FIGURES

Figure 2.1 Open stack of Internet to extract services and social ties of a person.	14
Figure 3.1 Different entities involved in modeling context-aware social networks.	33
Figure 3.2 A generic Venn diagram showing overlapping multiple sources of social ties where each social tie information might be extracted from more than one source.	35
Figure 3.3 Context topology for modeling different levels of context primitives.	41
Figure 3.4 A display divided into blocks containing credit values.	48
Figure 4.1 Block diagram of the proposed SenseFace context-aware personal social networking system.	52
Figure 4.2 Block diagram showing major components of personal social network extraction subsystem.	59
Figure 4.3 Dataset matrix stored within index server.	63
Figure 4.4 Forming personal social networks from the set of uncategorized social network dataset.	66
Figure 4.5 Block diagram showing components that extract context from the sensory data.	67
Figure 4.6 Block diagram of the context retrieval system from different heterogeneous Internet-based services.	73
Figure 4.7 Preparing raw sensory and social network dataset for higher layer context mapping using 3A model.	77
Figure 4.8 Block diagram showing major components within the context manager, external input sources of context or clients of context manager.	78
Figure 4.9 Dynamic service and community of interest adaptation subsystem.	81
Figure 4.10 COI matrix that represents each ego and his community of interest depending on the type of sensory data.	87

Figure 5.1 Popularity of existing API's exposed by existing Internet-based services (courtesy: ProgrammableWeb).	92
Figure 5.2 Body sensor network (a) Garmin Forerunner heart beat sensor, (b) Sun SPOT WSN mote, (c) An HTC dream running Android 2.2, (d) EZ-430 Chronos sports watch, (e) NXT brick including touch sensor, sound sensor, light sensor and ultrasonic sensor, and (f) Built-in sensors and communication channels comes with the HTC smartphone.	96
Figure 5.3 A JSON response from a temperature sensor as a result of a RESTful query from a remote service.	97
Figure 5.4 Two-way sensory data communication between a BSN and COI at a given physiological context.	98
Figure 5.5 Portal to visualize context-aware social ties and services with two metaphors (a) grid-based layout metaphor and (b) sliding window metaphor. ..	100
Figure 5.6 A web portal showing social network services (a) services with large rectangular icon 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 layout, and (c) an instance of context-aware retrieval of COI visualized in flash interface.	101
Figure 6.1 Seven samples of end-to-end delay of 12 randomly selected services.	104
Figure 6.2 Social tie extraction delay: Maximum and average delay of 25 randomly selected community oriented service clients.	106
Figure 6.3 Downstream sensory data access by COI.	108
Figure 6.4 Average delays in milliseconds in rendering a typical instance of several services and associated COI using the proposed framework.	111
Figure 6.5 Mean processing time of accessing SenseFace framework by 1000 users for 49.320 seconds.	112

Figure 6.6 Result details of total 1000 users accessing up-to ten randomly selected services at any given moment for 53.16 seconds.	116
Figure 6.7 Integrated feedback results from all the test subjects regarding the framework.	120
Figure 6.8 User feedbacks regarding global view of context-awareness.	121
Figure 6.9 Users' view about providing services based on a given user context.	122
Figure 6.10 Users' view regarding aggregation of one's all the social ties and consumed services.	123
Figure 6.11 User feedbacks regarding community oriented services.	124
Figure 6.12 User view of collecting a user and his/her ambient information from body sensor network.	125
Figure 6.13 User feedbacks regarding waiting time tolerability in rendering services within the SenseFace portal.	126
Figure 6.14 User view about benefit of single sign on for all the services.	127
Figure 6.15 User view about extracting content from services without social tie categories.	128
Figure 6.16 User responses on dynamic layout of the context-aware services and COI.	129
Figure 6.17 User comments about creating fine grained user context from context primitives.	130
Figure 6.18 User view about presenting total and/or context-aware social ties.	132
Figure 6.19 Non-technical users' feedback about different aspect of the framework. ...	133

LIST OF TABLES

Table 2.1: Comparison summary based on context related metrics.....	29
Table 2.2: Comparison summary based on framework related metrics.....	30
Table 3.1: Context schema definition.	39
Table 4.1: Deducing context primitives from sensory data.	72
Table 4.2: Example of raw web data received from services and corresponding deduced raw context data.....	76
Table 4.3: Deducing context primitives from sensory data and multimedia information of diversified Internet-based services.....	79
Table 6.1: Sensory data access round-trip delay between a BSN and members of COI.	108
Table 6.2: Concurrent access to one service without social tie class of service by 1000 users for 120 seconds.	115
Table 6.3: Usability and user satisfaction questionnaire (adapted from ISO 9241-11)..	118

GLOSSARY OF TERMS

AJAX – Asynchronous JavaScript and XML
API – Application Programming Interface
BSN – Body Sensor Network
COI – Community of Interest
COS – Community Oriented Services
ECG – Electro Cardio Gram
EMML – Enterprise Mashup Markup Language
FOAF – Friend of A Friend
GData – Google Data
GEDCOM – GEnealogical Data COMmunication
GPRS – General Packet Radio Service
GPS – Global Positioning System
GSM – Global System for Mobile Communications
EHR – Electronic Health Record
GUI – Graphical User Interface
IM – Instant Message
JSON – JavaScript Object Notation
LRDD – Link-based Resource Descriptor Discovery
MAC – Media Access Control
MMS – Multimedia Messaging Service
MSN – Microsoft Network
OAuth – Open Authorization
OS – Operating System
PDA – Personal Digital Assistant
PIM – Personal Information Manager
PSN – Personal Social Network
RPC – Remote Procedure Call
REST – REpresentational State Transfer
RSS – Really Simple Syndication
SMS – Short Message Service
SWST – Services without Social Ties
URL – Uniform Resource Locator
VPN – Virtual Private Network
Wi-Fi – Wireless Fidelity
WLAN – Wireless Local Area Network
WSDL – Web Services Description Language
XML – Extensible Markup Language
XRD – Extensible Resource Descriptor

CHAPTER 1: INTRODUCTION

Social networks have become tremendously popular in terms of information consumption, sharing and communication among groups of similar interests. Moreover, the popularity of social networks has been proliferated due to several multi-disciplinary breakthrough technologies offered by Web 2.0 and modern smartphones. Instead of creating, managing, sharing and accessing information on a specific place and time, a smartphone helps access social network information as well as contextual information of a user ubiquitously. 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. anytime and anywhere. To add to these advancements, we envision social network ubiquity by adding user context-awareness, which offers personalized and context-specific interaction with different services and communities of interest. The aim of this thesis is to design a context-aware social network framework that seamlessly bridges body sensor networks (BSN) with personal social networks (PSN). An additional aim is to develop models and algorithms that dynamically determine services, and communities of interests depending on a user's current context. In this chapter, we briefly introduce context-aware social networks and then show how the proposed framework contributes to these research fields.

1.1 CONTEXT-AWARENESS IN SOCIAL NETWORKS

“A social network is a social structure made of individuals (or organizations) called nodes, which are tied by one or more specific types of interdependency, such as friendship, kinship, financial exchange, dislike, sexual relationships or relationships of beliefs, knowledge or prestige.” [Wasserman, S., and Faust, K. (1994)]. Social

networks have been studied over a century from an interpersonal to international level in the context of sociology, anthropology, epidemiology, mass surveillance, biology, social psychology, information technology, health science, economics and communication systems. In this day, a person has wide opportunities of maintaining his/her social networks through smartphones and current Web 2.0 technologies such as collaborative blogging, social tagging, instant messaging (IM), Facebook, YouTube, Twitter and so on. Hence, numerous social network services offering multimedia information sharing have attracted millions of people around the globe. A survey by [Killworth, P. et al. (1990)] showed that on average, each person maintains approximately 1,500 social network ties of different strengths, which is assumed to be many-fold, due to the contribution of Web 2.0. However, a person only interacts with a subset of people from the total population of his/her social networks and uses a subset of services in any given context [Broens, T. et al. (2007)]. Unfortunately, to date, very little research has investigated this important phenomenon of social networks.

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." [Dey, A. K. (2001)]. 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. [Dey, A. K. (2001)]

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 [Phung, D. et al. (2009)], [An, J. et al. (2009)], [Kawahara, Y. et al. (2007)], [Shtykh, R.Y., and Qun, J. (2008)]. Context has been studied in areas such as personal context data (for example, user profile, user location, time of system access, to-do lists, types of end-devices used and contact lists), social context data (for example, social ties through social networks and interactions or types of information shared) and application based context data (for example, types of web services used, bandwidth and reliability requirements of each service, types of protocols and access mechanism needed for each service and URL of each service) [Gaonkar, S. et al. (2008)], [Bottazzi, D. et al. (2007)]. Previous research show that user context can be captured using both multimedia information stored in different Internet-based services and using BSN [Joly, A. et al. (2009)], [Velardi, P. et al. (2008)], [Sriram, J. (2009)], [Gartrell, C. M. (2010)], [Xu, X.-M. et al. (2008)].

The emerging smartphones offer several enhanced features such as increased CPU power and memory, Internet connectivity in multi-modal ways (for example, Wi-Fi, Bluetooth, 3.5G), facilities to interface with both its internal and external sensors, and numerous social network applications for chat, IM, audio/video collaboration and multimedia information sharing. A smartphone thus offers time and space independent services using its sheer computing and wireless communication capabilities, providing always online capability ubiquitously [Bardram, J. E. (2005)]. To be specific, a smartphone offers two important sources for information gathering: first, through built-in or external wearable sensors, forming a BSN, and second, through ubiquitous access to different services from the Internet. This opens the door of many new possibilities and next generation social network scenarios.

As discussed earlier, social networks connect people with people and with information that they are looking for. To achieve this information sharing, existing social network services employ typical media such as images, audio, video, text,

animation, online blogs, metadata, shared tagging, word documents, power point slides, RSS feeds, etc. One common aspect of these social network services is that a group of people creates the information source and others consume and/or manipulate. In a similar fashion, sensor networks can capture physical events and even go beyond the capability of what these existing social media can capture by operating without the need of any human intervention [Teixeira, T. et al. (2009)], [Tapia, E. M. (2008)]. For example, an elderly person carrying a smartphone paired with a blood pressure sensor can capture any unusual blood pressure levels and send this sensory information live to his/her personal social network that might include family members, a physician, emergency services, friends, colleagues and so on [Wu, W. et al. (2008)]. Hence, adding sensory media, which is the missing block of existing social networks, can open a new era of social computing paradigm, namely context-aware social networks, by providing the ubiquitous feature of anytime and anywhere [Bhardwaj, S. et al. (2008)], [Bottazzi, D. et al. (2006)].

1.2 MOTIVATION SCENARIO: UBIQUITOUS SOCIAL NETWORKS

Let us assume a scenario where a person named Alice interacts with different members of her social network throughout a day. Alice wears several sensors that keep track of her physiological phenomena, activities and ambient condition. The sensors are paired with her smartphone that has ubiquitous access to the Internet. Alice uses several Internet-based services to communicate, consume and share information with the members of her personal social networks. Let us also assume that Alice uses the following services for different professional and personal activities:

- audio/video conferencing services such as Skype and Adobe ConnectPro,
- communication services such as email and SMS,

- health services such as Google Electronic Health Record, Online Prescription, and Calories Burned/Weight Loss,
- location services such as home or outdoor location tracking using Wi-Fi/cell triangulation and GPS,
- sensor-assisted services such as heartbeat and blood pressure sensing,
- instant messaging services such as MSN, Yahoo Messenger, and Google Talk,
- social portals such as Facebook, YouTube, Flickr and Twitter,
- e-learning Services such as Moodle and WebCT,
- collaboration services such as Google Docs and
- event services such as Google calendar and Yahoo Weather.

On a typical day after getting up from bed, Alice was informed by Google calendar service that she had an online project meeting from 10 a.m. to 11 a.m. with a group of three research teams from three different universities. She also needs to take her child to the family physician's office for a medical appointment from 12:30 p.m. to 1:30 p.m., and then must join another project meeting from 2:30 p.m. to 4:00 p.m. with a group of industrial collaborators. At 5:00 p.m., she has to teach a course at the university. While coming from the afternoon meeting, due to severe weather and traffic conditions, she predicts she will be about 2 minutes late arriving to class, which is to start at 5:00 p.m. sharp. She needs to relay this delay to the students immediately.

After finishing teaching, Alice feels severe pain in her heart while driving home which is noted by the wearable ECG sensor. This alarming event needs to be instantaneously relayed to her concerned community of interest, such as the family physician, caregiver family members, nearby friends, living partner and emergency services. The family physician receives the alert message but needs to verify the severity level and sends an SMS message to Alice's smartphone requesting last five minutes of raw sensory data in order to analyze the context. After receiving the SMS

message, Alice's smartphone begins sending the requested sensory data directly to the family physician. In case of emergency, the family physician recommends that emergency services bring Alice to the hospital, and notifies her caregiver family members. The family physician also checks Alice's medical history from the Electronic Health Records, which is regularly updated.

In the above scenario, the types of people Alice interacts with and the types of services she uses may be completely different and require her to manually invoke these services and maintain the appropriate community of interest. In this scenario, the privacy issue is out of scope, although it is considered an important aspect of social networks. Technically, the above scenario requires a framework that has the knowledge of the global set of social ties and the services Alice is attached with. Then, depending on Alice's context, the framework can map necessary services and a subset of social network members. For example, in case Alice is delayed by two minutes to the classroom, she only needs to notify the students attending the class immediately. The framework should be able to track user context using the information coming from BSN and different services throughout the day ubiquitously, irrespective of time and location [Upendra, R., and Max, O. (2008)]. Similar to Alice, every person's context is different and everyone maintains different associations with people and services.

Several requirements can be derived from the above scenario. First, the framework should be able to merge the BSN with PSNs. For instance, if a sensory data has to travel from one's BSN to one's health physician or vice versa, then the framework needs to incorporate several network tiers including BSN, cellular/WLAN, Internet and social networks [Sun, F. et al. (2007)]. This includes the mobility support of both a person and his/her social ties, whether indoors or outdoors [Bardram, J. E. (2005)].

Second, the Internet is the social network of billions of people where a particular user's presence is stored in different heterogeneous sources. The framework should provide a facility to extract one's personal social network from the Internet and save it as a personal social network profile [Shtykh, R. Y., and Qun, J. (2008)]. Moreover, a person's social ties and associations with different services change over time. This requires the framework to update one's longitudinal personal social network information. Once the total set of social ties of a person is extracted from heterogeneous sources, they need to be filtered and categorized according to some social factors to be able to semantically map each social tie to a user context. For example, one's social ties extracted from Facebook and Twitter may fall into different categories such as family, friend, colleague, physician, etc. This can be achieved by employing social network analysis methods. The categorized social ties help us in defining community of interest (COI) that can be mapped with a given user context [El Helou, S. et al. (2009)].

Third, in order to find a user context, the person has to carry the wearable sensors and the smartphone and be connected with the Internet through cellular network or Wi-Fi. Fourth, the framework should provide a way of extracting user context from BSN and multimedia information contained within social network services and map it with a subset of consumed services and members of COI [Leimeister, J. M. et al. (2002)]. In other words, the framework should be ubiquitous, understand user needs and activities and recommend necessary lists of services and social ties.

1.3 PROBLEM STATEMENT AND CHALLENGES

As stated above, we can ubiquitously capture user context using both BSN and multimedia information contained within various services [Kawahara, Y. et al. (2007)], [Lorincz, K. et al. (2007)], [Leong, L. H. et al. (2005)]. Analyzing information

contained in one's personal social network provides very useful contextual information [Koolwaaij, J. et al. (2009)]. Similarly, sensors deployed in a human body or one's surrounding environment can provide a rich source of event and ambient information [Sriram, J. (2009)], [Tapia, E. M. (2008)]. However, there have been few efforts to combine both the sources of user context in social network domain. Merging social networks with sensor networks gives rise to our vision of the next generation, ubiquitous social networks, as follows:

“Social networks will be smart, be personalized, and understand both intra-user and inter-user's context by combining information of both BSN and social networks in real-time, anytime and anywhere.”

Lack of previous research in this exciting cross domain has left a gap between a user's personal social networks and his/her context. In fact, the type of service(s) a user uses and subset of people he/she interacts with at any given moment depends on his/her contextual factors such as age, ethnicity, profession, medical background, time of a day, location, interests, hobbies, physical activities and phenomena [Santana, P.C. et al. (2005)]. Hence, both an inter-user and an intra-user social network interaction are inherently diversified and completely depend on a user context [Silva, B. De et al. (2009)]. This has motivated us in designing and developing a context-aware social network framework called SenseFace that merges one's BSN with the PSN to dynamically find the community of interest and relevant social network services at any given moment.

Due to the nascent nature of this new research arena, we first identify the challenges to realize such a framework that are as follows:

1. The numerous growths of diversified Internet-based services led to difficulties in managing individual social ties and services [Sarkar, P., and Moore, A. W. (2007)]. People either consume services or communicate with members of personal social networks via diverse sources such as personal websites, web services, social

portals, emails, and blogs that require overwhelming varieties of complex authentication and access control mechanisms [Singh, L. et al. (2007)]. Hence, extracting information about all the members of one's PSN and all the services a person interacts at different temporal points is a challenging task [Zhang, J. et al. (2008)], [Miah, S. J. et al. (2009)], [Gaonkar, S. et al. (2008)], [Mika, P. (2007)], [Killworth, P. et al. (1990)].

2. Capturing user context ubiquitously from BSN and multimedia information contained within heterogeneous Internet services is a challenging task [Shtykh, R. Y., and Qun, J. (2008)], [Hanson, M. A. et al. (2009)], [Broens, T. et al. (2007)], [Ganti, R. K. et al. (2008)]. For the former case, accessing a particular live sensory data from a BSN over Internet, especially when both the person wearing the sensors as well as the COI are both mobile, is a challenging task [Nurmi, P. et al. (2010)], [Christopoulou, E. (2008)], [Munoz, M. A. et al. (2003)]. For the latter case, deducing user context from multimedia information contained in the form of email messages, RSS feeds, web services data messages, text messages, IMs, content management systems, to-do lists, calendar events, blog messages and other online sources is a challenging task [Gartrell, C. M. (2010)], [Joly, A. et al. (2009)].
3. Associating a user with a subset of social network members, for example, COI, and a subset of social network services at a given user context poses two challenges. First, both the inter-user context at any given moment and intra-user context throughout a day varies dynamically [Gartrell, C. M. (2010)], [An, J. et al. (2009)], [Joly, A. et al. (2009)]. Second, mapping user context with relevant services and COI needs to accommodate human preferences [Balasundaram, B. (2008)], [Bottazzi, D. et al. (2007)]. Hence, designing and modeling such a context-aware social network system is a challenging task.

1.4 CONTRIBUTIONS

The main focus of this thesis is to design and develop a ubiquitous framework that can dynamically determine services and COI depending on a particular context of a user. In order to handle the research challenges outlined in the previous section, the contributions of this thesis are as follows:

1. Design and development of algorithms to dynamically extract one's personal social network and consumed services from diversified sources of the Internet.
2. Design and development of models and algorithms to capture user context from both BSN and different heterogeneous services.
3. Design and development of algorithms to find a subset of social ties such as COI and subset of relevant services depending on user context.
4. Design and development of an open source framework, called SenseFace, which bridges BSN with PSN and dynamically recommends context-aware services and context-aware COI to users.

1.5 SCHOLARLY ACTIVITIES AND PUBLICATIONS

The following publications are the outcome of the research work presented in this thesis.

Journal Papers

- [1] **Md. Abdur Rahman**, Heung-Nam Kim, Abdulmotaleb El Saddik, and Wail Gueaieb. A Context-aware Framework toward Personal Social Network Services. *IEEE Transactions on Instrumentation and Measurement* (accepted), 2010
- [2] **Md. Abdur Rahman**, Abdulmotaleb El Saddik, and Wail Gueaieb. Building Dynamic Social Network From Sensory Data Feed. *IEEE Transactions on Instrumentation and Measurement*, vol. 59, no. 5, pp. 1327-1341, May 2010.

- [3] **Md. Abdur Rahman**, Abdulmotaleb El Saddik, and Wail Gueaieb. Augmenting Context-Awareness by Combining Body Sensor Networks and Social Networks. *IEEE Transactions on Instrumentation and Measurement* (in press), 2010.
- [4] **Md. Abdur Rahman**, Abdulmotaleb El Saddik, Wail Gueaieb. “Wireless Sensor Network Transport Layer: State of the Art.” In Subhas Mukhopadhyay and Ray Huang, editors, *Sensors: Advancement in Modeling, Design Issues, Fabrication and Practical Applications, Lecture Notes in Electrical Engineering*. Springer-Verlag, pp. 221-245, April 2008.
- [5] **Md. Abdur Rahman**, Suruz Miah, Wail Gueaieb, and Abdulmotaleb El Saddik. SENORA: A P2P Service Oriented Framework for Collaborative Multi-robot Sensor Network. *IEEE Sensors Journal*, Vol. 7, No. 5, pp. 658 – 666, May 2007.
- [6] Abdulmotaleb El Saddik, **Md. Abdur Rahman**, Souhail Abdala, and Bogdan Solomon. PECOLE: P2P Multimedia Collaborative Environment. *International Journal of Multimedia Tools and Applications*, Springer, vol. 39, no. 3, pp. 353-377, August 8, 2007.

Conference Papers

- [1] **Md. Abdur Rahman**, Abdulmajeed Alkhalidi, Jongeun Cha, and Abdulmotaleb El Saddik, “Adding Haptic Feeling to YouTube,” *ACM Multimedia*, Firenze, Italy, October 25-29, 2010.
- [2] **Md. Abdur Rahman**, Sadek Hamdan, Abdulmotaleb El Saddik, and Wail Gueaieb. “Context-Aware Social Networks Mashup: A Personalized Web Perspective,” *IEEE Instrumentation and Measurement Technology Conference (I2MTC 2010)*, pp. 1228 - 1233, Austin, Tx, USA, May 3-6, 2010.
- [3] **Md. Abdur Rahman**, Wail Gueaieb, and Abdulmotaleb El Saddik, “Ubiquitous Social Network Stack for e-Health,” *Medical Measurements and Analysis (MeMeA) 2010*, pp. 57 - 62, Ottawa, Canada, April 30-May 1, 2010.

- [4] **Md. Abdur Rahman**, Abdulmotaleb El Saddik, and Wail Gueaieb, "Data Visualization: Body Sensor Network to Social Networks," *IEEE International Workshop on RObotic and Sensors Environments*, pp. 157-162, Lecco, Italy, November 6-7, 2009.
- [5] **Md. Abdur Rahman**, Abdulmotaleb El Saddik, and Wail Gueaieb, "Data Visualization: From Body Sensor Network to Social Networks," *IEEE International Workshop on RObotic and Sensors Environments*, pp. 157-162, Lecco, Italy, November 6-9, 2009.
- [6] **Md. Abdur Rahman**, Mohammed F. Alhamid, Wail Gueaieb, and Abdulmotaleb El Saddik, "A Framework to Bridge Social Network and Body Sensor Network: an e-Health Perspective," *International Conference on Multimedia and Expo (ICME) 2009*, pp. 1724-1727, NY, USA, June 28-July 3, 2009.
- [7] **Md. Abdur Rahman**, Mohammed F. Alhamid, Wail Gueaieb, and Abdulmotaleb El Saddik, "A Body Sensor Network Toward Ambient Intelligence: an e-Health Perspective," *Medical Measurements and Analysis (MeMeA)*, pp. 22-25, Cetraro, Italy, May 29-30, 2009.
- [8] **Md. Abdur Rahman**, Abdulmotaleb El Saddik, and Wail Gueaieb, "SenseFace: A Sensor Network Overlay for Social Networks," *IEEE Instrumentation and Measurement Technology Conference (I2MTC 2009)*, pp. 1031-1036, Singapore, May 5-7, 2009.
- [9] **Md. Abdur Rahman**, Reza Ghasemaghaei, Abdulmotaleb El Saddik, and Wail Gueaieb, "M-IAR: Biologically Inspired Routing Protocol for Wireless Multimedia Sensor Networks," *IEEE International Instrumentation and Measurement Technology Conference (I2MTC 2008)*, pp. 1823 - 1827, Victoria, British Columbia, Canada, October 12-15, 2008.

- [10] **Md. Abdur Rahman**, Reza GhasemAghaei, Abdulmotaleb El Saddik, and Wail Gueaieb, "Ant Colony-Based Many-to-One Sensory Data Routing in Wireless Sensor Networks," *ACS/IEEE International Workshop on Wireless Internet Services (WISe'08)*, pp. 1005 - 1010, Doha, Qatar, April 1-4, 2008.
- [11] **Md. Abdur Rahman**, Md. Suruz Miah, Wail Gueaieb, and Abdulmotaleb El Saddik, "A Framework for Sensory-based P2P Collaborative Environment," *IEEE Instrumentation and Measurement Technology Conference (IMTC2007)*, pp. 1-5, Warsaw, Poland, May 1-3, 2007.

1.6 THESIS ORGANIZATION

The thesis is organized as follows. In Chapter 2, we first introduce necessary background concepts and terminologies followed by closely-related research works. The related studies have been categorized according to two broad areas: context-aware BSN and social networks. From the studied research works, we further identify some closely-related frameworks and compare them with the proposed framework in this thesis against the requirements discussed in Chapter 1. Chapter 3 presents the mathematical models and algorithms to extract PSNs from the Internet, capture user context from BSN and different services from the Internet, and find context-aware services and COI. Chapter 4 introduces the design of the proposed framework. In this chapter we illustrate three subsystems with the software design of each subsystem. Chapter 5 presents the development of a proof-of-concept implementation of the proposed framework followed by test results presented in Chapter 6. We summarize the thesis contents and contributions and present our future vision of the research work in Chapter 7.

CHAPTER 2: BACKGROUND AND RELATED WORK

2.1 OPEN STACK OF INTERNET

The open stack [OpenStack (2010)] 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. The open stack is shown in Figure 2.1. We now describe each layer of the stack in details.

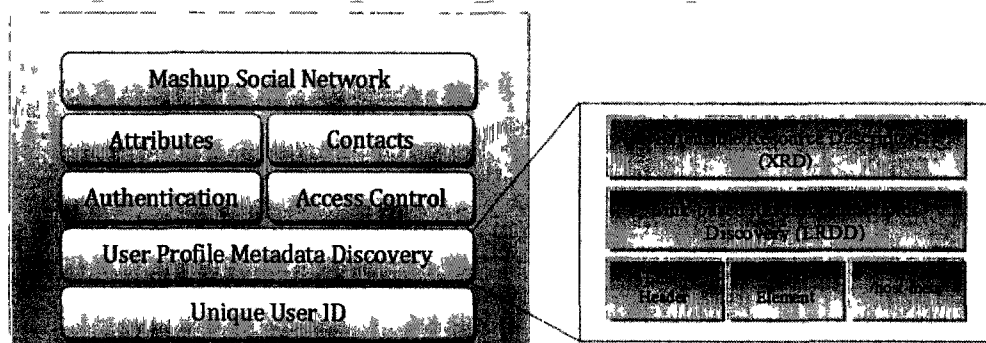


Figure 2.1 Open stack of Internet to extract services and social ties of a person.

2.1.1 Layer 1: Unique User ID

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

2.1.2 Layer 2: User Profile Metadata Discovery

The user profile metadata discovery stack includes three sub-layers as shown in Figure 2.1. 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. XRD¹ uses an XML schema to describe a resource (for example, the profile of a subject and his/her relationships to other linked resources or services like an address book, calendar, identity provider, or list of involved social networks.) LRDD² 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-meta³.

2.1.3 Layer 3-A: Access Control

Access control allows a COI to grant access to a user's private data, such as an address book from one site (the service provider) to another site (consumer site) without revealing the latter's identity. Popular access control mechanisms for the open stack generally include OAuth⁴, which offers two different access control formats: two legged and three legged, where each leg refers to different party involved.

2.1.4 Layer 3-B: Authentication

Authentication layer allows signing in with one's existing social network credentials such as Yahoo, Google, Facebook, Twitter, or any OpenID⁵ 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.

¹ <http://hueniverse.com/2009/03/the-discovery-protocol-stack>

² <http://tools.ietf.org/html/draft-hammer-discovery-03>

³ <http://tools.ietf.org/html/draft-hammer-hostmeta-05>

⁴ OAuth: <http://oauth.net>

⁵ OpenID: <http://openid.net>

2.1.5 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.

2.1.6 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 kept in various proprietary formats (such as Google's GData Contacts API or Yahoo's Address Book API) that are not portable in various social network platforms. Portable contacts provide a common protocol to share address books and other contact data.

2.1.7 Layer 5: Mashup of Social Networks

Social network services are a cloud of services based on Web Services Description Language (WSDL), REpresentational State Transfer (REST), XML/HTTP or database objects. The 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 Wiki pages, blogs, websites, portal servers, emails, faxes or called directly from a smartphone.

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,

the 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.

2.2 COMMUNITY OF INTEREST

A COI is a subset of a user's social ties out of the total set of social ties. The size of COI is of course variable depending not only on the subject but also on the context. For example, the size of COI of a user in the context of a group meeting is different from the size of COI when he/she is teaching a course. The research and development efforts in ubiquitous COI management systems in social networks are still quite immature with only a few solution proposals [Bottazzi, D. et al. (2006)]. Some systems assume that the set of COI members is determined a priori and that only registered members can be invoked [Santana P.C. et al. (2005)], while more advanced solutions provide presence-aware group management [Bottazzi, D. et al. (2003)], [Munoz, M.A. et al. (2003)], [Bardram, J. E. (2005)]. Most of these systems statically determine COI formation, mostly work within indoor environment, and provide the ability to add additional members from one's online community [Munoz, M.A. et al. (2003)], [Bardram, J. E. (2005)].

Different from the many proposed solutions, the COI proposed in this thesis is not statically determined but dynamically created based on additional user context conditions, such as one's location and activities, even in outdoor environments. Many of the currently available solutions tend to provide differentiated group communication facilities. In [Santana P.C. et al. (2005)], authors present a social network service that supports only asynchronous communication among the members of COI. However, some frameworks like [Bottazzi, D. et al. (2003)], [Munoz, M.A. et al. (2003)], and [Bardram, J. E. (2005)] offer synchronous forms of collaboration and provide context-aware mechanisms to identify appropriate COI for information

sharing. In particular, these solutions recognize the possibility of identifying COI on the basis of their roles in collaboration [Bottazzi, D. et al. (2003)], [Munoz, M.A. et al. (2003)] and their currently performed activities [Bottazzi, D. et al. (2003)], [Bardram, J. E. (2005)], or simply on the basis of their attributes [Bottazzi, D. et al. (2003)].

Other COI management solutions have recently emerged in the field of mobile ad-hoc networks that can dynamically form collaboration groups based on the physical proximity [Huang, Q. et al. (2004)], [Luo, J. et al. (2004)]. These solutions provide mechanisms to create, discover and join COI. However, none of them, even though exhibiting potentially promising features, have yet been applied to support context-awareness in an integrated fashion, leaving the burden on service developers to properly handle message forwarding, scheduling and format adaptation according to collaborating entity characteristics, user profile and device properties.

2.3 PERSONAL SOCIAL NETWORKS

Research in personal social networks mostly identifies the problem of creating a personalized network of people of similar interests. For example, Lisa Berkman [Berkman, L. (2000)] has studied the impacts of personal social networks on a patient's health condition. Her research links the influence on health status by the integrated communities and people surrounding the patient. Some social networks and services are even customized for special applications. For example, *hellohealth*⁶ is an online doctors and patients' community for online consultation. Similarly, *Imedix*⁷ is another social network dedicated for health and related topics. Twitter on the other hand facilitates a simple way of communication between a patient and his/her family physician. Using [BB Health Blog (2010)], patients can post their physical developments for their medical doctor just before a visit. Google also is investing its

⁶ <http://hellohealth.com>

⁷ <http://www.imedix.com>

technology in social network applications. For example, Google provides health data API for viewing and sending health data and allows many health applications to access and feed data.

In [Snijders, T. A. B. (2009)], the author has presented a tie and actor-oriented stochastic model called SIENA, which can be implemented as either agent-based or simulation-based social network analysis model. SIENA takes into consideration both endogenous network dynamics and the effects of exogenous variables. SIENA model is suitable for the social network analysis from longitudinal data. A dynamic social network analysis model using Euclidean latent space has been proposed in [Sarkar, P., and Moore, A. W. (2007)], which can scale up even if the number of social ties gets large. Although it uses stochastic processes similar to [Snijders, T. A. B. (2009)], when compared to [Sarkar, P., and Moore, A. W. (2007)] their work shows lower performance and higher algorithmic complexity. Authors in [Wyatt, D. et al. (2008)] have developed a prototype that attempts to uncover social relation patterns by capturing conversational patterns through sensors attached with PDAs. This approach is limited to face-to-face conversation while many of our real-life interactions or social ties are realized via other means such as SMS, email, IM, etc. The research presented in [Ganti, R. K. et al. (2008)] extends the concept of Microsoft sensor map by adding a human presence along with the sensor nodes. The users wearing GPS sensors upload the location information offline to the community portal to share the location information. The work presented in this thesis uses live location tracking while the person is outdoor or indoor.

Rather than measuring the quantity of social relationships, the work presented in [Velardi, P. et al. (2008)] takes into account the semantic concept that ties two persons through their communicative contents. Text mining and clustering is used to extract topics/concept that connects the ties. While the approach presented in this thesis uses the semantic content analysis concept, it also extends the social network

analysis horizon by taking into account the quantity of relationship. Fuzzy set has been used in [Yager, R. R. (2008)] to model social network vocabularies. Similar to the concept adopted in this thesis, social tie formation has been translated from human concept space to social network space using fuzzy logic. K-plex model [Balasundaram, B. (2008)] has been illustrated for finding a subgroup of social networks. Yet its viability in other sources like email, web services and social networks such as Facebook needs to be verified. The authors in [Chao, W. et al. (2007)] propose a local probabilistic method to express the relationship between a co-author pair in combination with topological measures and semantic similarity. Although the approach shows optimistic results for offline co-author dataset, it might not be suitable for online and distributed social network datasets.

2.4 CONTEXT-AWARE SERVICES IN SOCIAL NETWORKS

Capturing user context is an attractive area of research in the social network domain and numerous techniques have been studied in the past [Phung, D. et al. (2009)], [An, J. et al. (2009)], [Kawahara, Y. et al. (2007)], [Shtykh, R.Y., and Qun, J. (2008)]. 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, URL of each service and types of protocols and access mechanism needed for each service). Previous research show that user context can be captured using multimedia information stored in social networks [Joly, A. et al. (2009)], [Velardi, P. et al. (2008)], [Gartrell, C. M. (2010)], [Koolwaaij, J. et al. (2009)], [Gaonkar, S. et al. (2008)], [Bottazzi, D. et al. (2007)].

Authors in [Joly, A. et al. (2009)] studied existing features provided by social networking sites and proposed improved social communication experience by adding context-awareness into the social networking sites. The authors built a prototype called “Meeting Room Assistant,” where social communication is triggered by sharing contextual artefacts such as content or personal information. The proposed prototype leverages context information to assist meeting room activities, such as sharing, and allows users to focus on the communication instead of the sharing process, resulting in increased social awareness and more communication between people while not requiring them to spend more time on social networking sites. The prototype environment enriches the communication of participants without the burden of “social searching”, sending “invites” and requesting “friendship.”

The author in [Gartrell, C. M. (2010)] presents a framework for building social-networking-enabled context-aware services, called *SocialAware*. The framework has been tested with two proof-of-concept applications. The first, called *SocialAwareTunes*, is a context-aware music jukebox player that plays music that reflects the preferences of a group of co-located users. The second application, called *SocialAwareFlicks*, is a context-aware video system that plays movie trailers that reflect the preferences of one or more users watching a video display together. User context and relations between a group of co-users have been studied [Velardi, P. et al. (2008)] in terms of analyzing email, blogs, co-authored papers, and exchanged documents, where the authors present a novel algorithm to cluster users based on content of social network information. Rather than measuring the quantity of social relationship, it takes into account the semantic concept that ties two actors through their communicative contents. Text mining and clustering is used to extract topics/concept that connects the ties. While the approach presented in this thesis uses the semantic content analysis concept, it also extends the social network analysis horizon by taking into account the quantity of relationship.

Authors in [Koolwaaij, J. et al. (2009)] presented a context management framework called Context Watcher that enables mobile phone users to easily and unobtrusively share personal context data such as their location, heart rate, speed or view with members of social networks. In addition to sharing data, the context framework can adapt applications to the contexts such as “meeting with a supervisor”, “is a regular visitor” or “having the best condition of all your friends.” The application leverages the MobiLife context management framework for context discovery, exchange and reasoning. The functionalities provided by the framework are: the ability of knowing where people are located, keeping in touch without having to approach people directly, easy access to services (input parameters automatically provide context parameters like local weather, rich picture submission, public transport information, etc.), remote logging of activities and preferences, and the sharing of such information across different (mobile) applications and websites including Flickr.com and personal blogs.

Another interesting framework is called Micro-Blog [Gaonkar, S. et al. (2008)] and allows smartphone-equipped users to generate and share geo-tagged multimedia blogs. The framework shares data with social network members either through the Internet or a social tie with a smartphone in physical proximity of geo-tagged data. Micro-Blog allows members of social networks to send queries to a set of social ties located at a certain point of interest and receive responses from the interested members of that point. A context-aware middleware, called the Socially Aware and Mobile Architecture (SAMOA), has been proposed in [Bottazzi, D. et al. (2007)], which supports anytime, anywhere semantic and context-aware social networks. SAMOA takes into account two types of a mobile user’s context: location-based context and user profile-based context. SAMOA tries to decouple social network management functionalities from application requirements by integrating a set of common management facilities for personalizing location-dependent social networks and for

propagating social networks' visibility up to the application level. SAMOA can dynamically update services based on location and profile context.

2.5 CONTEXT-AWARE SENSOR FRAMEWORKS

Some of the existing frameworks employ BSN which consists of sensors to measure human body phenomena such as pressure, humidity, temperature, ECG and stress level to capture contextual information. The BSN helps to find the effects and causes that may have triggered any physical disorder. Researchers use these sensors for continuously sensing events, monitoring patients, and reminding drug and tele-monitoring needs. The data collected by those sensors is kept and stored for further analysis and used by medical research centers.

Numerous sensor network platforms have been introduced in the literature that might be suitable for the scenario mentioned in this thesis. The BTnode platform is an autonomous, versatile and flexible platform for rapid prototyping of heterogeneous and mobile wireless sensor networks [Beutel, J. et al. (2003)]. BTnode has been successfully implemented in many applications including remote product monitoring via mobile phone, wearable and ubiquitous computing demo applications, user interactions with cell phones, sensor networks research and local positioning [Beutel, J. et al. (2003)]. BTnode is suitable for applications that tolerate the sensor nodes in sleep mode until events are received from the Bluetooth radio or the sensors.

The Tmote series wireless sensor platform such as the Tmote Mini, Tmote Sky and Tmote Invent have been implemented in several mobile sensing test beds including the BikeNet project that represents the mobile networked sensing system for bikes [Eisenman, S. B. et al. (2007)], a medical sensor network that acts as a human Brain Computer Interface [Sarikaya, S. et al. (2006)], and an online person tracking system [An, X., and J. Wang (2006)]. iBadge [Park, S. et al. (2002)] is a lightweight power-aware sensor badge wearable by a human being which can capture human

interactions and send the captured data wirelessly to a base station. The Wireless Body Sensor Network proposed in [Aziz, O. et al. (2007)] comprises of tiny wireless sensor nodes that can be comfortably carried by any human, such as an older patient.

A wireless sensor node has been presented in [Bhardwaj, S. et al. (2008)], which can monitor several activities of elderly people such as detection of arrhythmia disease, norm calculation, orientation calculation and fall detection using multiple sensors such as ECG and accelerometer. The sensor node is capable of sending the live sensory data to a server where it is analyzed for possible health risks. If needed, the server can send the data to the doctor's PDA.

SHIMMER is a sensor node architected by Intel Digital Health Group which integrates computation, radio communication, high-fidelity tri-axial sensors and a large flash memory into a tiny, wearable enclosure. As a primary means of radio communication, it uses an IEEE 802.15.4 2.4GHz radio and a Bluetooth radio to facilitate streaming of sensory data at high rates. This sensor node has been actively used by several applications of assessing patients with neurological disorders [Lorincz, K. et al. (2007)]. The TinyNode 584 is an ultra-low power sensor node that supports data rates up to 153 KB/s and a range up to 300 meters. Using an external GPRS board that can run Java programs, the node can be remotely accessed from the Internet. TinyNode has been used for geo-monitor systems for harsh environments such as in [Talzi I. et al. (2007)].

BeTelGeuse [Nurmi, P. et al. (2010)] is a tool for Bluetooth data gathering. It turns a standard mobile device such as a cellular phone into a relay node which gathers data from a BSN over Bluetooth, and forwards it to a remote server over a mobile data service such as GPRS. A Medical Embedded Device for Individualized Care (MEDIC) [Wu, W. et al. (2008)] is a wearable sensor platform, intended for capturing and diagnosing remote patients' live physiological and contextual information. The

MEDIC prototype has been developed based on standard PDA and wireless sensor nodes with Bluetooth radio.

ZebraNet [Zhang, P. et al. (2004)] sensor nodes are specialized in tracking wildlife animals. The sensor node can be attached to the animal as a collar that has a GPS sensor to facilitate the online tracking of the subject. A mobile wireless sensor/actuator network to control bull activity during a breeding period has been shown in [Wark, T. et al. (2007)]. In [Hull, B. et al. (2006)], [Eisenman, S. B. et al. (2007)], a prototype of a mobile sensor network has been developed under the name of carTel. carTel facilitates collecting and processing sensory data locally from a human body and then delivering and visualizing it in remote locations in real time or with a delay, based on the availability of the delivery networks. Tracking human mobility using Bluetooth devices is shown in [Nordström, E. et al. (2007)]. sChat is a mobile WSN based chatting system where a Mobile-PC attached with Telos motes carried by mobile users can send a group of chat messages over a WSN of Tmote Sky motes [Sun, F. et al. (2007)].

BSN has been employed to develop different social network applications as well. An end-to-end patient monitoring system intended for Intensive Care Unit using ISO/IEEE 11073 standard has been presented in [Yao, J., and Warren, S. (2005)]. Plug and play sensor network captures the sensory data from a patient's body and then by using Bluetooth radio sends the information to the nearest monitoring server. The server in turn transforms the data to the end user standard (in this case, European Electronic Healthcare Record standard called EN13606) and sends it to a remote patient database. Using the ISO/IEEE 11073 standard several applications of home health monitoring systems have been implemented that use wearable sensor nodes [Yao, J., and Warren, S. (2005)], [Schmitt, L. et al. (2007)]. The framework designed by [Krcó, S. (2004)] incorporates different types of sensors that communicate with a PDA using Bluetooth. The framework enables remote users to access the sensor

network and query the sensory data via GPRS. The PDA acts as a control node which coordinates and stores data from various sensors, and transmits to remote health providers using an XML based protocol.

Another health monitoring model is designed by [Yang, B. H., and Rhee, S. (2000)] where a ring sensor is used for healthcare automation. The ring can be worn by a patient for monitoring pulse waves and blood oxygen saturation. The data collected by the ring sensor is transmitted to a home computer where the actual processing takes place. The ring is capable of detecting emergency situations since it is enriched with location estimation and activity recognition algorithms. A patient monitoring service was developed using a Wireless Application Protocol (WAP) based telemedicine system [Hung, K., and Zhang, Y. T. (2003)], enabling authorized personnel to inquire and retrieve a patient's vital physiological data such as blood pressure and ECG diagrams in Wireless Markup Language (WML) format via Bluetooth.

The system, proposed by [Noury, N. et al. (2000)] and [Rialle, V. et al. (2000)], is designed to address the falling issue of a patient, where a detection mechanism is used with infrared sensors and magnetic contact switches. The system consists of wireless sensors installed in a home that communicate wirelessly with in-home software running on a personal computer for the detection of emergency situations. The emergency situations considered are sleeping, lying after a fall, walking, etc.

The work presented by [Gao, T. et al. (2005)] tends to improve patient monitoring and response tracking. The system coordinates the communication among different entities of a health center, including medical professionals in a hospital, and other specialists available online through the system. A central web portal is used to enlist a patient's health records augmented with location information, and remote users can access patient information in real-time. The system also provides historical sensory information once the patient is taken to the emergency.

Other research by [Jovanov, E. et al. (2003)] facilitates monitoring a patient's stress level using a distributed wireless intelligent sensor system. The system consists of a body area network which uses different sensors attached to a patient's body and a PDA as a gateway between sensors and end-point terminals on the Internet. This system can be used for physiological evaluation of military members in training.

Authors in [Wyatt, D. et al. (2008)] have developed a prototype that attempts to find social relationships by capturing conversational patterns through sensors attached with PDAs. The research presented in [Ganti, R. K. et al. (2008)] extends the concept of a Microsoft sensor map by adding human presence along with the sensor nodes. Users wearing GPS sensor upload the location information offline to a community portal to share their location information. Other related research has shown the potential of involving social networks in e-Health applications. For example, [Smith, K. P., and Christakis, N. A. (2008)] have studied the impact of interconnection between health specialists and patients' social networks. Results show the trends of Web 2.0 developments and provide recommendations to leverage social networks.

Sensors provide rich sources of user context and have been used extensively by researchers in the past [Upendra, R., and Max, O. (2008)], [Teixeira, T. et al. (2009)]. Most research projects leverage the context of users in order to adapt the interaction between a user and a computing system [Christopoulou, E. (2008)], or for the system to proactively make decisions intended to support the user [Wang, X. et al. (2004)]. Such applications include Intelligent Meeting Rooms [Leong, L. H. et al. (2005)], Smart Homes [Gu, T. et al. (2004)], personalized mobile advertisement [Zhdanova, A. V et al. (2006)] and electronic healthcare [Broens, T. et al. (2007)]. In our case, we consider context as sampled information about people's environment and actions in time, and information contained in the multimedia information within the social network space. We propose to leverage this context information to give more

opportunities for a user to consume social network services and to communicate with relevant COI.

2.6 COMPARISON WITH RELATED WORKS

Tables 2.1 and 2.2 show the summary of a comparative study among the closely related works with SenseFace. The comparative study only considers those works that can be considered as a framework. We have logically divided the comparative study based on two high-level concepts: context-related metrics and framework-related metrics. Context-related metrics are those shown in Table 2.1 while framework-related metrics are shown in Table 2.2. Next we briefly describe each metric and how we define the metrics. An empty cell means the feature is not supported by the framework and a “Y” means the feature is supported.

BSN Support identifies whether a framework supports body sensor network. *Heterogeneous Service Support* defines whether the framework can extract content from diversified services or not. *Context Support* tells whether the framework supports only single/intra-user context or multi/inter-user context or both. Intra-user context refers to the change in one particular user’s context throughout a day while Inter-user context refers to the difference in contexts between any two or more number of users at any given moment. This originates from the fact that every user needs to access different services or communicate with different categories of people during different period of a day. *Source of Context* enlists from where the source is extracted, with possible values BSN and/or heterogeneous services. If a framework supports multiple services, the *Dynamic Service Adaptation* metric then states whether the services are rendered dynamically or pre-defined. Finally, the COI Support metric identifies whether the framework statically defines the social ties or whether it is dynamic.

Table 2.1: Comparison summary based on context related metrics.

Framework /System	BSN Support	Heterogeneous Service Support	Context Support (Intra/Inter User)	Source of Context	Dynamic Service Adaptation
SenseFace	Y	Y	Y (both)	BSN and services	Y
Gao, T et al. (2005)	Y		Y (Intra)	BSN	Y
Wu, W. et al. (2008)	Y		Y (Intra)	BSN	
Nurmi, P et al (2010)	Y		Y (Intra)	BSN	
Bhardwaj, S. et al. (2008)	Y		Y (Intra)	BSN	
Eisenman, S. B et al. (2007)	Y		Y (both)	BSN	
Gaonkar, S. et al. (2008)	Y		Y (Inter)	BSN	Y
Koolwaan, J et al (2009)	Y	Y	Y (Inter)	BSN	
Bottazzi, D. et al. (2006)	Y		Y (Inter)	Services	
Santana P C et al (2005)			Y (Inter)	Services	
Munoz, M.A. et al. (2003)			Y (Inter)	BSN	
Bardram, J E (2005)			Y (Inter)	Services	
Wynt, D. et al. (2008)	Y		Y (Inter)	BSN	
Ganti, R. K. et al (2008)	Y		Y (Intra)	BSN	
Phung, D. et al. (2009)	Y		Y (Intra)	BSN	
An, J et al (2009)	Y	Y	Y (Intra)	BSN	Y
Kawahara, Y. et al. (2007)	Y		Y (Intra)	BSN	
Joly, A et al. (2009)			Y (Inter)	Services	
Gartrell, C. M. (2010)			Y (both)	Services	

As shown in Table 2.2, some frameworks are limited to operate within an indoor environment, some work outdoors and some support both. These features are captured in the *Environment* column. *Communication Services with COI* identifies whether the frameworks facilitate one or more means of communication among one's social ties. *Direction of Communication* is classified as downstream or upstream. If the framework supports BSN, upstream communication refers to sending sensory data from BSN to social networks. Downstream refers to sending sensory data from social

networks to BSN. If a framework does not support BSN, upstream communication refers to sending information from a person to his/her social ties and downstream communication refers to sending information from social ties to a person.

In addition to supporting Internet-based services, some frameworks also support existing social networks such as Facebook, Twitter and MySpace. This is captured by the *Social Network Support* column. *Mobility Support* refers to the ability to provide service even when a user is on the mobile. Whether a framework supports all the layers of open stack or a subset is identified by *Open Stack Support*. If a framework uses open source services and provides open source code, we call it *Open Source*. Finally, if a framework embeds privacy control system, we refer it by *Privacy Support*.

Table 2.2: Comparison summary based on framework related metrics.

Framework /System	Environment (Indoor/ Outdoor)	Communication Services with COI	Direction of communication (Upstream/Downstream)	Social Network Support	Mobility Support	Open Stack Support	Open Source	Privacy Support
SenseSpace	Both	IM, SMS, MMS, Email, Fax, RSS, APNS, voice mail, Chat, video	Both	Y	Y	Y(Full)	Y	No (out of Scope)
Gao T et al. (2005)	Both	PSTN IP video conferencing, IP telephony	Upstream		Y			
Wu, W. et al. (2008)	Both	Not mentioned	Both		Y			
Nurmi P et al (2010)	Outdoor	Not mentioned	Upstream		Y		Y	
Bhardwaj, S. et al. (2008)	Indoor	Text message	Upstream		Y			
Eisenman S B et al. (2007)	Outdoor	Audio, SMS,	Both		Y			
Caoniar, S. et al. (2008)	Both	Blog, text message	Both		Y		Y	Y
Koolwaal J et al (2009)		IM, Chat	Upstream	Y	Y		Y	
Botazzi, D. et al. (2006)	Outdoor	Voice command, sound signal, multicast text message	Both		Y			
Santana P C et al. (2005)		BLOG, RSS, Voice command IM	Both	Y				
Munoz, M.A. et al. (2003)	Indoor	IM	Upstream		Y			
Bardram J E (2005)	Indoor	Tele-pomter PSTN, Chat	Both		Y			
Wyatt, D. et al. (2008)		Speech	Upstream	Y	Y			Y
Ganti R K. et al (2008)	Outdoor	Offline data uploading	Both	Y	Y			Y
Phung, D. et al. (2009)	Indoor		Upstream		Y			
An J et al. (2009)	Both	Voice	Upstream		Y			
Kawahara, Y. et al. (2007)	Both		Upstream		Y			
Joly A et al (2009)	Indoor	Email	Both	Y		Y(Partial)		Y
Garrett, C. M. (2010)	Indoor		Upstream	Y		Y(Partial)		No (out of Scope)

2.7 SUMMARY

In this chapter we have first elaborated the open stack which is leveraged by the proposed framework to extract content from heterogeneous and proprietary services. We then summarized our findings of the closely-related research frameworks and prototypes. We took lessons from those research works to model and design the proposed SenseFace framework. We studied the strengths and weaknesses of each work by keeping the requirements of context-aware social networks in front. Although many frameworks show partial potentials, they lack features in providing complete support of a BSN and Internet-based context-aware system that can dynamically adjust the COI and services. Finally, we have provided a comparative study among 19 frameworks with respect to 13 features which show that the proposed framework excels other existing works in almost all the avenues.

CHAPTER 3: MODELING UBIQUITY AND CONTEXT-AWARENESS IN PERSONAL SOCIAL NETWORKS

In this chapter we illustrate modeling of the context-aware social network framework. The purpose of such modeling is to capture important entities, different attributes and dimensions of each entity and relationships between or among them so that the framework can be realized by appropriate algorithms, computing systems and human logic. The modeling can be broadly categorized into three logical units. The first unit is to extract two important classes of information: 1) who are the social ties of a subject, which is referred to as personal social network, and 2) which services does a subject consume or subscribe to. Once the social ties and services have been modeled using mathematical expressions, the next unit of modeling is to define user context.

We first describe a context topology that helps us in defining primitive user contexts and then we show how to model complex and compound context expressions. Context modeling has been categorized according to two context sources: body sensor network and Internet-based services. The final stage of modeling is responsible for mapping each user context with a subset of social ties and a subset of services.

To keep consistency, we use the same scenario described in Section 1.2 where a person named Alice interacts with different members of her social network throughout a day. Alice wears several sensors that keep track of her physiological phenomena, activities and ambient condition. The sensors are paired with her smartphone that has ubiquitous access to the Internet. Alice uses several Internet-based services to communicate, consume and share information with the members of her personal social networks. The overall modeling process is summarized in Figure 3.1.

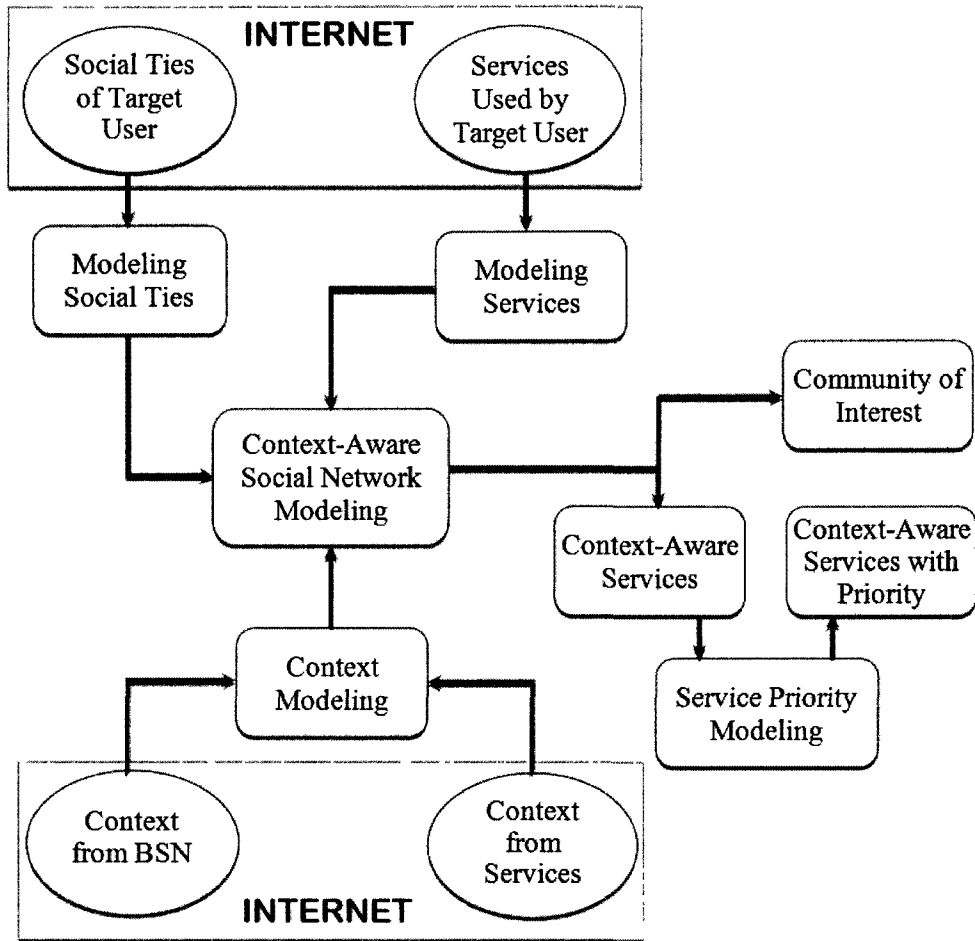


Figure 3.1 Different entities involved in modeling context-aware social networks.

3.1 PERSONALIZED SOCIAL NETWORKS AND SERVICES

We assume that Alice either communicates with each member through a service or that the relationship between Alice and each social tie can be captured through a service. For example, the face-to-face meeting with Alice and her family physician can be captured or related through the event reminder service such as online calendar or smartphone calendar. Similarly, social ties information can be obtained

from sources such as email and IM clients, social portals such as Facebook, smartphone contact lists, etc.

3.1.1 Deriving Personal Social Networks from Heterogeneous Sources

We assume that a user (U) has total n number of social ties, which is represented as a set $T(U)$. This can be represented as:

$$T(U) = \{t_1, t_2, t_3, t_4, \dots, t_n\} \quad (1)$$

Where each tie is represented as $t_i, 1 \leq i \leq n$. According to [Killworth, P. et al. (1990)], the value of n might be about 1,500 on an average. $T(U)$ can be represented by the following 1 by n social tie matrix:

$$[t_1 \ t_2 \ t_3 \ t_4 \ \dots \ t_n]$$

Each member of the set $T(U)$ might be decentralized in m number of diversified social tie sources such as smartphone personal information manager (PIM) database, POP or IMAP based email clients, social portals such as Facebook, Twitter, and blogs, instant message clients and so on. Hence, when we extract each member's information (t_i from diversified sources), equation (1) has to be adjusted as following:

$$T^*(U) = \{T_1^*, T_2^*, T_3^*, T_4^*, \dots, T_m^*\} \quad (2)$$

Where each T_j^* is a set of social ties of the subject U extracted from the j -th service and m is total number of diversified and unique services through which the subject communicates with the social ties. If a subject has subscribed to 4 services, then some examples of the subset T_j^* might be following:

$T_1^* = \text{social ties developed through video conferencing client such as Skype}$

$T_2^* = \text{social ties developed through social portal such as Facebook}$

$T_3^* = \text{social ties developed through smart phone contact list}$

$T_4^* = \text{social ties developed through email client}$

Due to recent trends of multimedia information sharing through different Internet-based services, a subject has varied options of communicating with a member of his/her social ties. For example, a particular social tie might be connected through video conferencing service, online document sharing service, email client IM service, etc. This fact has been pointed out in Figure 3.2.

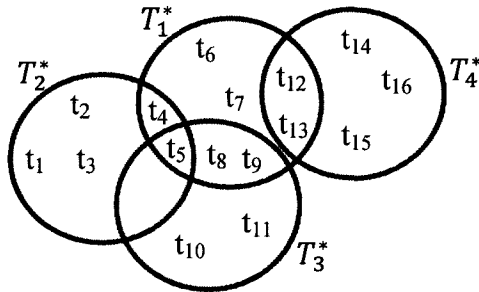


Figure 3.2 A generic Venn diagram showing overlapping multiple sources of social ties where each social tie information might be extracted from more than one source.

Each element of each subset of T_j^* is actually a member of a social tie as presented in Equation 1. The purpose of grouping each member of a social tie of Equation 1 is that a subject communicates with each member of social ties through different communication media (email, fax, SMS, MMS, cell phone, etc.). Equation 2 represents each social tie through one or more of these services. Hence, each social tie from Equation 1 might fall into one or more of the subsets of Equation 2. For example, a subject might have communication with a member of a social tie through more than one channel, which results in the following relation between Equations 1 and 2:

$$\sum_{j=1}^m |T_j^*| \geq |T(U)| \quad (3)$$

Semantic modeling of a community of interest based on human understandable metric such as user context requires each member of a social tie to be represented by

Equation 2 to further categorize according to some social factors such as family, friends, professional, sports, etc. In summary, there exists a vocabulary of social categories, each of which represents a subset of social ties coming from the social tie space of $T^*(U)$. We define this new, higher layer set of social ties as $T^{**}(U)$, which can be defined as the following:

$$T^{**}(U) = \{T_1^{**}, T_2^{**}, T_3^{**}, T_4^{**}, \dots, T_{sc}^{**}\} \quad (4)$$

Where sc is the total number of different social categories of a user.

Similar to equation (3), the relationship between the sets $T^{**}(U)$ and $T(U)$ is as follows:

$$\sum_{j=1}^{sc} |T_j^{**}| \geq |T(U)| \quad (5)$$

3.1.2 Extracting Internet-based Services from Heterogeneous Sources

We assume that the subject has subscribed to diversified types of services, whose total set can be defined as S . Through these services, the subject either consumes information or shares information with a subset of his/her social ties. The set of services S is divided into two mutually exclusive subsets. The first subset is called community oriented services (COS) whose upper bound is m , as shown in Equation 2. COS are those services in which the subject and each social tie is connected through one or more services, for example a video conferencing service, which involves the subject and a subset of social ties. The COS can be expressed as follows:

$$COS(U) = \{s_1, s_2, s_3, s_4, \dots, s_m\} \quad (6)$$

Where $s_i | \{1 \leq i \leq m\}$ is an instance of service which connects a subject with a subset of his/her social ties. The other subsets of services are those where interaction only takes place between a subject and a particular service. No member of any social tie is involved, and these are called services without social ties (SWST). For example,

sensor-assisted services provided by BSN and Internet-based services including a subject checking a flight status, current weather, bus schedule, driving direction on a map, currency exchange rate, etc. We assume that the total number of such SWST is p . The SWST can then be expressed as following:

$$SWST(U) = \{s'_1, s'_2, s'_3, s'_4 \dots, s'_p\} \quad (7)$$

For the simplicity of the modeling, we assume that the members of the COS and the SWST are mutually exclusive. In this case, the following conditions are then true:

$$COS(U) \cap SWST(U) = \emptyset \quad (8)$$

Total space of services S can then be expressed using the following expression:

$$S = COS(U) \cup SWST(U) \quad (9)$$

$$|S| = (m + p) \quad (10)$$

3.1.3 Composite Services and Social Tie Modeling

As discussed before, the service categories defined by COS also involve members of social ties. Because the upper bounds of both Equation 2 and Equation 6 are same, for example m , we can define the subset of social ties related to each service by making the following relation:

$$COS'(U) = \{ (s_i, T_i^*) | s_i \in COS(U), T_i^* \in T^*(U) \} \quad (11)$$

$$SWST'(U) = \{ (s'_j, Y) | s'_j \in SWST(U), Y = \emptyset \} \quad (12)$$

where s_i is a scalar variable representing a unique service defined in equation (6) and T_i^* is a vector quantity representing a subset of social ties related to that service defined in equation (2). Hence the definition of resulting composite service is as following:

$$S' = COS'(U) \cup SWST'(U), \text{ where } |S'| = m + p \quad (13)$$

For the sake of simplicity, we assume that equation (13) is re-written as follows:

$$S = \{s_n \in S'\} \quad (14)$$

Equation 14 provides the information about each service bundled with social ties, which is used to be mapped with extracted context to find context-aware services and the community of interest. If we want to represent categorized dataset of composite services and social ties, which is called the personal social network (PSN), we leverage Equation 4 and 11 as follows:

$$COS''(U) = \{ (s_i, T_j^{++}) | \forall i, j: s_i \in COS(U), T_j^{++} \subseteq T_j^{**}(U) \} \quad (15)$$

The above equation conveys very important characteristics of the PSN of a subject, which helps in relating a subset of social ties of a particular category to higher layer concepts such as user context.

3.2 MODELING USER CONTEXT

In this section we focus on context modeling without making any assumptions on the underlying domains of context. Instead, the context model defines an approach which allows the system designer or developers to express their domain-dependent contexts. The purpose of context modeling is to try to capture different states of user context and represent them using some generic mathematical expressions. It represents information that is important for the user to know if the ubiquitous environment senses them and, in some situations, to control them. User context can be expressed as a collection of user information that can be obtained from diversified Internet and ad-hoc sources that can be used to characterize the user's ambient, personal, physiological and consumed service state. In this literature, user context is captured from two different sources: body sensor network and Internet-based services [Rahman,

Md. A. et al. (2010b)]. Context model leverages both the service modeling and social tie modeling. Context modeling uses Equations 1 to 15.

Table 3.1: Context schema definition.

<i>Context Definition</i>	Description
<i>context-ID</i>	Unique ID of the context.
<i>context-Name</i>	Name of the context.
<i>context-CreationDate</i>	Date of creation of the context model.
<i>context-LastModified</i>	Date of last modification of the context model.
<i>context-URI</i>	Location where the context schema is stored.
<i>context-Creator</i>	Human or non-human subject who created the context.
<i>context-Events</i>	Event(s) that should trigger the context. In this modeling, sensors and live event information coming from diversified sources of Internet are the event sources.
<i>context-Location</i>	Physical location(s) related to the context.
<i>context-Objective</i>	Purpose of the context.
<i>context-Services</i>	Services/applications/resources necessary or related to realize the context. For example, communication devices (smartphone) and sensors, applications, software agents, and network resources (bandwidth, data rate).
<i>context-SocialTies</i>	Members of one's social ties involved in the context.
<i>context-AccessControl</i>	Information or parameters needed to access different services, events and social tie information from the Internet, BSN or other private sources defined at the current context. Privacy or security issues are also part of this parameter.
<i>context-Related</i>	Related/parent/child contexts.
<i>context-Quality</i>	Precision, probability of correctness, trustworthiness, resolution and up-to-date.

We first define necessary variables of user context to be filled up during different stages of context modeling. This serves as a context vocabulary or schema that can be used to verify the completeness of a unique context. Table 3.1 provides different context parameters that together express a complete definition of user context. Modeling user context is an attractive area of research and numerous techniques have been studied in the past [Baldauf, M. et al. (2007)], [Bolchini, C. et al. (2007)]. Similar to the recommendations found in [Preuveneers, D. et al. (2004)] and [Kaenampornpan, M. and E. O'Neill (2004)], 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, weekend, vacation, etc.), user activities (walking, sleeping, lying, talking, running, driving, etc.), user physiological information (heartbeat, blood pressure, etc.), to-do lists, and types of end-device used, to name a few.
- Social context data including contact lists, 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 such as a subject's past context information stored in a database similar to user profile or resource profile.
- 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 periods of a day.

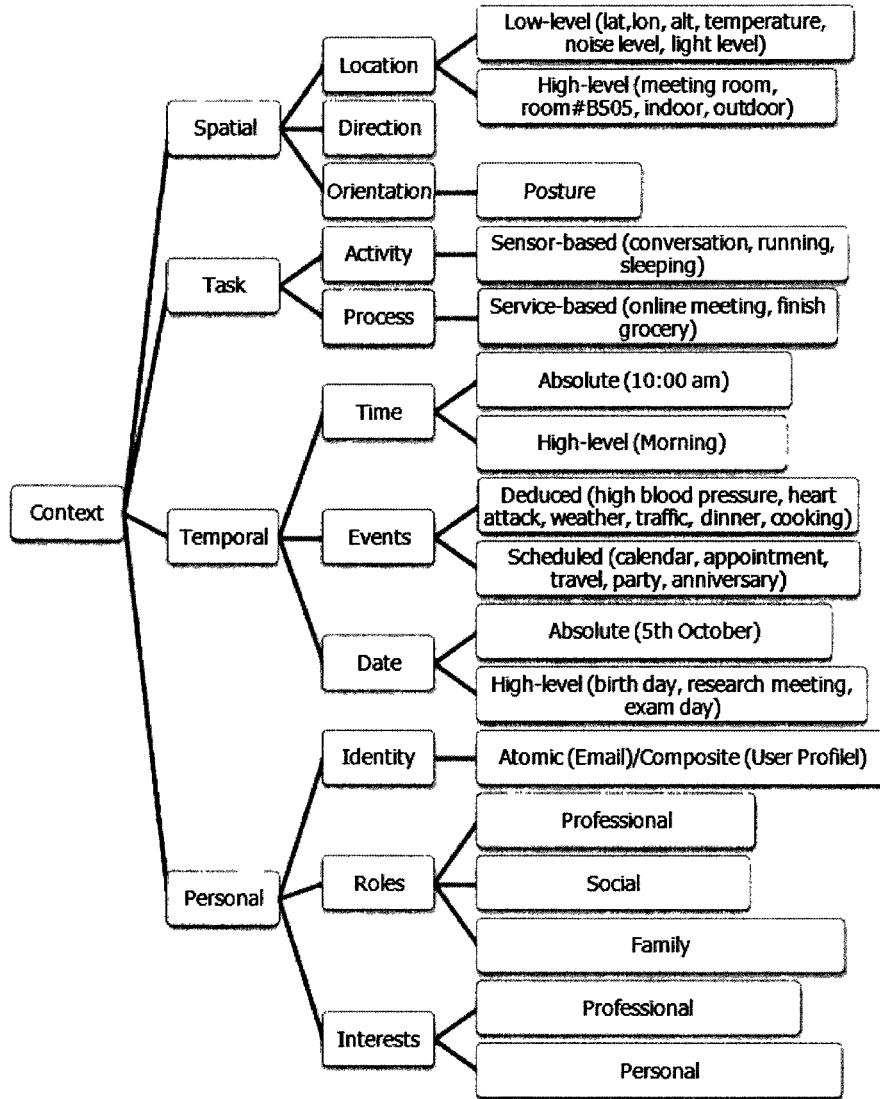


Figure 3.3 Context topology for modeling different levels of context primitives.

To simplify the modeling steps, we define a tree-based context topology shown in Figure 3.3 that helps in expressing different combinations of context expressions using four context categories. These four categories of contexts are spatial, temporal, task-based and personal. The next level, like the children of each category, helps in defining fine-grained and low-level user context. For example, the spatial context

might fall into one of the three sub-categories, namely the location, direction and orientation. The third level of user context actually deals with a primitive class of user context. For example, in the case of location-based context, user location can be expressed in high level notation such as: Room Number 2016 of School of Information Technology and Engineering of the University of Ottawa, ON, Canada, or it might be low-level such as in terms of latitude, longitude pair. Each path through one of the leaves of the topology tree leads to one unique expression of user context. Now we formally express the user context modeling technique. We express the set of context primitives as the following:

$$C = \{c_1, c_2, c_3, \dots, c_l\} \quad (16)$$

where C is the set of all context primitives in terms of vocabulary from one of the four categories such as spatial, task, temporal and personal. Each granular context expression from set C is formed using the tree shown in Figure 3.3. For example, let us consider four different contexts of a user such as *sleeping at night*, *taking a shower*, *cooking at kitchen*, and *engaged in meeting*. Using the topology tree, we can break down each of the four high-level contexts into smaller chunks of primitive contexts as shown below. The following breakdown of primitives might vary from subject to subject.

$c_1 =$ SUBJECT LocatedIn BEDROOM [Spatial Context]

$c_2 =$ BEDROOM LightLevel LOW[Spatial Context]

$c_3 =$ TIME is MIDNIGHT[Temporal Context]

If (c_1) and (c_2) and (c_3) Then

Context = *sleeping at night*

$c_4 =$ SUBJECT LocatedIn WASHROOM [Spatial Context]

$c_5 =$ WATERHEATER LocatedIn WASHROOM [Spatial Context]

$c_6 =$ WASHROOM DoorStatus CLOSED [Task Context]

$c_7 =$ WATERHEATER Status ON [Task Context]

If (c_4) and (c_5) and (c_6) and (c_7) Then

Context = *taking a shower*

$c_8 = \text{SUBJECT LocatedIn KITCHEN [Spatial Context]}$
 $c_9 = \text{ELECTRICOVEN LocatedIn KITCHEN [Spatial Context]}$
 $c_{10} = \text{ELECTRICOVEN Status ON [Task Context]}$
 If (c_8) and (c_9) and (c_{10}) Then
 Context = *cooking at kitchen*

$c_{11} = \text{SUBJECT LocatedIn MEETINGROOM [Spatial Context]}$
 $c_{12} = \text{SMARTPHONE LocatedIn MEETINGROOM [Spatial Context]}$
 $c_{13} = \text{SMARTPHONE Status ON [Task Context]}$
 $c_{14} = \text{CALENDAR AlertMessage ACTIVE [Temporal Context]}$
 If (c_{11}) and (c_{12}) and (c_{13}) and (c_{14}) Then
 Context = *engaged in meeting*

Similarly, each element of the set C is called specific context because it represents one single and unique aspect of the context definition. The context primitives are bounded by a universe of discourse, for example, defined by a vocabulary of context. More than one specific context can be happening at the same time such as time-related context units: time::afternoon, time::meeting, etc. and location related context units: location::work, location::school, location::lab, location::(latitude, longitude), etc. Parameters of Equation 16 do not indicate which categories of context they belong to. In order to express categorized contexts, Equation 16 can be re-written as follows:

$$C' = \{C'_1, C'_2, C'_3, C'_4\} \quad (17)$$

where C' is the set of context sets classified according to each category, $C'_1 =$ set of spatial context primitives, $C'_2 =$ set of task-related context primitives, $C'_3 =$ set of temporal context primitives and $C'_4 =$ set of personal context primitives. The following expressions define the relation between different variables of Equations 16 and 17:

$$C'_i \subseteq C \quad (18)$$

$$\bigcup_{i=1}^4 C'_i = C \quad (19)$$

Once the atomic context primitives have been identified, the next task in modeling is to define a unique and complete context expression, which is performed as follows:

$$C^* = \{c_1^*, c_2^*, \dots, c_n^*\} \quad (20)$$

where c_i^* is formed using a combination of primitive contexts chosen from one or more elements from each of C_1'' , C_2'' , C_3'' and C_4'' . However, it is not a necessary condition that at least one context primitive has to be present from each context category. Sometimes one or more context categories might be absent within the expression of c_i^* , depending on the type of context. Sometimes, one or more context primitives might be chosen from a single category. For example, the context *engaged in meeting* contains two spatial context primitives from C_1' ,: one from C_2' , one from C_3' and none from C_4' . A particular element of C^* i.e. c_i^* is a complete and unique context definition such as *subject is engaged in teaching at the university*. Although Equations 16-20 mostly deal with high-level user expressions, the actual sources of events that will trigger an existing high level context-primitive are obtained from sensors or services. Hence, we express the user context in terms of low level context primitives as follows:

$$C'' = \{C_s, C_{sc}\} \quad (21)$$

where C'' is the set of contexts classified according to their source, such as whether the context is captured from sensory data or from social network services, C_s is the set of context primitives that is triggered by the sensory data and C_{sc} is the set of context primitives that is triggered by the multimedia information contained in diversified social networks. The relationship between C'' and C can be expressed as $C_s \cup C_{sc} = C$. We can expand Equation 21 using context primitives based on the source of each context using the following expressions:

$$C_s = \{c_{s1}, c_{s2}, c_{s3}, \dots, c_{sk}\} \quad (22)$$

where $c_{si} \in C$ and $C_s \subset C$.

$$C_{sc} = \{c_{sc1}, c_{sc2}, c_{sc3}, \dots, c_{scj}\} \quad (23)$$

where $c_{scj} \in C$ and $C_{sc} \subset C$ and $C'' = C_s \cup C_{sc}$.

3.2.1 Extracting User Context from Body Sensor Networks

We assume that a user/subject uses d number of sensors, either wearable on the human body or deployed to surrounding environment to observe a particular phenomenon. We express the set of sensors as the following:

$$SEN = \{sen_1, sen_2, sen_3, \dots, sen_d\} \quad (24)$$

Each sensor can be a source of several context primitives. For example, the sensor can be used to define locations such as home, lab, shopping mall, university cafeteria, etc. If a person has d number of wearable sensors, then the problem would be to map each sensory data type with a context primitive of set C_s as following:

For any instance of context $c \in C_s$ and $sen \in SEN$ (derived from Equations 22 and 24 respectively), we define some one-to-one function f on sen into c , $f: sen \rightarrow c$ so the following relation holds:

$$C_s(c_{si}) = \{(c_{si}) \mid \exists sen_i \in SEN, f(sen_i) = c_{si}\} \quad (25)$$

This will help high-level software components that capture actual sensory data to manage high-level user context base.

3.2.2 Extracting User Context from Heterogeneous Services

For any composite service $s \in S'$ and any instance of context $c \in C^*$ (derived from Equations 14 and 20 respectively) we define some one-to-one function f' on s into c , $f': s \rightarrow c$. With respect to a particular context, $c_{scj} \in C^*$, a set of services is defined as:

$$C_{sc}(c_{scj}) = \{(c_{scj}) \mid \exists s_i \in S', f'(s_i) = c_{scj}\} \quad (26)$$

3.3 MAPPING USER CONTEXT TO COMMUNITY OF INTEREST AND SERVICES

3.3.1 Dynamic Context-Aware Services

For any composite service $s \in S'$ and any instance of context $c \in C^*$ We define some one-to-one function g on s into c , $g: s \rightarrow c$. With respect to a particular context, $c_i^* \in C^*$, a set of services is defined as:

$$CS(c_i^*) = \{(s_i, T_i^*) | s_i \in S', T_i^* \subseteq T^*(U), \forall s_i: g(s_i) = c_i^*\} \quad (27)$$

Equation 27 maps each complete context from the set of C^* with a subset of services from S' . Therefore, a set mapping each instance of context with a set of services and associated social ties is defined as:

$$CS = \{CS(c_i^*) | c_i^* \in C^*\} \quad (28)$$

Because the services might be coming from either sensory data or rest of the services, we expand the definition of context-aware services based on sensory data as follows:

$$CS(c_{si}^*) = \{(s_i, T_i^*) | s_i \in S', T_i^* \subseteq T^*(U), \forall s_i: g(s_i) = c_{si}^*\} \quad (29)$$

$$CS = \{CS(c_{si}^*) | c_{si}^* \in C^*\} \quad (30)$$

Similar expressions of context from social networks can be deduced as the following:

$$CS(c_{scj}^*) = \{(s_i, T_i^*) | s_i \in S', T_i^* \subseteq T^*(U), \forall s_i: g(s_i) = c_{scj}^*\} \quad (31)$$

$$CS = \{CS(c_{scj}^*) | c_{scj}^* \in C^*\} \quad (32)$$

At the end of this part of modeling, all the definitions of a complete context can be captured as defined in Table 3.1. What follows is the association of priority with each service to facilitate dynamic laying out of each service within the service rendering interface.

3.3.2 Context-based Communities of Interest

For any composite service $s \in S'$ and any instance of context $c \in C^*$ we define some one-to-one function g'' on s into c , $g'': s \rightarrow c$. With respect to a particular context $c_k^* \in C^*$, a set of services is defined as:

$$CS(c_k^*) = \{(s_i, T_j^{***}) | s_i \in S', T_j^{***} \subseteq T_j^{++}(U), \forall s_i: g''(s_i) = c_k^*\} \quad (33)$$

T_i^{***} defines a subset of social ties from $T^*(U)$ of a particular service s_i . In other words, the pair (s_i, T_i^*) shown in Equation 11 is transformed into (s_i, T_i^{***}) after adding context. Similar to context-aware services, community of interest can also be modeled depending on the source of context. For the case of sensory media-based context, we can formulate the expression of community of interest as:

$$CS(c_{sk}^*) = \{(s_i, T_j^{***}) | s_i \in S', T_j^{***} \subseteq T_j^{++}(U), \forall s_i: g''(s_i) = c_{sk}^*\} \quad (34)$$

The above expression can be modified for context coming from services as follows:

$$CS(c_{sck}^*) = \{(s_i, T_j^{***}) | s_i \in S', T_j^{***} \subseteq T_j^{++}(U), \forall s_i: g''(s_i) = c_{sck}^*\} \quad (35)$$

3.3.3 Service Priority Modeling

Usually, context-aware services are rendered to a GUI such as web-based portal. A portal is placeholder of services where each service is rendered within each display unit called a portlet. Instead of rendering the same service in the same place all the time, the service priority modeling aims at dynamically allocating each service a

spatial location on the portal based on acquired priority or credit. Priorities are attached to services to differentiate between them on the display at run time. These priorities are reflected from the context of the user.

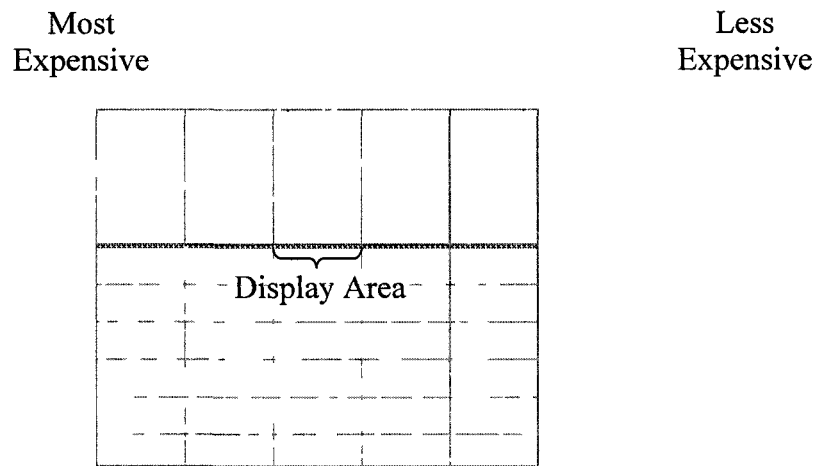


Figure 3.4 A display divided into blocks containing credit values.

The layout of each context-aware service is based on the real estate market theory [Rahman, Md. A. (2009a)] and responsible for laying out the services according to the credit earned with respect to a user context. This is done by two steps:

1) *Initialization step.*

This divides the user interface into equal sized blocks with each having a price according to its position on the display; for example, the blocks that fall within the middle region are more expensive than the side ones, as seen in Figure 3.4.

2) *Allocation step:*

The content provided by each service is displayed on the most expensive block that can be afforded by this service. Therefore, the important services are displayed on the important areas, and vice versa.

As mentioned in Equations 27-32, CS is the set of all the context-aware services realized by Table 3.1. The real estate market theory is based on credit values, which needs to be normalized to a set of quantized values of priorities with a certain threshold between two successive priority values. For simplicity, we assume the threshold between two successive priority values is one, which can be chosen based on application requirements. In order to assign a priority with each service based on user context, we first quantize the priorities $P = \{P_1, P_2, P_3, P_4 \dots, P_l\}$, where each priority instance falls within the range $P_i \in \{1, 2, \dots, p\}$. The value of p determines the depth and fineness of number of services a user wants to appear at any given context. The less the value of p , the less will be the difference between two closely matched services. Each context-aware service can be assigned a priority based on user preferences as well. Context-aware services with attached priority can be defined as following:

For any composite service $s \in S'$ and any priority $p \in P$, we define some function g on s into p , $g': s, g(s) \rightarrow p$ such that

$$\forall s \in S', g'(s, g(s)) = \{p | \forall p \in P\} \quad (36)$$

$$CS'(c_k^*) = \{(s_i, p, T_i^{***}) | s_i \in S', T_j^{***} \subseteq T_j^{++}(U), p = g'(s_i, g(s_i))\} \quad (37)$$

$CS'(c_k^*)$ defines the context-aware services and social ties, called community of interest where each service is also annotated with a priority value so that the service rendering interface such as portal can dynamically adjust the spatial location of each service window like a portlet.

3.4 SUMMARY

In this chapter, we have shown a generic modeling of context-aware social network framework. The modeling has been illustrated using four logical and consecutive steps. During the first step, we show the expressions to identify the variables and entities to extract information about social ties and services used by a subject from different heterogeneous sources of the Internet. At the end of this modeling step, the framework can extract one's PSN containing social ties and services. Next, user context has been modeled with the help of a tree-based context topology. The context topology allows three levels of context granularity. We have shown how to capture user context from BSN and diversified sources from the Internet. Finally, we have elaborated detailed expressions of forming complete context expressions from context primitives. Once context has been modeled, we map each complete context unit with a subset of services and social ties during the third step of modeling. Finally, each context-aware service is allocated a credit, which is converted to a priority and based on the acquired priority, each service is allocated a position dynamically on the service rendering interface.

CHAPTER 4: DESIGN OF CONTEXT-AWARE SOCIAL NETWORK FRAMEWORK

In this chapter we elaborate the design of the framework that consists of three main subsystems. The first subsystem is used to collect the raw information about a subject's social ties and different services he/she consumes from the Internet. The challenges during designing this subsystem is to choose the right algorithm that can communicate with complex and proprietary sources of existing heterogeneous Internet-based services and extract both social ties and the service's information of a subject. During the extraction process, the framework leverages the open stack of the Internet, illustrated in Section 2.1.

The second subsystem of the framework collects live user contexts from both body sensor network and heterogeneous sources of diversified services. In this section we first illustrate the design of the body sensor network in terms of hardware, software and communication architecture. Next, we elaborate how we capture user context from multimedia content of diversified social network services and realize the user context modeling shown in Section 3.2.

The third subsystem of the framework utilizes a novel *Ubiquity Stack* that maps a set of services and social ties with a particular user context. We present the algorithms to accomplish the context mapping then present the rendering of the context-aware services and corresponding community of interest in appropriate user interfaces. Figure 4.1 shows the high-level architecture of the proposed context-aware framework. As shown in the figure, the Internet and the sensors are external to the framework from where the framework gathers raw information to find the context information. We have taken advantage of different third party open source software to accomplish our goal. Those software components are represented by dashed boundary

(see Figure 4.1). However, the rest of the components are developed as part of this thesis initiative. Following this are the details of each of the three steps mentioned above.

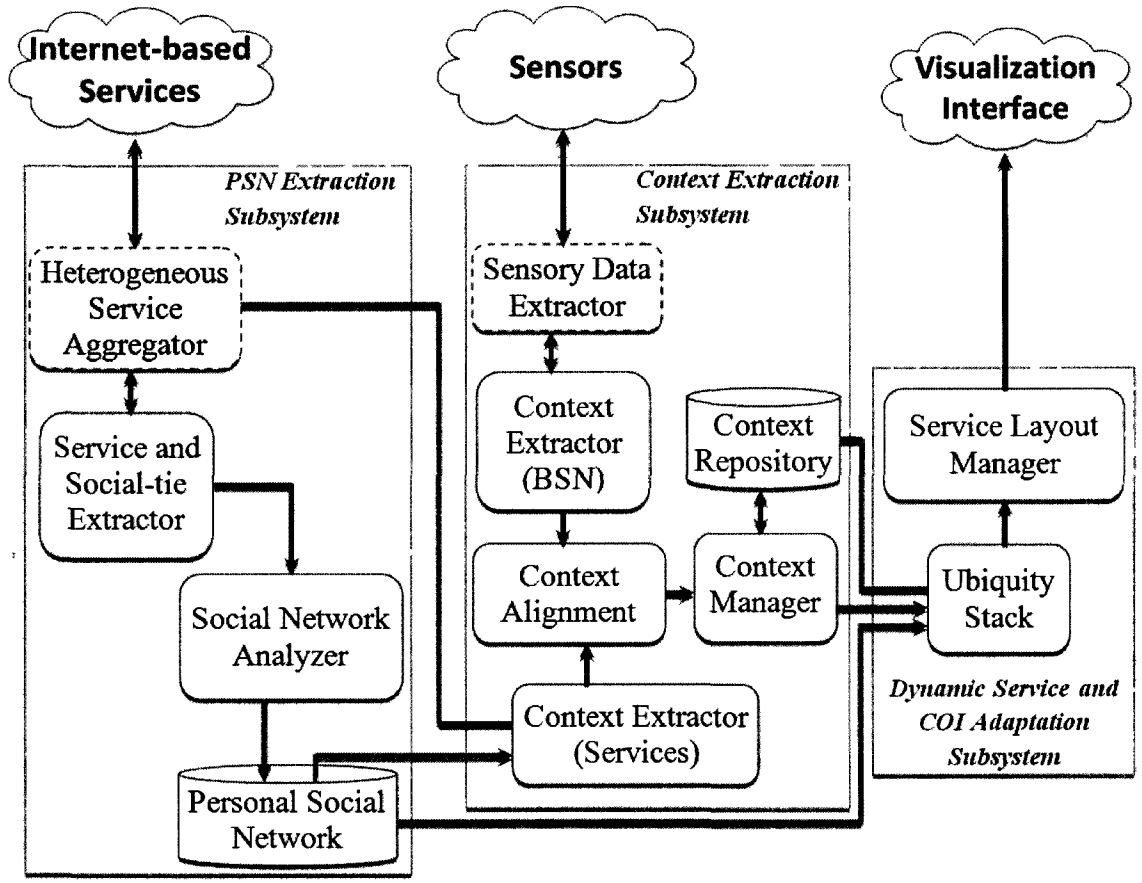


Figure 4.1 Block diagram of the proposed SenseFace context-aware personal social networking system.

PSN Extraction Subsystem

This subsystem consists of tools and components that facilitate the realization of the open stack of the Internet to crawl heterogeneous sources of services and extracting one’s personal social network. As shown in Figure 4.1, the Internet is

external to the subsystem, which is the repository of billions of people's personal social network. This subsystem consists of the following major components:

- *Heterogeneous Service Aggregator*
This component employs the algorithms that use open stack to provide information on how to access existing Internet-based services and extract social ties from each service. It employs *proxy server*, which acts as a rendezvous point for *Service and Social Tie Extractor* to provide each service definition. It is noteworthy that many components within *Heterogeneous Service Aggregator* are borrowed from third party open source software providers.
- *Service and Social Tie Extractor*
This component hosts two sub-components. The first is the *Service Extractor* which embeds the logic and access control mechanisms to gather information regarding a subject's involvement with heterogeneous sources of services and scrapes information about individual service. The second is *Social Tie Extractor* which filters social ties' information from the *Service Extractor* or from the web.
- *Social Network Analyzer*
The *Social Network Analyzer* uses existing social network analysis methodologies to create a personal social network. It takes the indexed uncategorized social network dataset and categorizes it according to a subject's social context vocabulary.
- *Personal Social Network (PSN) Database*
PSN Database stores the up-to-date mirror of a subject's personal social network that is used by other two higher layer subsystems to extract context and to render context-aware services and social ties to appropriate user interfaces.

Context Extraction Subsystem

The context extraction subsystem employs the components that are capable of handling raw context data coming from heterogeneous and proprietary sources. This includes sensors, email, text messages and instant messages, and maps data from each category of context into some standard format that can be processed by the context manager and stored in the context repository. Next, we briefly outline each of the major components of this subsystem.

- *Sensory Data Extractor*

This component runs inside a personal gateway such as a PDA or smartphone and communicates with the physical sensors to collect the sensory data. It pushes the sensory data to *BSN Context Extractor*. We have leveraged many third party open source sensory data analyzer software within this component to capture and analyze sensory data.

- *BSN Context Extractor*

BSN Context Extractor employs low-level sensory data receiving, storing, analyzing and communication functionalities.

- *Context Extractor from Heterogeneous Services*

Service Context Extractor reads the service definition of a user from PSN dataset and then scans through each of the services from the Internet to gather user context from live multimedia information contained within the services.

- *Context Alignment*

In order to integrate the heterogeneous sources of context data, this component bridges semantic disparities and results in a uniform context metadata.

- *Context Manager*

Context Manager receives context primitives available from the *Context Alignment* component, employs context aggregator logic to create complete context expressions and stores the context in *Context Repository*.

- *Context Repository*

The primary job of the *Context Repository* is to store the schema of context primitives as well as complete context model and register and notify context clients, such as dynamic service and COI adaptation subsystem.

Dynamic Service and Community of Interest Adaptation Subsystem

This subsystem takes the results produced by both PSN and Context Extraction Subsystems and provides a subset of context-aware services and social ties. In order to do this, this subsystem employs a novel *Ubiquity Stack*. The subset of services and social ties are then fed to a dynamic layout manager which adapts the layout of each service based on the assigned priorities. Following is a brief description of the major components within the subsystem.

- *Ubiquity Stack*

Ubiquity stack can intelligently follow user context change and recommends a new set of services with priority and community of interest. It also updates the priority of each service in the *Context Repository*.

- *Service Layout Manager*

This component receives a subset of services and COI lists from the *Ubiquity Stack* that need to be rendered to the current user visualization interface. It manages the layout of the services to the user interface by assigning a priority to each service so that each service is rendered to a location based on a dynamically assigned credit value.

4.1 EXTRACTING PERSONAL SOCIAL NETWORK AND SERVICES INFORMATION

People create many short-term ties through diversified services. Existing services are so diverse and proprietary in nature that dynamically extracting information about each consumed service information and social ties from a person's social network is a challenging task. First, due to a recent trend of involvements with different communities through different Web 2.0 technologies, it is hard for a user to keep track of every service and the social ties connected through that service statically [Smith, H., Rogers, Y., and Brady, M. (2003)]. Second, a person's social network tie information might be distributed in different proprietary services. It requires a complex authentication and extraction mechanism with 2-way, 3-way or n-way back and forth communication between a user and the concerned social network services [Sunyaev, A., Kaletsch, A., and Krcmar, H. (2010)]. For example, extracting electronic health record (EHR⁸) information from Google Health Data API⁹ requires complex access control as well as an authentication mechanism which is different than extracting social tie information from Microsoft HealthVault¹⁰. Moreover, once a social tie's profile is extracted, merging the profile and health record requires privacy and domain-specific knowledge. Third, the huge relational information between a user and his/her social ties is scattered in sources such as websites, social networks, emails, social bookmarks and blogs, to name a few. Nonetheless, collecting all the social ties of an individual from heterogeneous networks of a large dimensional dataset is a great challenge [Rahman, Md. A. et al. (2010a)]. A promising approach is to use open Internet stack, as discussed in Section 2.1.

⁸ <http://www.ncrr.nih.gov/publications/informatics/ehr.pdf>

⁹ http://code.google.com/apis/health/docs/2.0/developers_guide_protocol.html

¹⁰ <http://www.healthvault.com>

In this thesis, we utilize the open stack structure to access heterogeneous social networks over the web. However, due to opens stack's nascent nature, we detail the realization of the open stack through this chapter. We assume that the global set of existing Internet-based services is deployed within the *Service Clients* component. *Service Clients* uses open stack to access the actual service provider. For example, a Gmail service client has the domain specific knowledge of accessing Gmail server; a Gmail social tie client has the domain specific information of extracting social ties from the Gmail server and so on.

A properly designed *Service Clients* algorithm is envisioned to store the path to an Internet resource, bandwidth, round trip time, delay, URL patterns, HTTP access methods, response types and authentication patterns, among many others. Managing these algorithms individually for each service at different temporal points poses a major inconvenience, and hence a challenging task. Service discovery algorithm can be used to crawl over the Internet to constantly find new links of any particular resource and examine any external information (for example, resource metadata, usage metadata, resource profile) about it. For example, a URI may refer how to access the resource while its HTTP header provides information on how to read its content.

Services and social ties of a person can be represented as a tripartite graph with hyperedges [Mika, P. (2007)]. Each user communicates his/her social network through services, thereby creating a ternary association among the subject, the services and the social ties. The set of vertices is partitioned into three disjoint sets:

$$A = T(U) \tag{38}$$

$$C = S \tag{39}$$

$$A^* = T^*(U) \tag{40}$$

where A refers to the space of actors consisting of social ties derived from Equation 1, C is the concept space consisting of set of services derived from Equation 9, and A^* refers to the set of social ties of the subject U based on its connecting source (derived

from Equation 2) such as the service element from set C through which U communicates with the particular social tie. Thus the PSN space to explore can be defined as:

$$R \subseteq A \times C \times A^* \quad (41)$$

The above network represents a hypergraph with ternary edges, where each edge represents the fact that a given actor is connected with a certain instance of community through a certain instance of concept space. So the tripartite hypergraph can be represented as follows:

$$H(R) = \langle V, E \rangle \quad (42)$$

$$V = A \cup C \cup A^* \quad (43)$$

$$E = \{\{t_i, s_i, T_i^*\} | (t_i, s_i, T_i^*) \in R\} \quad (44)$$

Because the concept space can be a web service providing social network ties, or email provider or similar entities requiring special authentication mechanism, it is rather useful to break down the tripartite graph into bipartite graphs. The first bipartite graph shows the association between a user and a concept (AC graph), the second graph (CA* graph) shows the relation between each concept and related social ties, and the third graph (AA*) is actually the instance that we are interested in i.e. the association between each user and his/her social ties. To summarize, the AC graph portrays the association of a user with a set of services and the AA* graph shows the social ties of a user through each service. In other words, the purpose of *Service Clients* algorithms is to construct the AC and the AA* graphs, which use different layers of open stack to extract social tie and service information.

The *Service Client* algorithms are utilized by three components within the PSN extraction subsystem. The total set of SenseFace framework hosted services is accessed by the *Heterogeneous Service Aggregator* component. After the successful

aggregation, *Service* and *Social Tie Extractor* extracts actual service and social tie information of a particular subject from the Internet. The *indexer* stores each extracted service and social ties pair in an index server. This serves as a raw social network dataset of a particular subject. This raw social network dataset is then subjected to the social network analysis process which categorizes the social ties according to the subject's social affiliations. This categorized (service, social tie) pair is called the PSN.

Later on, the PSN dataset is used by the framework for higher layer actions such as context-aware retrieval of services and social ties. The salient components of the PSN extraction subsystem is shown in Figure 4.2 where the *Internet-based Services* serve as input to this subsystem.

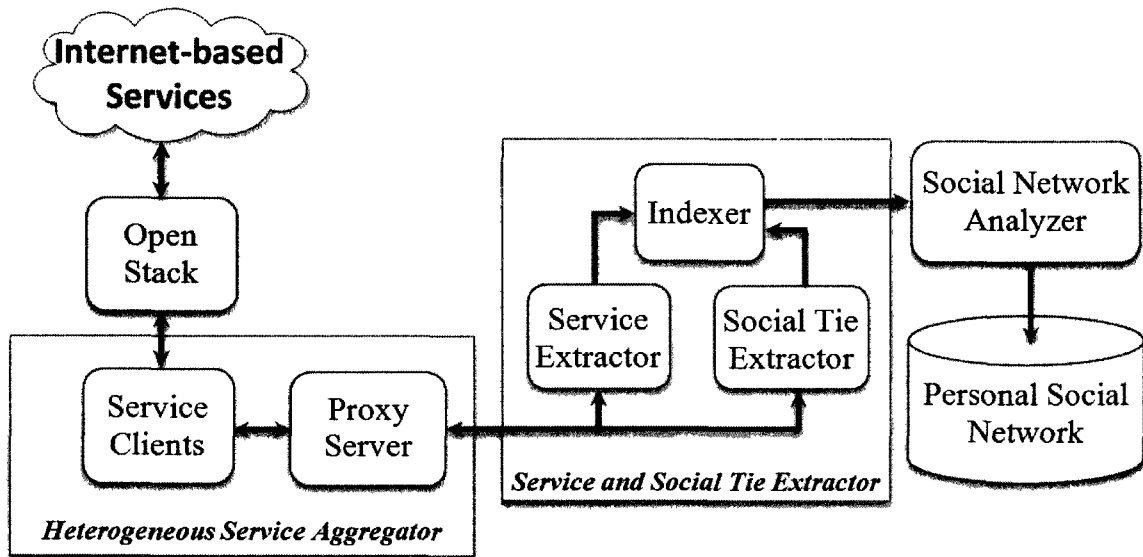


Figure 4.2 Block diagram showing major components of personal social network extraction subsystem.

4.1.1 Aggregation of Heterogeneous Services

The aggregation of one's association with diversified services and social ties is performed collaboratively by the *Open Stack*, *Service Clients* and *Proxy Server*

components. We assume that a client module of each unique service is hosted within the *Service Clients* component, as discussed in Equations 9 and 10. Theoretically, the *Service Clients* component is assumed to host clients of all the services that a typical subject consumes on a daily basis. The primary job of each service client is to expose a set of APIs that refers to different depths and breadths of features offered by a particular service. Moreover, a service client also keeps some additional proprietary metadata that are unique and a pre-condition to access a given service from the actual service provider. For example, types of authentication protocols supported, types of access control mechanism required, types and formats of responses exposed by the service provider, format of response payload, etc.

Each service client is responsible to load appropriate open stack modules with proper order such that it can negotiate with the actual service provider to successfully extract the content and social ties. Depending on the complexity of the service, a service client may need to utilize all or a subset of the open stack layers. Ideally, the open stack is assumed to be able to support any type of service available to date.

The *Service Clients* component is responsible to extract live contents and social ties from external services. This might make each service client overwhelmed with simultaneous accesses from the *Service and Social Tie Extractor*. To manage load balancing and scalability, the framework uses proxy servers. Each *Proxy Server* actually listens to *Service and Social Tie Extractor's* request and initializes appropriate numbers of personalized service clients.

4.1.2 Extracting Social Ties and Services Information

This component tries to find almost all the possible connections between a subject and services or web entities in terms of a URI in order to create the AC graph. For example, the concept space a user is connected to. Every such connection is realized by a successful establishment of all the services a subject is connected to,

types of required authentication, how many steps to finally consume the service, types of parameters needed, and the URI of the service. At the end of this step, a matrix is created which points to a subject's concept space. Before the next step, a subject needs to provide the authentication information required to connect to each service and pull the social ties' information from that domain.

Algorithm 1: Initiation of AC graph

```
Discover-Service (ServiceName, URI)
begin
    Collect initial user profile;
    Create XML skeleton at proxy server;
    Update XML file with initial parameters;
    If (URI isReachable) then
        Load Open Stack modules;
        Extract available APIs to access a particular service and social ties;
        Update the XML file in proxy server;
end
```

Prior to the initiation of PSN extraction, we assume that the subject provides at least an initial profile, such as an email address, blog or personal website that uniquely defines him/her or that it can check out a pre-defined list of options to search from. We use *Proxy Server* to delegate the call to appropriate service clients. Depending on the number of concurrent service requests, more number of proxy servers may be employed within the subsystem. A proxy server stores the list of services a subject is connected to in an XML format. For each concept space, such as finding social ties' information from Facebook, the PSN extraction algorithm maintains one unique XML file. The initial discovery algorithm for the AC graph is shown in algorithm 1. To explain how algorithm 1 works we assume that a subject has social ties through Facebook. The subject has declared that she intends to use OpenID for Facebook

authentication. The algorithm then tries to find the URI of the authentication service either at an OpenID server or any third party and initiate the authentication process. Upon successful authentication, the algorithm tries to realize the type of application or interface to access Facebook API. All discovery results are reflected in the XML file that has been created to access Facebook.

Sometimes domain names change, URL patterns alter, resources are relocated to a different server, subscriptions to some services change, HTTP access methods to some URI change, and so on. The same resource can thus be accessed using various ways from various mirror sites. Hence, we need to update the access information from time to time. Algorithm 2 is responsible to maintain the XML file with updated access information.

Algorithm 2: Update of AC graph

```

Update-Service (XML-file-URI)
begin
    Parse XML file from proxy server and create DOM;
    if (URI isReachable) then
        For each child-URI do
            Check accessibility of each API related to the service;
            Update the XML file;
    end
end

```

Both the *Service Extractor* and the *Social Tie Extractor* employ extraction algorithms to fetch respective personal social networks live from the web. Each extraction module is dedicated to one concept instance, for example, one to connect to Facebook, one to connect to an email server, one to connect to electronic health record and so on. The combination of responses brought back by all the individual service extraction modules for different services constitute the social ties of a subject.

Algorithm 3 shows the pseudo code of extracting the actual content from services. At the end of this step, the framework indexes the dataset containing raw information about social ties and services' information of a subject and stores them in an index server.

```

Algorithm 3 · Formation of CA* graph


---


Extract (XML-file-URI)
begin
  Parse XML file from proxy and create DOM,
  For each (Service-URI) do
    Load appropriate open stack APIs to access social ties,
    Extract social ties,
    Store service and social ties pair for indexing,
  end

```

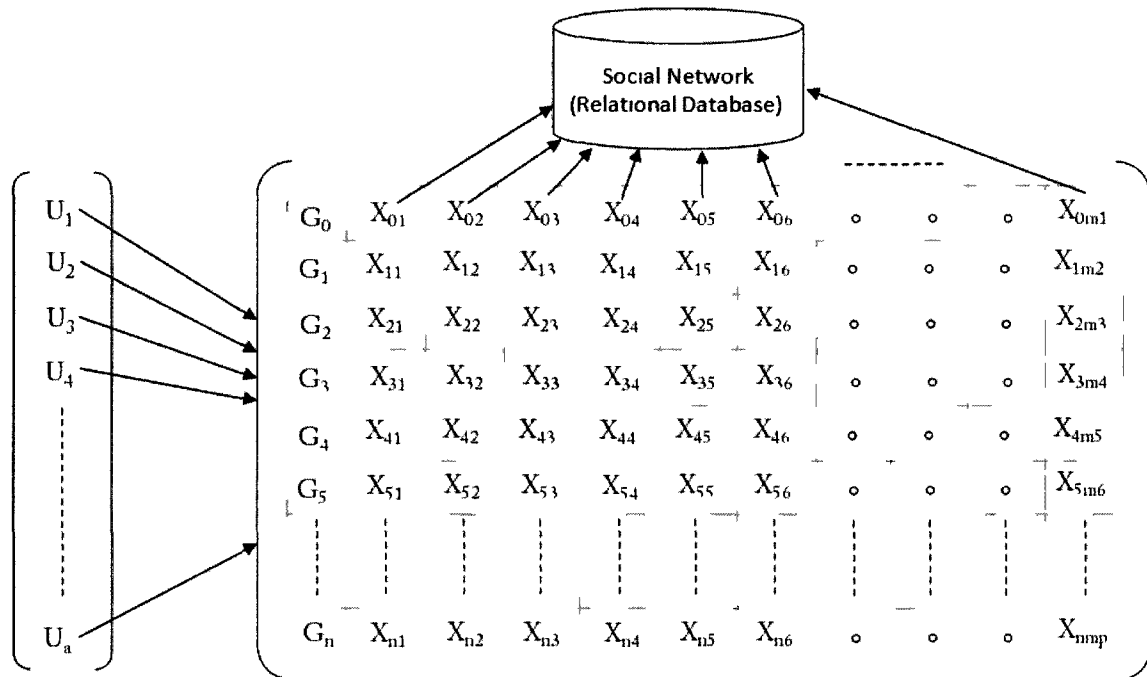


Figure 4.3 Dataset matrix stored within index server.

Finally, a mapping is done to merge the AC space with the CA* space and a global social tie space (AA* graph) is created. The AA* graph is realized by exposing standard interface for all the social ties of a subject. The resulting AA* graph can be represented by a matrix similar to the one shown in Figure 4.3. In Figure 4.3, U is the matrix of subjects' dataset whose social ties we want to categorize, which can be represented as $U = \{u_1, u_2, \dots, u_a \mid u \in A\}$ and $G = \{U_{i=0}^m G_i \mid G_i \in S\}$. For example, G_0 represents the vector of social ties related to one's professional network such as LinkedIn¹¹, G_1 is the social ties through family hierarchy such as GEDCOM¹² file, G_2 is the ties through email networks such as Google mail and so on. X_{ij} is an instance of a social tie with whom a subject u_i has social ties. The dataset collected so far is the amalgamation of the whole social network space of a user without a demarcation of which sub group he/she belongs to [Perer, A., and Shneiderman, B. (2006)]. For example, the social ties extracted from one's email server needs to be analyzed to get a semantic categorization of social ties. The next subsection handles the social network analysis process to categorize each social tie based on a user provided categories.

4.1.3 Social Network Analysis

Once the raw social network dataset is ready, the next step is to find which social tie belongs to which PSN category. Common PSN classification might be family, friends, colleagues, religious, sports, research and so on. For example, all the social ties through the Facebook social networks can be categorized and mapped into each PSN category. The idea is to use the matrix formed after employing the algorithms shown in the previous sections and analyze the matrix to find the PSN

¹¹ <http://www.linkedin.com/>

¹² <http://en.wikipedia.org/wiki/GEDCOM>

where each social tie from the matrix shown in Figure 4.3 falls within one or more categories of the PSN.

Given the actors (subject and social ties), social network analysis models tend to find a tie along with strength of ties between a subject and the related social ties [Snijders, T. A. B. (2009)], [Sarkar, P., and Moore, A. W. (2007)], [Velardi, P. et al. (2008)] and [Balasundaram, B. (2008)]. After analyzing an extensive survey, we conclude that the SIENA model [Snijders, T. A. B. (2009)] fits very well to the desired goal of this research. SIENA uses statistical analysis of social networks, which assumes that a set of actors and all the ties between these actors are known beforehand. The model takes into account different social network dynamics such as fixed and dynamic actor and tie covariates, social network tendencies such as reciprocity, transitivity, homophily and assortative matching. Because social tie changes might occur very frequently, SIENA allows collecting longitudinal network panel data in different temporal positions over a longer time period.

SIENA assumes that a tie between a subject and a member of his/her social networks maintains a state, rather than a simple event. Another assumption of SIENA model is that every actor knows all other actors of his/her social networks and controls all the outgoing ties. SIENA is based on the principle that the first observed network of social ties is given at the onset of the simulations and the model tracks the change between every two observed time points that are being modeled. We define different PSN categories, as shown in equation (4), for each subgroup of social network such as *friendship* $\rightarrow G'_0$, *kin* $\rightarrow G'_1$, *religious* $\rightarrow G'_2$, *family* $\rightarrow G'_3$, *research* $\rightarrow G'_4$, *colleague* $\rightarrow G'_5$, *health* $\rightarrow G'_6$ and so on, which can be summarized by the following equation.

$$G' = \bigcup_{i=0}^{sc} G'_i \quad (45)$$

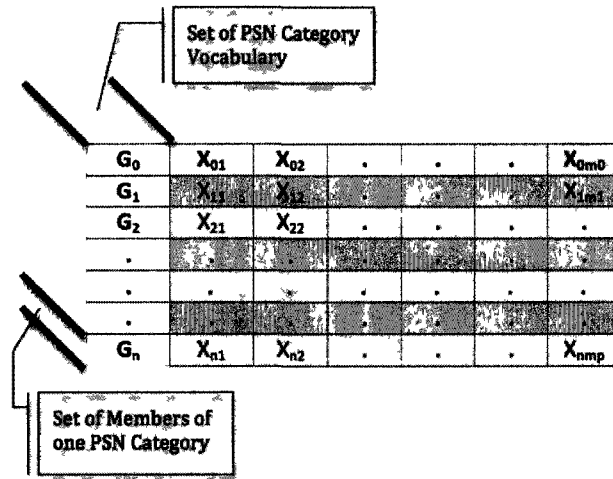


Figure 4.4 Forming personal social networks from the set of uncategorized social network dataset.

We prepare the dataset and encoding parameters according to the requirements of SIENA model. After the analysis using SIENA model, the resulting social network can be modeled similar to the one shown in Equations 38 to 44 except that the concept space is now composed of the PSN subcategories. Each social tie from the matrix shown in Figure 4.3 is now relocated to one or more of the concept spaces of PSN categories, similar to the one shown in Figure 4.4, where the first column in a particular row refers to a particular PSN categories, as described in Equation 45, and the rest of the columns of that row refer to the social ties associated through that social category.

4.2 COLLECTING USER CONTEXT FROM BSN AND INTERNET-BASED SERVICES

The context extraction subsystem mainly consists of six major components (see Figure 4.1): *Sensory Data Extractor*, *BSN Context Extractor*, *Context Extractor from Services*, *Context Alignment*, *Context Manager* and *Context Repository*. The first

component runs within the smartphone and extracts sensory data from physical sensors while the second component communicates with smartphone to pull sensory data from different sensors or push a query asking the current state of all or a subset of sensory data. This component then tries to infer context primitives from the available sensory data.

The third component scans through the list of services stored in the PSN database, downloads the multimedia information from the Internet and deduces context primitives from the extracted content. The fourth component handles the heterogeneity of context sources and helps in creating a unified output format for the subsequent context aggregation and complete context creation steps. The fifth component such as the *Context Manager* employs the necessary logic to combine all the gathered context primitives from the *BSN Context Extractor* and the *Social Network Context Extractor* and stores them in the *Context Repository*. The *Context Repository* holds the context values in the format shown in Table 3.1. Now we describe the functionalities of the subsystem.

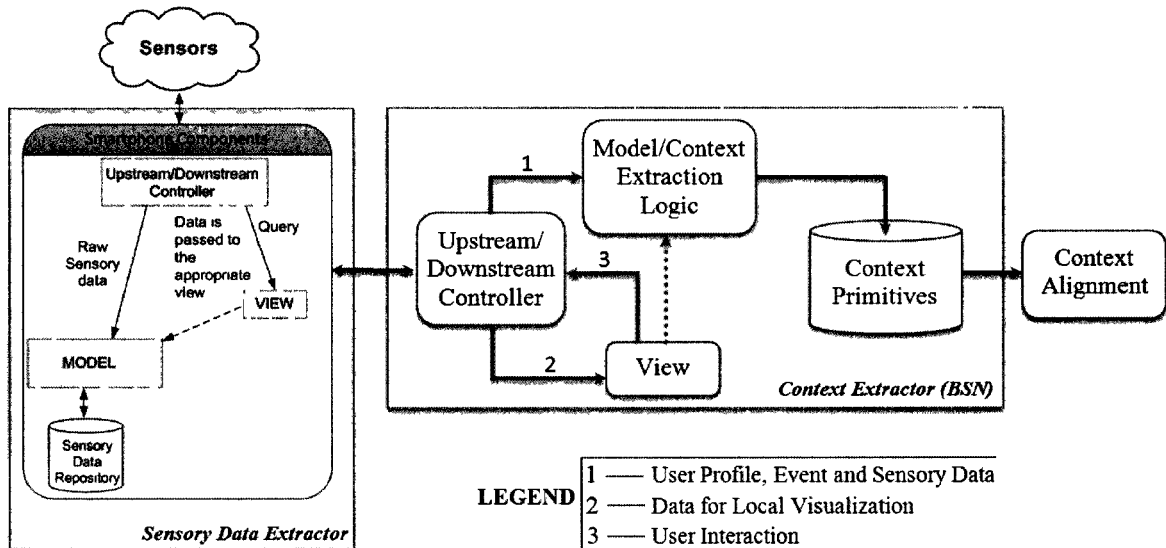


Figure 4.5 Block diagram showing components that extract context from the sensory data.

4.2.1 Context Extraction from Body Sensor Network

Context extraction from BSN involves two design aspects: the first, designing the body sensor network that follows user mobility and the second, context extraction components running within the context extraction subsystem. The first design aspect is concerned with the components that actually form the BSN and deployed as an application within the smartphone operating system. The second design paradigm is concerned with the reception of sensory data, analyzing it, extracting context primitives from the sensory data and supplying the primitives to the *Context Extractor*. The components of the context extraction system from sensory data are shown in Figure 4.5.

Sensory Data Extractor

It is responsible to collect raw sensory data from the BSN, convert it to intermediate level primitives and push it to the context retrieval component. The BSN consists of several wearable sensors, each of which detects a distinct physical or contextual phenomenon, and a local or personal gateway, which can be a smartphone or PDA. However, the use of a smartphone as local gateway gives us three relaxations. First, a user can carry both sensors and a smartphone both in home and/or outdoors. Hence, the BSN is never detached from the person throughout the whole day. Second, even if there is no Wi-Fi connection available, the framework can use the data services of the cellular network to communicate with the members of the social network. Finally, the rapid increase in persistent memory (for example, 32 GByte), high processor capability and data communication speed help in assuming that the smartphone can store raw sensory data, employ complex algorithm to fuse and trigger the data push mechanism.

In order to establish a two-way communication channel between a sensor and the smartphone, the BSN needs to choose a proper wireless standard that is available

in both sensor and smartphone technologies. Existing wireless standards vary in consumed power, transmission range, provided bandwidth, number of channels, number of devices supported per network, types of supported topologies, types of sensory devices and complexity in design. Among surveyed wireless standards, Bluetooth is found to be a suitable technology to support soft real-time applications at the wireless personal area network level [Nurmi, P. et al. (2010)]. It is suitable for a personal operating space of around 10-100 meters where relatively less number of sensory devices requiring moderate data rate can create an ad-hoc network. In particular, Bluetooth is designed for single-hop inter-device ad-hoc communication where nodes do not sleep much of the time and as a result network access is fast or the latency is less. Moreover, Bluetooth is now a de facto standard in smartphones. Many wearable sensors also come with integrated Bluetooth modules that make it a proper open source wireless standard to bridge between BSN and the cellular network.

Once a BSN is formed, the next challenge is on how a BSN communicates with the outer world. We assume that the BSN does not pose any burden on regular activities of a person. As mentioned earlier, using Bluetooth technology, a mobile phone can be paired with the master node of BSN to receive the sensory data. Both the memory and the processing power of a mobile phone are limited and this poses a challenge on the application of which data to store and for how long. Hence, we adopt the strategy of storing the data for about ten minutes (this is a variable and can be changed based on application requirement and technological advancement) in the local smartphone memory, and if no event occurs, we flush the sensory data. Another design aspect is to decide when to send the data over the wireless carrier; sending data over a cellular network is still expensive and thus needs to be used sparingly.

The *Controller* co-ordinates the interaction between low-level software components that actually capture the raw sensory data and the corresponding high-level software components. The high-level software clients store the raw sensory data

in the Smartphone's internal/external memory. The raw sensory data is then classified based on some pre-defined algorithms [Rahman, Md. A. et al. (2009b)] and converted from raw sensory data to some primitive values. The *Controller* is also responsible to listen to remote queries. For example, in response to a heart specialist's request, the *Controller* can initiate streaming raw heartbeat data for a requested period of time. For the purpose of brevity, we assume that the controller employs a privacy setting model as described by [Kapadia, A. et al. (2007)] to provide remote access of sensory data. For the scenario of upstream sensory data pushing from the BSN to the context extraction system, the *Controller* receives the sensory data from the different sensors and dispatches them to the *Model* components for storing and analyzing the data. If the captured sensory data remains within a defined limit, the *model* component simply flushes the stored data from the local smartphone memory after a pre-defined amount of time (in our example, ten minutes). In case the data needs to be sent to a remote server, the controller invokes appropriate *View*, populates the stored sensory data primitives together with occurred events and ambient information and sends them to the *Context Extraction Subsystem*.

As shown in Figure 4.5, in the case of a downstream query from members of social ties or from the context extraction system, the *Controller* receives a query, dispatches it to the proper *Model* to receive the current state of the sensors and sends the query result back to the requester. For better accuracy, the *Model* component possesses a data integration module. For example, the location data can be obtained from the internal sensors (GPS, GSM, and WLAN) and the external sensor (beacon). However, location obtained from a GPS module is more accurate than that obtained from a GSM unit and the beacon data describes the precise position within a small periphery. Thus the sensory data providing similar functionality may provide different levels of location granularity [Zhan, Y. et al. (2009)].

BSN Context Extractor

The BSN context extractor plays two important roles in the framework. First, it encapsulates all the logic to communicate with the *Sensory Data Extractor* component. This includes receiving and storing sensory data primitives and user profile from the *Sensory Data Extractor* and dispatching queries from authorized members of one's social network to the appropriate BSN. Second, it facilitates analyzing the sensory data primitives and associating context value.

As shown in Figure 4.5, the *Controller* receives the incoming sensory data and dispatches it to the context extraction module, which acts as model component. The *Controller* maintains both upstream and downstream communication. The *Controller* exposes two interfaces: through the downstream interface where each registered user's profile and sensory data can be accessed by authorized members of the social networks, and by using the upstream interface where a smartphone can upload sensory data. The *Model* component separates the incoming sensory data from user profile information and passes the sensory data to the appropriate context extraction logic. We further semantically classify the sensory data into three categories: physical sensory data that represents human body phenomena, ambient information and different event/activity information. The *Model* component employs different business logic to analyze diversified sensory data. In this way, the *Context Extractor* allows the incorporation of any privacy laws, data access control and financial policies within the model.

One of the important tasks of the *Model* component is to augment the received sensory data primitives with context value. We assume that appropriate business logic, defined by domain specific subject matter experts and containing threshold values for given sensory data, are customized for each user. Depending on the alert level, the *Controller* chooses appropriate view for an appropriate end-user by populating the view with relevant data coming from the *Model* components. The view can be

customized based on end-user devices and preferences. The view also responds to the interactions by the authorized members of COI to give him/her access to different sensory data. The sensory data with added context information is stored within the *Sensory Data with Context Augmentation* relational database as (name:value) pair. Table 4.1 shows some sample data that is stored in the database. As soon as one or more new value pairs are inserted within the database, it pushes the data to the *Content Alignment* component to take high level decision or to check whether user context has been changed.

Table 4.1: Deducing context primitives from sensory data.

(User : Alice)→registered user
(Temperature: 31°C)→hot region
(Light Intensity : 10,100 lx) →less light condition
(Latitude : 45° 19' N, Longitude : 75° 40' W)→Ottawa, Canada
(blood pressure : 160)→Unusual blood pressure
(pulse : 145) → higher than regular
(GPS: null for long period) and (two end-point GPS data location near subway station) → vehicle used:subway.

4.2.2 Gathering Context from Internet-based Services

Collecting user context from the multimedia information contained within heterogeneous services poses two challenges. First, it requires an up-to-date definition of the subscribed or associated services and then scans through each of the services to extract the multimedia information. This enables the context analyzing unit to infer user context from the analyzed data. Second, employing appropriate logic to extract context information from the available social network data to aggregate services augmented with context information and to store in a repository to be used for further

higher layer context analysis processes. Figure 4.6 shows different salient components of the context gathering system. For better representation of the concept, we describe the context gathering process with two logical units. The first is the extraction of multimedia data from different services over the web, and the second is deducing context primitives from the extracted content. Next, we elaborate each of the logical units.

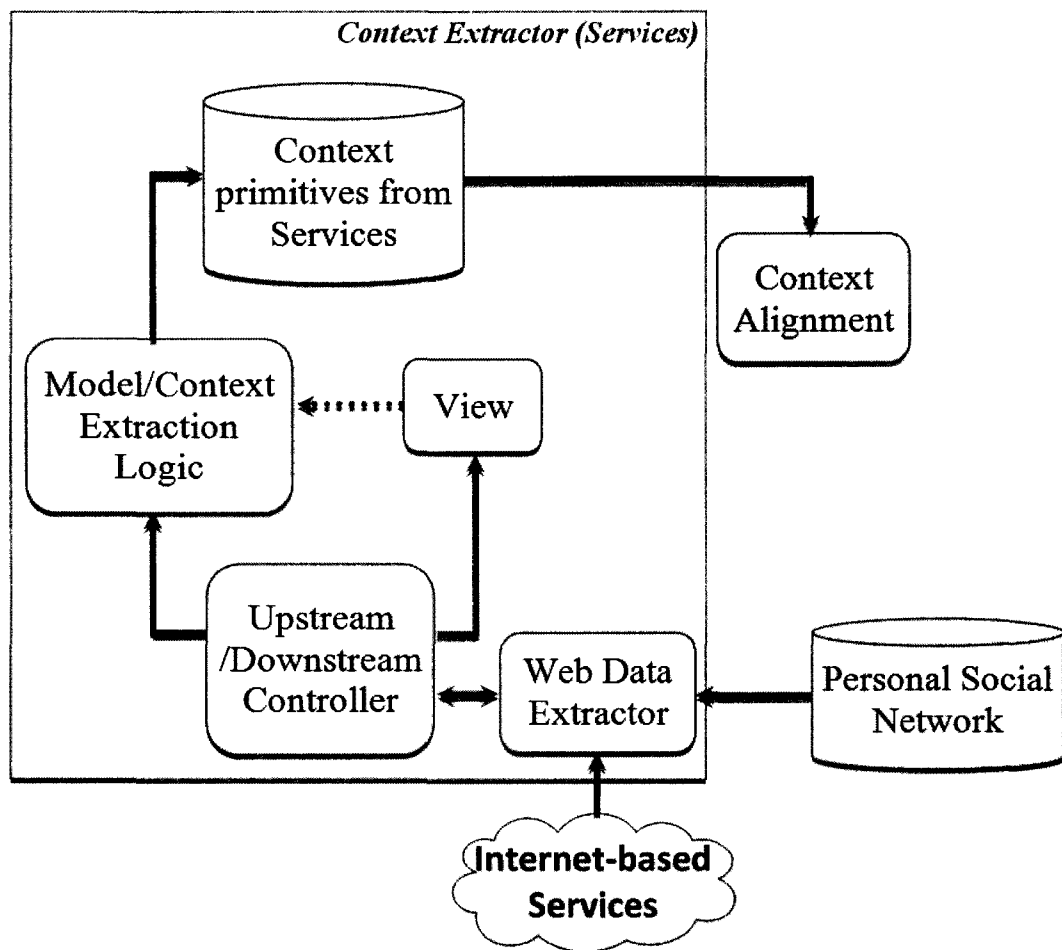


Figure 4.6 Block diagram of the context retrieval system from different heterogeneous Internet-based services.

Multimedia Web Data Extraction

The *web data extractor* takes part in the actual content extraction process from heterogeneous Internet-based services. It first collects the list of services from the personal social network database of the PSN extraction subsystem. It then passes the extraction request to the *service extractor* module. In ideal case, the web data extractor employs same number of content processing APIs as the *Service Clients* module of the *PSN Extraction Subsystem*. Each content-processing API encompasses a wide variety of methods including social network data formatting, range checking and data validation, algorithmic computation and normalization, augmentation, and interpretation/abstraction of low-level social network data into higher-level information that complies with the subsequent phases of context processing subsystems. For example, an event checker scans through the calendars of a user and if one or more new event(s) are within a pre-defined threshold of time, these services are subjected to a context-checking process by the context checking components that will be discussed later. In order to raise this alert, the service extractor simply raises a flag that a particular service has new content [Sun, S. (2009)]. All the services with new message content are tagged and subject to pass through context treatment process. Another dimension of *Web Data Extractor* service is that it can be configured to capture some metadata such as frequency of state change, accuracy of content retrieval, and granularity level. In this way, a user expresses her intention of how often a particular service might change its state with new content or how important the service is.

Context Extraction from Social Network Data

The *Controller* receives the content of each service as a payload and pushes the content and the metadata about the service provider to the appropriate *Model* components. It also notifies a user with the updated services to the *view*. The *Model*

component embeds logic to extract context information from the content payload. It is already mentioned that only those services are being subjected to the context extraction process that carries a new content or event. For example, a new Facebook status message update by one's friend simply passes the Facebook service to the context extraction process where the actual content analyzing logic is present. The co-location context can be used as an indication about a subject's surroundings [Kalasapur, S. et al. (2009)]. In addition, we can collect intra-user context such as location provided by third party social network services. The *Model* component employs the algorithms and domain specific knowledge of extracting context from each service. Hence, the richer the set of services in terms of depth and breadth, the richer the framework will be. As shown in Figure 3.3, context extraction system uses the context topology to identify the category of each incoming content payload and calls appropriate function, logic or model component that can extract the context from the actual content.

Algorithm 4: Extract Context from Services

```

Find_Context (UserID, ServiceID, ContentPayload)
begin
    Identify UserID, ServiceID;
    Identify Service Handler corresponding to UserID, ServiceID;
    Call Service Handler and pass ContentPayload;
    Receive and categorize context data;
    if (success==true) then
        | Store raw context data;
    else
        | ContentPayload is damaged or error in extraction process;
end

```

The algorithm of the context extraction process is shown in Algorithm 4. As shown in Algorithm 4, the *Controller* passes the service, which is flagged with a context extraction process, to the *Find_Context* method with the user and service metadata as well as actual content in terms of payload. The extractor then identifies the actual model component that is capable of analyzing the content (such as a Twitter message containing one's current flight location). This is then passed to the flight-checking module while a Twitter message containing a calendar event message is passed to the calendar parser module and current weather data is passed to the weather inference module, etc. Each service context handler finally classifies each processed context primitive into one of the categories shown in Figure 3.3 as described in Equation 17 and Equation 26. Samples of some stored context values are shown in Table 4.2.

Table 4.2: Example of raw web data received from services and corresponding deduced raw context data

<i>(User:Alice)</i> →Subject
<i>(Email-Sender:Tom)</i> →Friend of Alice
<i>(Email-Attachment:CalendarEvent)</i> →Meeting between Tom and Alice
<i>(Facebook-StatusMessage-Sender:Tom)</i> →Now in doctor's office till 2:30 p.m.
<i>(Twitter-Message-Sender:Pharmacist)</i> →Prescription refill ready to be picked up at 5:30 p.m. from Pharmacy
<i>(Weather-Message:WeatherService)</i> → Heavy rainfall this morning
<i>(Calendar:CalendarService)</i> →Daughter's doctor appointment at 9:30 a.m.

4.2.3 Alignment of Heterogeneous Context Data

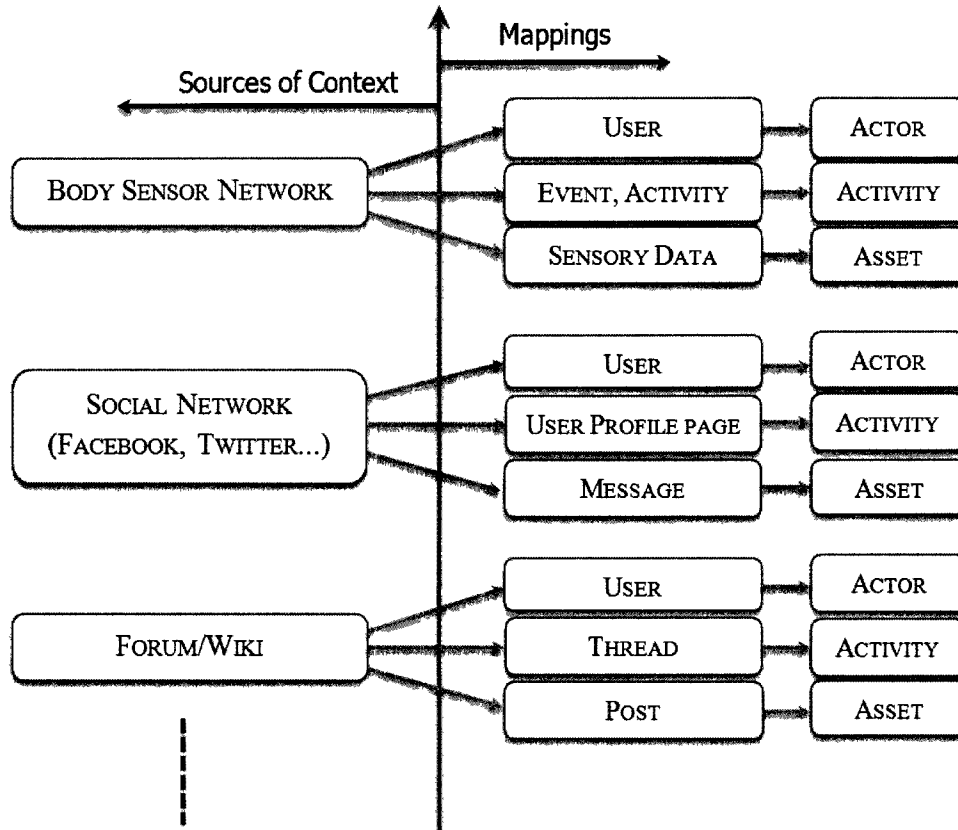


Figure 4.7 Preparing raw sensory and social network dataset for higher layer context mapping using 3A model.

Raw context data gathered from heterogeneous sources including Internet-based sources such as calendar, email, SMS, social portals, etc. and ad-hoc sources such as BSN tend to preserve diversified proprietary formats which should be transformed into a unified form. To this end, we adopt the Actor, Activity and Asset (3A) interaction model [El Helou, S. et al. (2009)] to serve the goal of context data alignment. We employ the 3A model to map every service and sensor in a BSN to one of the entities of the 3A model as shown in Figure 4.7. 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 such as sensory data, a Twitter message, etc. Thus, context information can 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. The mapping produces a unified context format of defined context primitives that can be easily converted to a context primitive by the *context primitive generator*.

4.2.4 Context Management

The *context management system* assumes that the raw context data is collected from the physical environment and finally aggregated and classified. Decision making processing takes place by the *context manager*. Figure 4.8 shows the major components that lay the backbone of the *context manager*. We now illustrate details of the components.

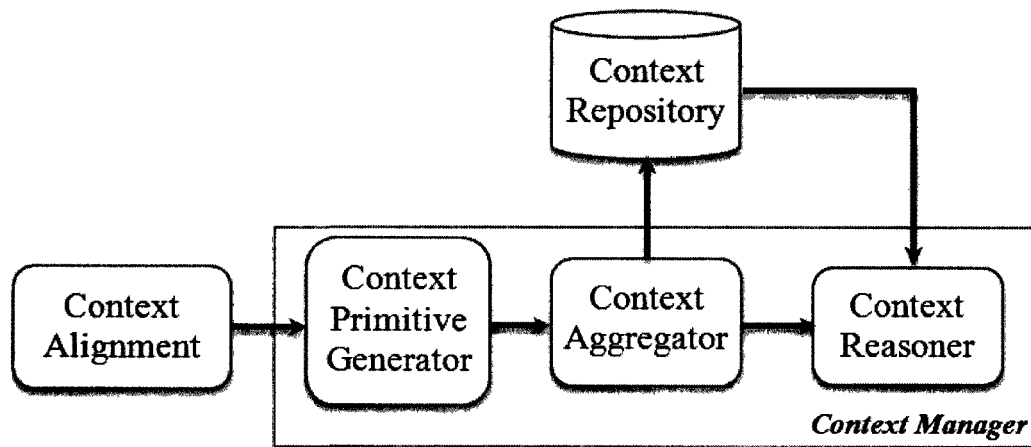


Figure 4.8 Block diagram showing major components within the context manager, external input sources of context or clients of context manager.

Context primitive generator

This follows a two-tier augmentation process with regards to its source. The first tier of augmentation is performed with regards to its source as defined in Equation 17 such as whether the primitive falls within spatial, task-related, temporal or personal category. The second tier of augmentation is done based on whether the primitive is coming from sensors or social networks, as defined in Equation 21. *Context primitive generator* takes the input from the 3A model of the *context alignment* component and generates the context primitives in the form shown in Section 3.2. In order to generate context primitives, it uses the context topology shown in Figure 3.3. As defined in Equations 22 and 23, context primitives can be categorized according to their source such as whether they are from sensors or from services. Accordingly, primitives' generator follows the model shown in Equations 25 and 26 to augment each context primitive's source. A scenario of context primitives has been shown in the following table:

Table 4.3: Deducing context primitives from sensory data and multimedia information of diversified Internet-based services.

<i>Values Received from Sensors/Services</i>	<i>Complete Context</i>
<i>(agenda received from online meeting client) and (people nearby are co-authors) and (location is university) and (cell phone status off)</i>	The subject is in meeting with researchers
<i>(blood pressure ≥ 160) and (pulse ≥ 140) and (sweat flag is on) and (Latitude - $45^{\circ} 19' N$, Longitude - $75^{\circ} 40' W$) and (GPS signal is absent)</i>	The subject is doing exercise.

Context Aggregator

Context aggregator uses logic to dynamically aggregate primitive contexts and create complete context, whether it is complex or compound. This module can be

designed to run according to some efficient aggregation techniques found in the literature [Ocampo, R. et al. (2008)], [Ryu, H. et al. (2007)].

Context Reasoner

Context Reasoner is responsible to derive higher level context information (a subject engaged in meeting at the University) from atomic level context primitives. Depending on the context information provided by *Context Aggregator*, a number of services might be involved. For example, assume a subject has access to three different audio/video conferencing tools. However, he/she has to use only one of them depending on the context, like depending on the type of community involved, types of end device available, types of mobility pattern or bandwidth available. In order to reduce context noise [Mikalsen, M. et al. (2006)], a subject might want to use some quality of context metrics defined in *Context-Quality* field shown in Table 3.1, to increase the quality of the framework. However, the subject is free to inspect and modify the outcome at any temporal point after the *Context Reasoner* has declared a complete context. The output of this module is a complete definition of user context that uniquely defines a certain state of the user. In fact, the output of this unit is fed to the *Ubiquity Stack* of the dynamic service and COI adaptation subsystem to map each complete context with a subset of services and social ties.

Context Repository

At the repository, context metadata along with values are stored in a referential database realizing the schema shown in Table 3.1, allowing context clients such as *Context Manager* and *Ubiquity Stack* (see Figure 4.1) to facilitate flexible and efficient lookup and browsing facilities [Gifford, D. K. et al. (1991)], [Gopal, B. and Manber, U. (1999)].

4.3 CONTEXT-AWARE SERVICES AND COI

This subsystem includes a *Ubiquity Stack*, a dynamic service layout management system. The *Ubiquity Stack* communicates with the context collection subsystem to gather user context dynamically and the PSN extraction subsystem to get hold of a mirror PSN database. The dynamic service layout management system manages the logic such as how to manage the physical layout of each context-aware service and community of interest on the user interface. Different components of the subsystem are shown in Figure 4.9.

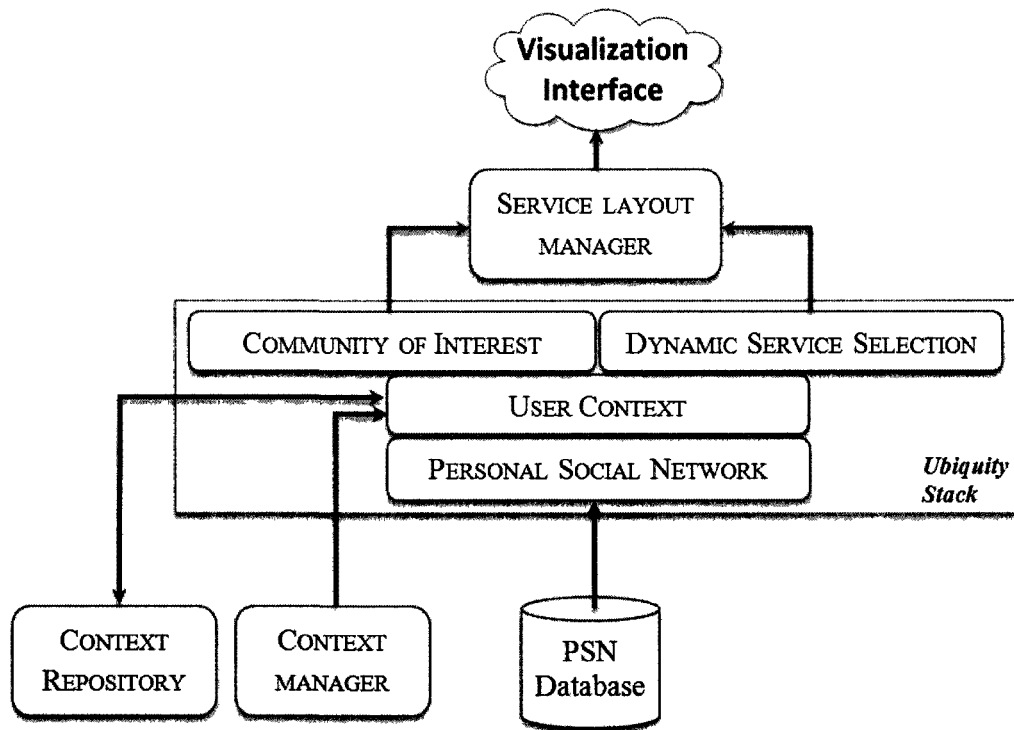


Figure 4.9 Dynamic service and community of interest adaptation subsystem.
4.3.1 Ubiquity Stack

The role of this stack is to provide a ubiquitous overlay network by providing the logics, protocols and algorithms to map user context with a subset of services and

COI. This layer leverages a mirror dataset of one's PSN as the PSN database holds the personalized list of services, access control details and social ties information. The stack facilitates the storing of one's PSN in a smartphone or on 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 relatively quickly. Also, this layer facilitates a user to specify higher layer concepts such as ontology, 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.

Layer 1: Personal Social Networks

This layer plays an important role in the framework by maintaining an upstream communication with the PSN database. As discussed earlier, the PSN database 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 this layer is capable of receiving any update in terms of services or social ties of a subject. This layer also plays a salient role by providing categorized social ties of a subject, as defined in Equation 45, to the upper layers. There are several benefits to such a 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, phonebook contact list, 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 [Rahman, Md. A. et al. (2010b)].

Layer 2: User Context

This layer always keeps an up-to-date user context definition supplied by the context processing subsystem discussed earlier. As soon as the user context changes, as reported by the context subsystem, this layer dynamically reflects the new user context definition, stored by the context repository. Context manager notifies that user context has been changed by supplying the new contextID. This layer then inquires the context details from the context repository using the supplied contextID. Once details about the context are available to this layer, it shares this information with the two upper layers to adapt to this context change by linking to a new subset of social ties and services.

Layer 3A: Community of Interest

This layer semantically maps each social tie from the PSN category with a subject depending on the user's context. In order to map the user context with PSN, we incorporate semantic knowledge. In the case of physiological contexts such as *alarming heart rate*, a mapping is done among sensor profile (ECG sensor), resource profile (smartphone ECG handler application and ECG handler web services), user profiles (family physician, immediate caregiver, family member, friend, colleague, and emergency contact), and context profile (physiological, event-based, spatial and temporal).

This layer employs higher layer logic to map the user context provided by the bottom layer with a subset of social ties. In its simplest form, a user can perform this job manually. This mapping can also be done semantically by employing algorithms, presented by [Rahman, Md. A. et al. (2010a)]. Assume that an elderly person is having breathing problem, which has been identified by the breathing sensor. Upon receiving the sensory data, the smartphone marks the threshold as *alarming* and thus pushes the sensory data and the user context information such as location, the breathing and

activities information during the last ten minutes to the context extraction subsystem. This layer then identifies the community of interest that is linked with this breathing instability event (family physician, immediate caregiver family member and geographically close family member or friend). Another scenario is a trigger coming from temperature sensor as a result of a fire in a house. The subset of social networks that are affected due to the event like the neighbours, emergency services, family members and friends would need to be contacted.

In order to map the two domains, namely user context and social ties from the PSN, we need to incorporate semantic knowledge. Semantic similarity can be defined as a concept by which a set of terms are assigned a metric based on the likeness of their meaning. In the process of assigning such a metric, ontology plays an important role [Rahman, Md. A. et al. (2010a)]. By defining similarity measures, we can establish semantic relations between two concept spaces. In the context of this thesis, ontology is suitable for representing such mapping. In general, ontology can be formalized as follows:

$$\mathbf{O} = \{CN^*, \{R_{cn_1^*, cn_2^*}\}\} \quad (46)$$

where O is the ontology, CN^* is the set of concepts described by the ontology, $\{cn_1^*, cn_2^*\} \in CN^*$ are two concepts and the semantic relation between the concepts can be represented by $R_{cn_1^*, cn_2^*}: CN^* \times CN^* \rightarrow \{0,1\}$.

Although ontology provides a rich source of semantic knowledge representation, modeling social relations depends on semantics, context, desire and intentions of a subject which can be vague and may differ from user to user. Generally, people use linguistic categories to map in real life. Fuzzy logic is specialized in modeling these vague modes of social reasoning [Smithson, M. J., and Verkuilen, J. (2006)], [Sanchez, E. (2006)] and takes into account the stochastic

component of the human reasoning. The crisp relation in social life can thus be represented by the fuzzification of the ontology definition provided in Equation 11.

$$O_f = \{CN^*, \{R_{f(cn_1^*, cn_2^*)}\}\} \quad (47)$$

where O_f is the ontology after fuzzification, and $R_{f(cn_1^*, cn_2^*)}$ is the fuzzy relation between the concepts c_1^* and c_2^* which is defined as:

$$R_{f(cn_1^*, cn_2^*)} = F(R_{cn_1^*, cn_2^*}) : CN^* \times CN^* \rightarrow \{0,1\} \quad (48)$$

Equation 48 is a generalized expression of fuzzy ontology for any domain. The fuzzy ontology assumes that each object is related to every other object in the ontology, with a degree of membership assigned to that relationship based on fuzzy logic [Sanchez, E. (2006)]. We model the above fuzzy ontology using the following four steps [Rahman, Md. A. et al. (2010a)]:

- Fuzzification: mapping each sensory data type into a fuzzy set, given the crisp inputs from context repository and social ties from PSN.
- Determining individual rule, to semantically map the input and the output in the fuzzy domain.
- Determining the aggregate output of all the fuzzy rules.
- De-fuzzification: mapping each fuzzy output to a crisp set of outputs such as a vector containing COI.

An important factor which defines the breadth of the members of COI is the threshold value T_c of the above fuzzy ontology. Details about fuzzy ontology threshold selection processes can be found in [Quan, T. T. et al. (2006)], [Bahri, A. et al. (2009)], [Lee, C.-S. et al. (2009)], [Kong, H. et al. (2009)] and [Lam, T.H.W., and J.N.K. Liu (2008)]. Value of T_c determines whether or not a particular social tie falls within a concept space as defined by PSN category. Since the semantic relationship

between two concepts is varied in the range [0, 1]; the value of T_c also lies in that range: varying from 0.0 (do not match at all) to 0.9 (fully matches) while constructing the fuzzy context. The region of T_c that determines whether a member of PSN concept domain is related to a particular context is chosen as 0.3. That is, when T_c is set to 0.3, the PSN keywords that have membership values less than 0.3 will be filtered out of the constructed COI space. Currently there are two approaches to define the value of T_c , either manually or automatic. For automatically finding the value of T_c from one's PSN databases, we follow the approach illustrated in [Quan, T. T. et al. (2006)]. Although we assume that it requires a subject matter expert to create a fuzzy ontology, it is not realistic to assume that it is perfect and that any fuzzy value is well-defined and suited to any type of application. Thus, we adopt a mechanism to change the fuzzy values and the threshold in order to fit them in the best way to a specific application domain.

As shown in equation (37), if c_k^* is user context and T_j^{***} is the subset of social ties associated with this context, then we can rewrite equation (48) by substituting the concepts c_1^* and c_2^* with c_k^* and T_j^{***} respectively as following:

$$R_f(c_k^*\{s_\nu T_j^{***}\}) = F\left(R_{c_k^*\{s_\nu T_j^{***}\}}\right) \quad (49)$$

As the context sources might be sensory media or web media, we further expand the above equation according to equations (34) and (35) as following:

$$R_f(c_{sk}^*\{s_\nu T_j^{***}\}) = F\left(R_{c_{sk}^*\{s_\nu T_j^{***}\}}\right) \quad (50)$$

$$R_f(c_{sck}^*\{s_\nu T_j^{***}\}) = F\left(R_{c_{sck}^*\{s_\nu T_j^{***}\}}\right) \quad (51)$$

After the de-fuzzification process (by utilizing equations (20) and (45), the output looks like the matrix shown in Figure 4.10 below. Each user U_i has $\{c_1, \dots, c_n\}$ number of complete contexts stored by the SenseFace framework where each context c_i is composed of a subset of social ties and services from $\{G_0^*, \dots, G_m^*\}$

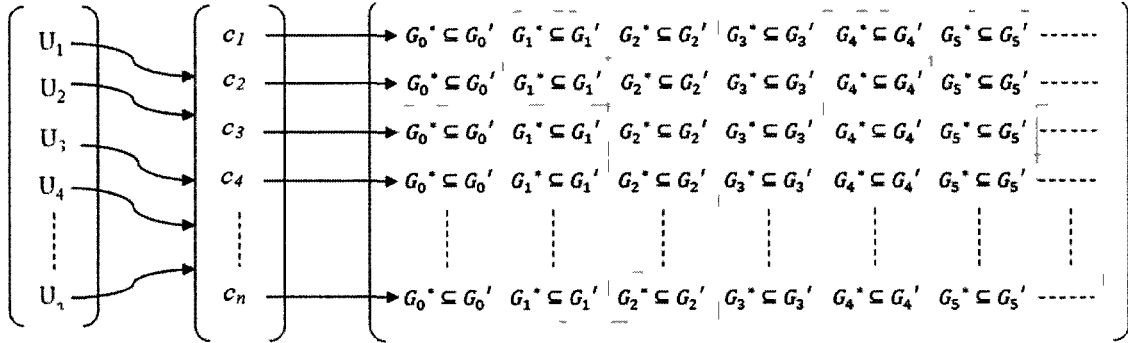


Figure 4.10 COI matrix that represents each ego and his community of interest depending on the type of sensory data.

As a result of the de-fuzzification, the context aware overlay consisting of the categorized social ties space is re-organized, which is identical to Equation 45. This maps each distinct context type with a COI so that as soon as a user context is changed in the physical space, the context-aware system triggers a context change within the framework and semantically tries to answer who to contact at this situation.

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. Because social ties are assumed to be extracted from services and interactions between a subject and her social ties take place through the service plane, the mapping of services and user context can be achieved through Equations 46 to 51. Once a subset of services is mapped with a particular context instance, it assigns credits to individual services within the range $[0, 30]$ in accordance with the real estate algorithm described in Section 3.3.3, where 0 is the lower and 30 is the upper credit bound. It is noteworthy that only services that pass the context manager are tagged with a credit value so that the *service layout manager* can use these credits to convert to corresponding priority values to layout individual services to a particular location within the user interface.

4.3.2 Service Layout Manager

This component performs four important tasks. First, it maps the credit value of each context-aware service and normalizes it to a priority value. Second, it dynamically creates a spatial grid-based layout of the user's visualization interface and allocates each service to a certain grid-unit, based on the priority value. As soon as user context is changed, a new service layout is pushed to the layout manager, which refreshes the user interface dynamically. Third, this component is responsible to extract the content of each service, provided the user is connected with the Internet. Finally, this component receives an up-to-date list of COI from the *community of interest* sub-layer to be rendered to the user interface. In addition to simply showing the list of social ties, this component can embed business logic regarding what kind of context information to be annotated with the user list. For example, location, current activity, availability, visibility, accessibility, social category, connecting service type and emotions can also be attached to the user listing.

4.4 SUMMARY

This chapter outlines the design of the proposed framework in the light of three subsystems. The first subsystem is called the personal social network extraction subsystem, which pulls one's association with existing services and social ties. The second subsystem performs the context extraction process from two sources: the body sensor network and Internet-based services. Finally, the dynamic service and community of interest adaptation subsystem maps the services and social ties with the extracted user context. The services are attached with a priority to be rendered to the user visualization interface such that the most critical services are rendered at the center and the rest of the services are laid out progressively toward the outskirts.

CHAPTER 5: FRAMEWORK IMPLEMENTATION

The SenseFace framework has three subsystems that work together where the dependencies among different components of each subsystem have been illustrated in Section 4. To maintain a coherent presentation style, we illustrate details about the implementation of the framework by dividing it into three parts. The first part of the framework implements the PSN Extraction Subsystem by realizing the open stack of the Internet as illustrated in Section 2.1. The second part of the framework collects raw sensory and social network data from BSN and diversified services respectively and infers context values by analyzing the contents. The third part then maps the inferred contexts with a subset of services and social ties and populates them on a dynamically updated portal. Now we describe each part with the necessary hardware, software, protocols, and topologies we have chosen to implement.

5.1 PERSONAL SOCIAL NETWORK EXTRACTION SUBSYSTEM

In order to discover one's personal social network from the Internet, we leverage different layers of open stack detailed in Section 2.1. We now present the details of each phase of the implementation.

5.1.1 Open Stack for Personal Social Network Discovery

We have used Google email or Yahoo email as *Unique User ID*. In order to realize the *User Profile Metadata Discovery*, we have used the LRDD and XRDS¹³ library for PHP. We use Extensible Resource Descriptor (XRD) as simple XML-based schema for describing various social network resources. The XRD schema has two sections: the first one describes the resource itself and the second one describes the

¹³ <http://github.com/willnorris/php-xrd>

relationships to other resources. For example, the self-describing section includes information such as which API version the endpoint supports, while the relationships section can indicate where to find the address book of a social tie associated with the resource. We use Link-based Resource Descriptor Discovery protocol (LRDD) to associate a descriptor (in our case XRD) to the resource (for example, social network, authentication service) it describes. The LRDD is responsible for retrieving the links for a resource, using the various discovery methods. *Authentication* has been realized using OpenID single sign-on, which works with one's existing social network credentials such as Yahoo, Google, Hotmail, Facebook, Twitter, AOL or any OpenID provider. Among three existing open source platforms of single sign-on authentication mechanisms, JanRain¹⁴ looks most promising for the SenseFace framework than OpenSSO¹⁵ and Shibboleth¹⁶ because more than 80 per cent of the OpenID-enabled services support JanRain. Additionally, it is very lightweight compared to the rest.

An *access control* mechanism has been implemented using both 2-legged and 3-legged OAuth¹⁷ protocol. To unify the extracted *contacts* from different proprietary API formats such as Google's GData Contacts, Yahoo's Address Book and Microsoft's Live Contacts, we have implemented portable contacts using the RPX¹⁸ library. An android OS-based version of portable contacts has been implemented using jpoco¹⁹. In order to retrieve user's *attributes* as much as possible from the authenticating server such as full name, picture, birthday, country, gender, email, phone number, home address, nickname and height, we implement both OpenID Simple Registration Extension²⁰ and Attribute Exchange²¹. We have integrated several *Mashup* APIs such

¹⁴ <http://www.janrain.com>

¹⁵ <https://opensso.dev.java.net>

¹⁶ <http://shibboleth.internet2.edu>

¹⁷ <http://code.google.com/p/oauth-php>

¹⁸ <https://rpxnow.com>

¹⁹ <http://code.google.com/p/jpoco>

²⁰ http://openid.net/specs/openid-simple-registration-extension-1_0.html

as Apache Shindig²², Yahoo Pipes, Google Maps, Twitter, GoogleAJAXSearch, GoogleAppEngine, YouTube and Dipity²³.

The above-mentioned protocols and API's provide the ground of realizing different layers of the open stack. Because all of the above mentioned protocols and APIs are open source and most of them come with a PHP library, we have implemented them using a PHP framework called XAMPP²⁴, which acts as a web server. The installed XAMPP comes with a pre-built Apache web server 2.2.14, MySQL Database server 5.1.41 and PHP 5.3.1. As for the database support, we use MySQL which is built-in with XAMPP. In order to support the most popular social networks and existing service APIs, we have decided to support REST API [Fielding, R. T. (2000)]. 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 and it is light weight compared to SOAP messages. As shown in Figure 5.1, most of the current service providers now expose their services using REST interface. We have configured PHP modules as a XAMPP service cron that can be configured to automatically run at a specific interval to discover any update to a user's social tie and service list. In order to realize different layers of the open stack, we have configured three proxy servers with two of them having open IP that are not bound to any corporate firewall. The following are the configuration of the proxy servers:

- *Proxy server 1*: OS-Windows XP professional, RAM-1GByte, Processor-32 bit, 3.6GHz Intel Pentium 4, IP address-137.122.89.108, Java Version - jdk1.6.0.

²¹ http://openid.net/specs/openid-attribute-exchange-1_0.html

²² <http://shindig.apache.org>

²³ <http://www.dipity.com>

²⁴ <http://www.apachefriends.org/en/xampp.html>

- *Proxy server 2*: OS-Windows Server 2003, RAM-4GByte, Processor-32 bit, 3.1 GHz Intel Xeon with 4 processors, IP address-137.122.89.112, Java Version - jdk1.6.0.
- *Proxy server 3*: OS-Virtual Machine running Ubuntu 9.10 using vmware-server.2.0.2-203138, RAM-2GByte, Processor-3.1GHz Intel Pentium 4, Java Version - jdk1.6.0, Database-Postgresql 8.4.1.

The proxy server 1 runs Algorithm 1 and the proxy server 2 runs Algorithm 2 as described in Section 4.1.1. Both algorithms have been implemented using PHP. We use both AJAX and Reverse-AJAX for the web server and proxy server communication during different phases of the algorithms.

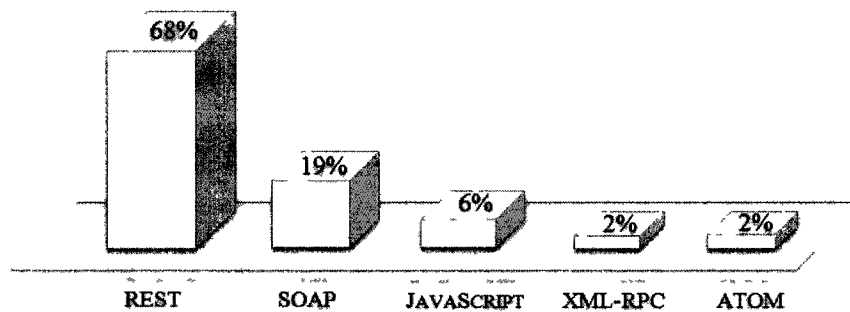


Figure 5.1 Popularity of existing API's exposed by existing Internet-based services (courtesy: ProgrammableWeb²⁵).

5.1.2 Personal Social Network Extraction

Algorithm 3, described in Section 4.1.2, has been implemented using PHP where each module is configured with the logic and protocols to connect to each service and extract the needed social ties from that service. For example, one module handles Facebook and the extraction of social ties from it, one module handles

²⁵ <http://www.programmableweb.com/apis>

Twitter, another handles Google mail and so on. Some of the developed services are third party web services and some are developed as part of this thesis initiative. The third party web services client uses either proprietary or open source APIs. As a proof of concept, the extraction algorithms have been configured to support the following services:

- *IM and audio/video networks*: Yahoo Messenger, MSN Messenger, Google Talk, SKYPE and ooVoo²⁶.
- *Social portals*: Facebook, YouTube, Flickr, LinkedIn, Twitter, Xanga, Last.fm, Mevio, Badoo, Faces, Xing, Friendfeed, MySpace, Friendster, Livejournal, Tagged, Orkut, Plaxo, Fdcareer, Hi5 and Vimeo.
- *Content management systems (CMS)*: WebCT, Moodle, Drupal, Wordpress, PhpBB, Alfresco²⁷, Joomla, Symfony and Drupal.
- *Social tagging*: Bookmarking tags from del.icio.us²⁸, photo tags from Flickr, scientific publication tags from CiteULike²⁹, tags describing people's goals and plans from 43Things³⁰ and tags containing description of highlighted contents from any webpage from Diigo³¹.
- *E-mail*: The e-mail servers that have been already integrated with the system are Live/Hotmail, OperaMail, Yahoo!, AOL mail, Gmail, and LYCOS mail. In order to collect more dataset from an institutional web mail server, we have implemented the Atmail³² server that we have configured as an IMAP server and imported the emails from MCRLab³³ mail server since January 2004 to date. We analyze email archives by observing direct recipient/CC/BCC list(s)

²⁶ <http://www.oovoo.com>

²⁷ http://wiki.alfresco.com/wiki/Main_Page

²⁸ <http://delicious.com>

²⁹ <http://www.citeulike.org>

³⁰ <http://www.43things.com>

³¹ <http://www.diigo.com>

³² <http://www.atmail.com>

³³ <http://www.mcrlab.uottawa.ca>

and their locations, frequency of communication, types of content/attachment shared, and types of events referenced within the email.

- *Information from online presence:* We use family ties in the form of GEDCOM³⁴. Using Google Social Graph API, which uses XFN³⁵, FOAF³⁶ markups and publicly declared connections.
- *Searching research networks:* We extract co-author lists within publication domains such as IEEE, ACM, Springer, and other indexing services such as DBLP³⁷ and CiteSeer³⁸. In addition, using indirect searches on Google Scholar Portal also provides publication ties. We extract research ties through conferences, online publishing and projects. We fetch co-author information in formats such as RDF, XML, N3³⁹, TriX⁴⁰, N-Triples⁴¹ and Turtle⁴².
- *Search from mobile devices:* We have implemented modules to extract social ties in the form of contact list and services' information in the form of to-do lists and calendar events from smartphone operating systems including Apple iPhone, Open Handset Alliance's Android and Symbian.
- *Health Services:* Google Electronic Health Record.
- *Location Services:* Google Maps, Yahoo Maps, Intel PlaceLab, Google Gears and real-time location update from the twitter posting by Lufthansa⁴³ web service, Friends on Fire⁴⁴.

³⁴ <http://en.wikipedia.org/wiki/GEDCOM>

³⁵ <http://gmpg.org/xfn>

³⁶ <http://www.foaf-project.org>

³⁷ <http://www.informatik.uni-trier.de/~ley/db>

³⁸ <http://citeseerx.ist.psu.edu>

³⁹ <http://en.wikipedia.org/wiki/Notation3>

⁴⁰ <http://swdev.nokia.com/trix/trix.html>

⁴¹ <http://www.w3.org/2001/sw/RDFCore/ntriples>

⁴² <http://www.w3.org/TeamSubmission/turtle>

⁴³ http://twitter.com/lufthansa_DE

⁴⁴ <http://apps.new.facebook.com/onfire>

- *Communication Services*: SMS, Fax, MMS, Voice mail, RSS and APRS.
- *Event Services*: Google/Yahoo/Microsoft Exchange calendar, Yahoo Weather, Smartphone calendar and OC Transpo⁴⁵ bus schedule event notification.

In order to index the discovery results, we have implemented open source Sphider⁴⁶ as *indexer*. The indexer has been deployed in the XAMPP server. We provide the results stored by the indexer to the SIENA model to obtain the relationship according to some category such as friends, family, profession, etc. The resultant dataset (such as the PSN) is then stored to a MySQL database. In order to visualize PSN we use Walrus⁴⁷ graph visualization tool.

5.2 CONTEXT EXTRACTION SUBSYSTEM

This subsystem first collects contents from both body sensor network and different services and then derives user context by analyzing the content. We now describe the implementation of the subsystem. The BSN comprises of a smartphone with several internal and external sensors (see Figure 5.2). We have chosen two types of smartphones. The first is Nokia N95 running Symbian S60 3rd Edition OS with 8GByte internal memory, and the second is HTC G1 Dream running Android 2.2 OS. We leverage the internal sensors of both of the smartphones to capture different sensory data. HTC G1 comes with the following internal sensors: orientation, accelerometer, light, magnetic field, proximity, temperature, audio and camera. The Nokia N95 has the following internal sensors: accelerometer, motion, camera, audio, tilt and magnetometer. As external sensors, we have used a LEGO NXT⁴⁸ brick, a

⁴⁵ <http://www.octranspo1.com/splash>

⁴⁶ <http://www.sphider.eu>

⁴⁷ <http://www.caida.org/tools/visualization/walrus>

⁴⁸ <http://mindstorms.lego.com/en-us/Default.aspx>

Garmin Forerunner⁴⁹, a Sun SPOT⁵⁰ WSN mote, and an EZ-430⁵¹ Chronos sports watch. The NXT brick is paired with touch, sound, light and ultrasonic sensors. We have replaced the NXT legacy firmware with leJOS NXJ⁵² Java firmware. The Garmin Forerunner 50 is attached with a heart rate monitor that provides the smartphone instantaneous, maximum and average heart rate in beats per minute. The EZ-430 measures acceleration, temperature and height of a person and together with a chest belt, we can measure the heart rate variability and the speed. The Sun SPOT WSN mote comes with three-axis accelerometer, temperature and light sensor.

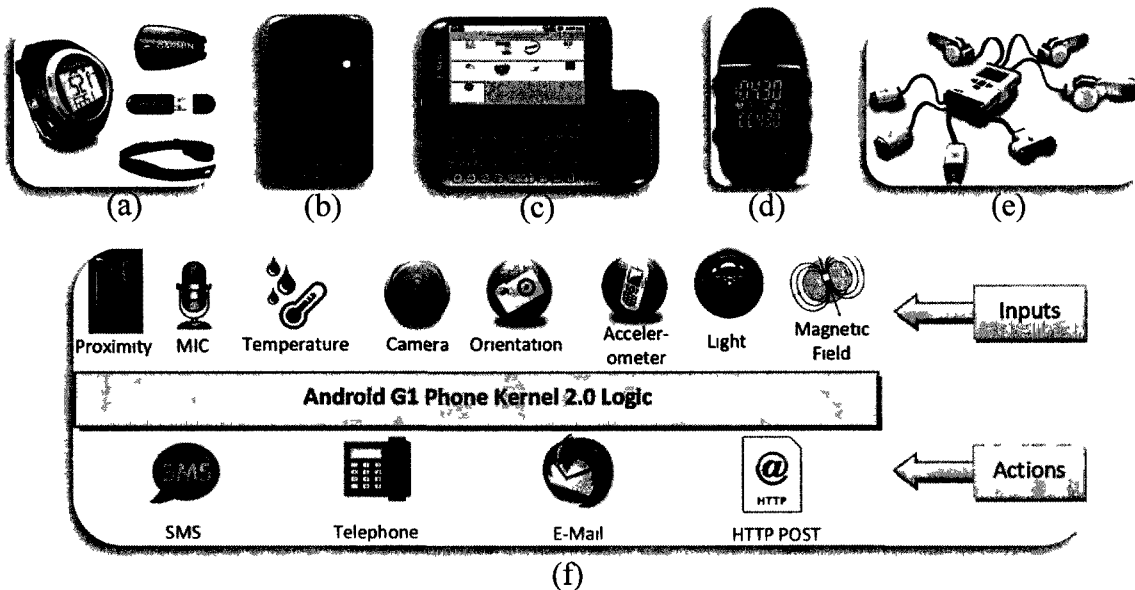


Figure 5.2 Body sensor network (a) Garmin Forerunner heart beat sensor, (b) Sun SPOT WSN mote, (c) An HTC dream running Android 2.2, (d) EZ-430 Chronos sports watch, (e) NXT brick including touch sensor, sound sensor, light sensor and ultrasonic sensor, and (f) Built-in sensors and communication channels comes with the HTC smartphone.

⁴⁹ <http://www8.garmin.com/buzz/fr50>

⁵⁰ <http://www.sunspotworld.com/>

⁵¹ <http://processors.wiki.ti.com/index.php/EZ430-Chronos>

⁵² <http://lejos.sourceforge.net>

We have used Symbian C++ for low level and J2ME for high level application programming for the Nokia N95 smartphone. The Android-based application has been developed using Java programming language. As shown in Figure 5.2(f), the Android 2.2 comes with some built-in communication facilities including SMS, telephony, email and HTTP POST method. We have ported the i-jetty⁵³ personal web server within the HTC G1 phone to communicate with the external servers. We follow RESTful architecture to define the access pattern of each sensor. For example, the REST interface to access temperature data using HTTP GET method can be as following:

<http://137.122.89.108/SenseFace/REST/resources/people/arahman/sensors/temp>

The result of a query is pushed back to the inquirer in either XML or JSON format. A response in JSON format is shown below:

```
{
  restResponse: {
    sensorType: 'Temperature',
    locationCaptured: {
      latitude: 45.415338,
      longitude: -75.671339
    },
    unitOfMeasurement: 'C',
    value: 20,
    timeStamp: '23Apr09-1137'
  }
}
```

Figure 5.3 A JSON response from a temperature sensor as a result of a RESTful query from a remote service.

⁵³ <http://code.google.com/p/i-jetty>

Using AJAX, the smartphone web server can push sensory data to the *context extractor* server, while using reverse AJAX the BSN receives remote server pushed data over the web, which allows a server to push events to the smartphone web server. An implementation scenario has been illustrated in Figure 5.4 where a BSN establishes two-way communication with a member of one's COI via a remote server. The BSN context extraction service module has been implemented using an XAMPP server which stores the sensory data primitive in a MySQL database. Finally, the context extraction logic has been developed using different third party applications written in Perl, Python, Java and PHP programming language.

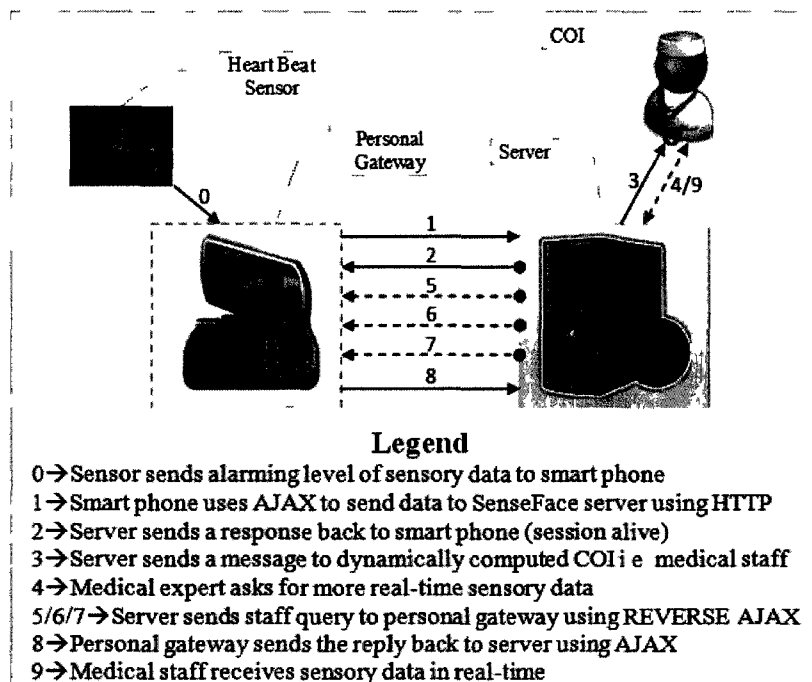


Figure 5.4 Two-way sensory data communication between a BSN and COI at a given physiological context.

The 3A modeling has been realized using PHP programming language. The *Web Data Extractor* is implemented using XAMPP server. As part of the Internet-based services, we have deployed separate service modules within the *Web Data Extractor* as described in Section 5.1.2. To aggregate contents from different services we have adopted Enterprise Mashup Markup Language (EMML⁵⁴). As for the view, the framework exposes service interfaces in REST, RSS, and Atom format that can be consumed on Wiki pages, blogs, websites, portal servers, emails and faxes, or can be called directly from a smartphone. The context extraction modules have been implemented using Konstrukt⁵⁵. The context topology shown in Figure 3.3 has been implemented using EMML. The fuzzy logic modules have been implemented using Python library called FuzzPy⁵⁶.

5.3 DYNAMIC SERVICE AND COI ADAPTATION SUBSYSTEM

Web portal has been implemented using open source iMoogle⁵⁷, which uses PHP and JSON to dynamically control the layout and content of each portlet. iMoogle portal is based on JavaScript and the number of portlets, their layout, spatial location and the size of each of them can be dynamically changed by server side PHP script. The portal uses two metaphors; one is grid-based (see Figure 5.5(a)) and the other is based on a sliding window (see Figure 5.5(b)). We provide an authoring interface to facilitate a user in associating a service with context(s) and thereby assign credit to each service. Currently we have developed portal interfaces for both desktop browsers as well as smartphone browsers. COI explorer has been implemented using prefuse⁵⁸ visualization toolkit.

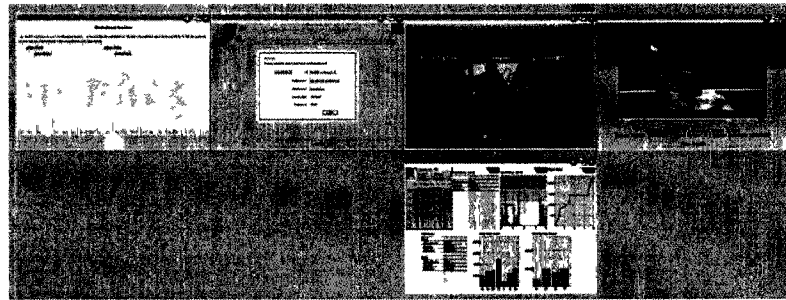
⁵⁴ <http://en.wikipedia.org/wiki/EMML>

⁵⁵ <http://konstrukt.dk>

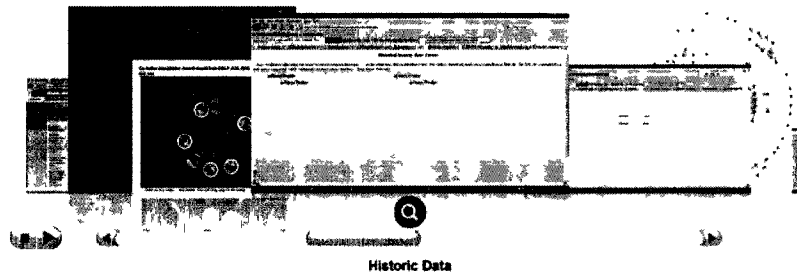
⁵⁶ <http://code.google.com/p/fuzzpy>

⁵⁷ <http://www.moonkiki.com/moonkiki/imoogle>

⁵⁸ <http://www.prefuse.org>



(a)



(b)

Figure 5.5 Portal to visualize context-aware social ties and services with two metaphors (a) grid-based layout metaphor and (b) sliding window metaphor.

The following is a scenario when a user logs in to the portal and visualizes dynamically calculated context-aware services and social ties. We assume a user uses Google email as single sign-on. After Google email server authenticates the user, the portal sends a request to the *Service Layout Manager* component to render the context-aware services. The dynamic service selection and community of interest has been implemented using PHP modules that dynamically creates a JSON file containing the COI and the services with spatial layout and size of each portal, and the list of COI with available contextual information such as location of each user, current activity, etc. Figure 5.6 shows how a relative change in a user's context is reflected within the portal.

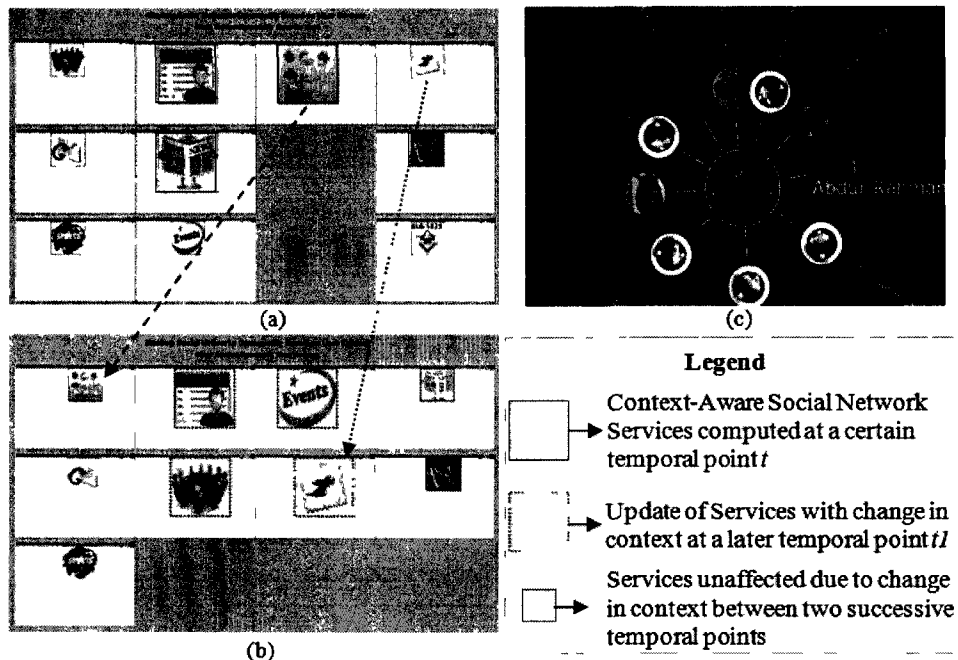


Figure 5.6 A web portal showing social network services (a) services with large rectangular icon are services that need immediate attention from the user at a given context, (b) updated user context dynamically updates the portal content as well as layout, and (c) an instance of context-aware retrieval of COI visualized in flash interface.

5.4 SUMMARY

In this chapter we have elaborated the implementation of each subsystem. We have described the hardware, sensors, software architectures, smartphones, operating systems, databases, web server, web portals, web services, programming languages, third party and our own developed APIs, protocols, schemas and other relevant technologies within each subsystem.

CHAPTER 6: TEST RESULTS

We have tested individual components of each subsystem shown in Figure 4.1. The testing has been performed over a 16 month period from December 2008 to June 2010. Tests have been conducted at different times of the day to accommodate varying location, Internet and cellular network traffic conditions. The test results reveal several factors: first, how well the *PSN Extraction Subsystem* is able to collect the total set of services and social ties' information. Second, the performance of the context extraction subsystem is primarily dependent on the live content extraction from both sensory media and diversified services. Hence, the better and less noisy the contents are, the better the *Context Extraction Subsystem* will perform. Finally, depending on the semantic mapping process, the subset of the social ties and services are selected, which is rendered dynamically to the portal. Hence, we measure the delay in finding current context, mapping context-aware services and COI and rendering them dynamically to a portal.

The test results have been arranged into three sections. Section 6.1 shows the run-time performance of the framework where we collect several metrics such as delay, number of successful and failed requests from the interactions between physical users and the framework. Section 6.2 shows the outcome of the benchmarking results that were carried out to measure some more metrics such as scalability and concurrency. Finally, Section 6.3 presents the subjective evaluation of the framework. We set up a demo test bed of the proposed framework prototype having salient features. We then conducted a usability test from volunteers of different age, profession, gender, and ethnicity followed by filling up of a questionnaire. This section presents detailed analysis of the user feedbacks.

6.1 RUN-TIME PERFORMANCE STUDY

SenseFace framework's context-awareness depends on the live content coming from both BSN and Internet-based services. Contents extracted from services affect the framework in twofold. First, the PSN of each subject is constructed and stored in the database. Second, content extracted from the services are fed to the context extraction subsystem in real time. Similarly, content extracted or pushed from the BSN are sent to the context extraction subsystem for further processing.

6.1.1 Service and Social Tie Access

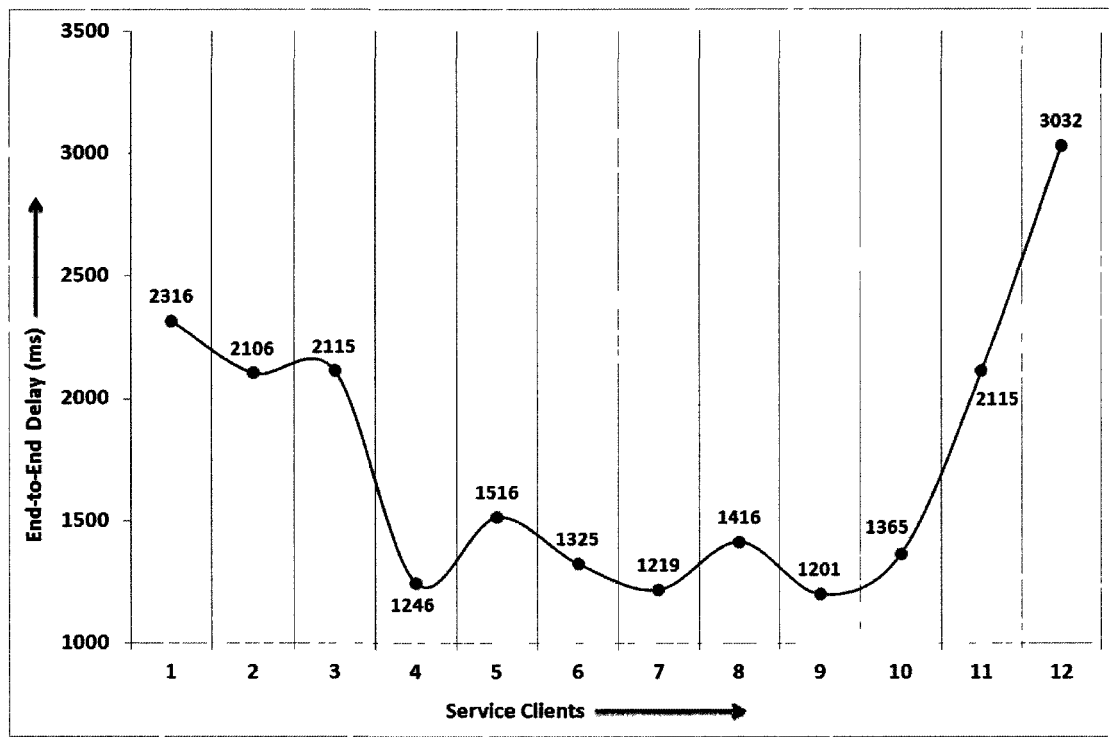
In order to test the PSN subsystem, we first attempt to measure how the *Service Clients* module works. As discussed earlier, the *Service Clients* host the client modules of each service provided by SenseFace framework. Each service client attempts to communicate with the actual service provider by utilizing all or a combination of open stack layers. Each service client holds the definition of APIs and methods to access the actual service. The issues with the *Service Clients* are twofold: first, initializing each service client with the right protocols, and second, attempting to communicate with the actual service provider. We tend to measure two metrics in this process: number of failed attempts and the end-to-end delay in accessing services. The former affects the context extraction subsystem because a particular context primitive of a service depends on the content (for example, location context depends on the successful extraction of a location data from location service client. The latter affects both the context extraction subsystem and the rendering of the layout by the *Service Layout Manager*.

Service access delay

Measuring the delay in accessing a particular service plays crucial role because the context extraction as well as the dynamic service and COI rendering subsystem

need to dynamically extract the live content. We calculate the end-to-end delay in accessing and downloading content from a service provider as follows:

Round Trip Delay of Extracting Content from a Service = (The delay in initializing the appropriate service client + instantiating appropriate protocol stacks from the open stack + negotiating with the actual service provider + accessing and parsing the appropriate content extraction API + storing the results in the indexer)



LEGEND

1-Yahoo Messenger, 2-Facebook, 3-WebCT, 4-del.icio.us, 5-Gmail, 6-GEDCOM, 7-DBLP, 8-contact List, 9-Google EHR, 10-airline location tracking, 11-SMS, 12-OCTranspo bus schedule tracking

Figure 6.1 Seven samples of end-to-end delay of 12 randomly selected services.

Figure 6.1 shows the maximum end-to-end delay of 12 randomly selected services from each category listed in Section 5.1.2. We have taken 2,000 samples of end-to-end delays for each of the 12 services shown in Figure 6.1. Although some services fall within the category of COS, the delay only considers accessing and

downloading the content only, not the social ties. The delay values shown in Figure 6.1 are consistent with the delay values captured from similar test environments such as in [Bottazzi, D. et al. (2006)]. Next we measured the maximum and average time in extracting social ties from community oriented services (COS), such as the services that link a subject with his/her social ties (as described in Equation 6).

Social tie extraction delay

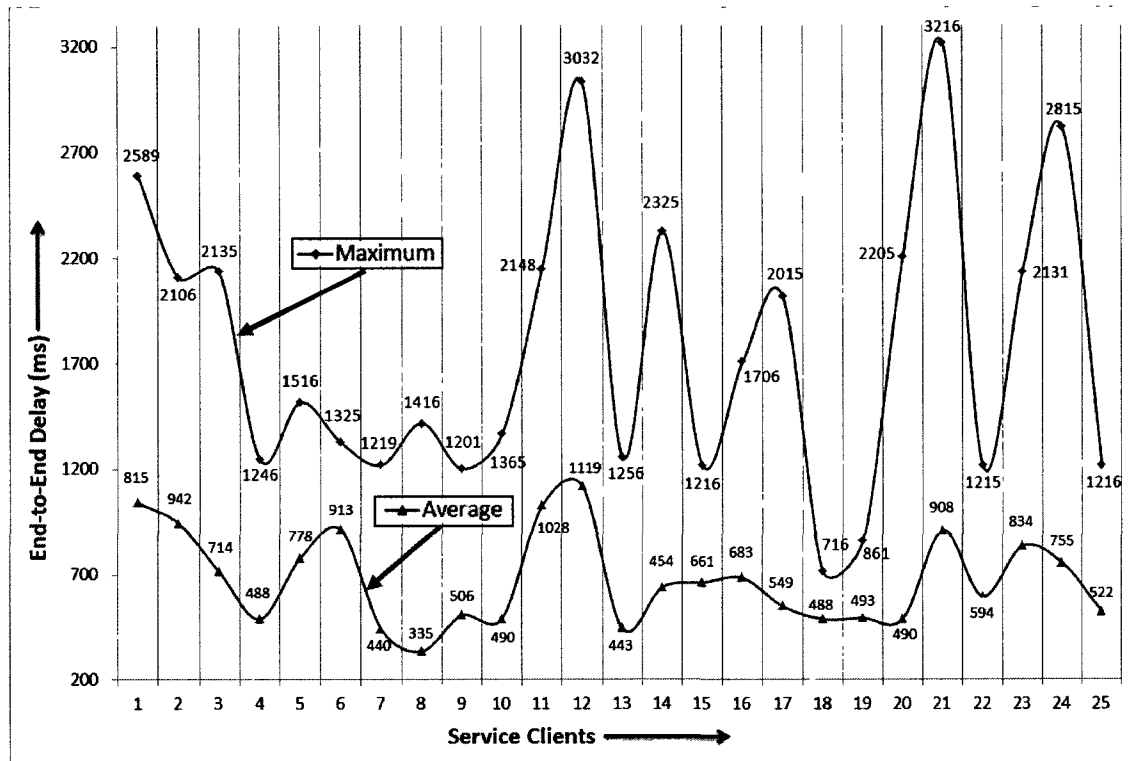
Social ties can be extracted through relevant services at the expense of some additional processing and delay on top of the service access delay. The average end-to-end delay in can be calculated as follows:

Avg. Round Trip Delay of Extracting a Social Tie from a COS = *(The delay in initializing the appropriate service client + instantiating appropriate protocol stacks from the open stack + negotiating with the actual service provider + accessing and parsing the appropriate social tie extraction API + storing the results in the indexer) / N*

Where N is the total number of test instances from physical users. In order to calculate the social tie extraction delay, we have selected 25 services that are categorized as COS. For each of the 25 services, we calculate the average end-to-end delay from 2000 test instances, for example $N=2000$. Figure 6.2 shows the maximum (upper curve) and average social tie extraction delay (lower curve) of the 25 COSs.

If we compare Figures 6.1 and 6.2, we can deduce some conclusions. For example, accessing Facebook (service client # 2 of Figure 6.1) took a maximum of 2,106 milliseconds while extracting social ties from Facebook (service client #1 of Figure 6.2) took a maximum of 2,589 milliseconds. We assume that the extra delay was taken by accessing the social ties on top of accessing the service itself. After analyzing other services, we have noticed similar results. In an initiative to capture the

failed attempts, we have found 741 failed attempts from $2000 * 25 = 50,000$ samples, which is below 0.015 per cent.



LEGEND

1-Facebook, 2-YouTube, 3-Flickr, 4-Alfresco, 5-Twitter, 6-Orkut, 7-Livejournal, 8-MySpace
 9-43Things, 10-Last.fm, 11-Google Talk, 12-DBLP, 13-Skype, 14-WebCT, 15-Wordpress
 16-Gmail, 17-Atmail, 18-Yahoo Mail, 19-Hotmail, 20-FOAF, 21-Google Scholar Portal
 22-Contact List, 23-Google calendar, 24-Yahoo calendar, 25- Microsoft Exchange calendar

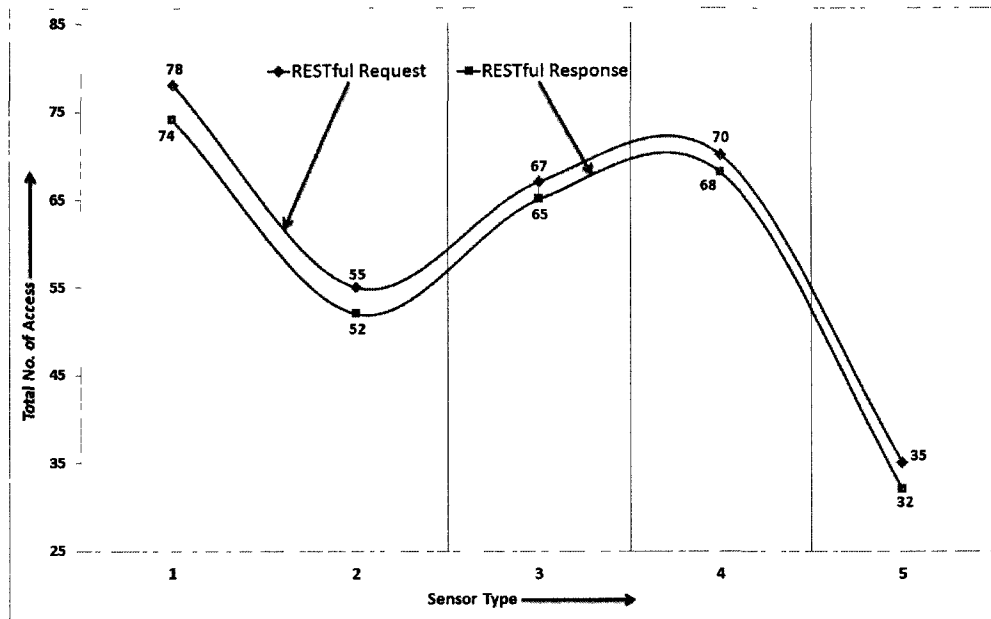
Figure 6.2 Social tie extraction delay: Maximum and average delay of 25 randomly selected community oriented service clients.

The failed attempts might be caused by several reasons. First, in some instances the actual service provider becomes inaccessible due to server maintenance or other issues. In these circumstances, we bypass the service accessing to avoid waiting, and rather scan the rest of the services. Second, service clients have been

implemented using different programming languages such as .NET, Python, Perl, Java and PHP. Each service runs as a daemon that is called by web server. The daemon might fail due to server overloading or any particular compiler's non-responsiveness. Third, the service clients may temporarily not have Internet access. Fourth, a particular service client might fail to initialize appropriate open stack modules with proper user profile parameters. There might be other internal or external factors responsible for the failed cases that need to be examined in future work.

6.1.2 Sensory Data Access

Sensory data plays a crucial role in providing a subject's physiological and ambient context. We tested the body sensor network in both upstream and downstream cases. In order to test the upstream sensory data communication, we defined some activities and associated sensory data threshold that should trigger a data push to a remote server. We tried to simulate some events shown in Tables 4.1 and 4.2 to find out the performance of the sensory data push. For example, to test the heartbeat-related event, we assumed that when a person runs, his/her heart rate increases, which is also a sign of a person's high heartbeat rate under alarming situations (the heart rate increases beyond 180 beats per minute). As for the upstream communication, up to 98 per cent of time, the framework was able to push sensory data to the *BSN Context Extractor*. We attributed the 2 per cent of failure to the poor data coverage by the cellular networks while a person is roaming. To test the downstream data communication, we tried to find different sensory data in a RESTful way. Figure 6.3 shows the total number of RESTful requests sent from the *Downstream Controller* of *BSN Context Extractor* and the total number of correct responses received from the *Sensory Data Extractor*. The sensory data access poses some delay between a sending a request and receiving a valid response with correct payload. Table 6.1 shows the average delay of the instances of successful responses that are shown in Figure 6.3.



LEGEND

1-Temperature, 2-GPS, 3-Motion, 4-Accelerometer, 5-ECG

Figure 6.3 Downstream sensory data access by COI.

Table 6.1: Sensory data access round-trip delay between a BSN and members of COI.

Sensor Type	#Successful REST Response	Avg. Delay (s)
<i>Temperature</i>	74	1.55
<i>GPS</i>	52	3.25
<i>Motion</i>	65	1.34
<i>Accelerometer</i>	68	1.65
<i>Heart Rate</i>	32	1.23

The delay shown in Table 6.1 refers to round trip delay and calculated as follows:

Avg. Delay = downstream processing delay (at web server) + downstream network delay (Wi-Fi/cellular) + processing delay (at smartphone 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 sometimes took more time than Wi-Fi or cellular-based location tracking techniques. We identified several reasons of the miss ratio. One: how the smartphone and the sensors are aligned and the distance between them has an impact on channel loss ratio because the sensors are wirelessly connected with the smartphone using Bluetooth. Two: the mobile OS puts several restrictions on background user written applications while the smartphone is in sleeping or conversation mode. Three: in many cases, the pairing between a sensor and the smartphone became distorted and thus lost the I/O request. Four: a subject was outside of coverage of both Wi-Fi and cellular data network while he/she was mobile.

For example, while the subject was entering the University of Ottawa campus, the mobile OS switched to Wi-Fi mode. A request for instantaneous heart-rate from a member of COI would result in missed I/O as the University of Ottawa requires VPN to connect to the Internet. Another scenario caused the loss in I/O while the subject was driving to an area where cellular network coverage was poor or absent. Finally, in the case of location tracking we used cell triangulation, Wi-Fi triangulation and GPS signals. Where Wi-Fi and cell triangulation was not present and the smartphone had to resort to GPS signal, it took about one to two minutes to get a GPS fix (depending on the available satellites). Frequent mobility of a subject from indoor to outdoor or vice versa is the major cause of this location-related delay.

We have attempted to relate the miss-ratio shown in Table 6.1 with the above mentioned five causes. Out of 14 missed I/O incidences, five instances of failure were identified as University VPN issue, three were reported as Bluetooth connection loss between the sensor and the smartphone, two were caused by Wi-Fi or cellular data coverage problem, two were lost while the person was in conversation mode, and the remaining two were caused by other unidentifiable causes that 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. Some of the recommendations are as follows:

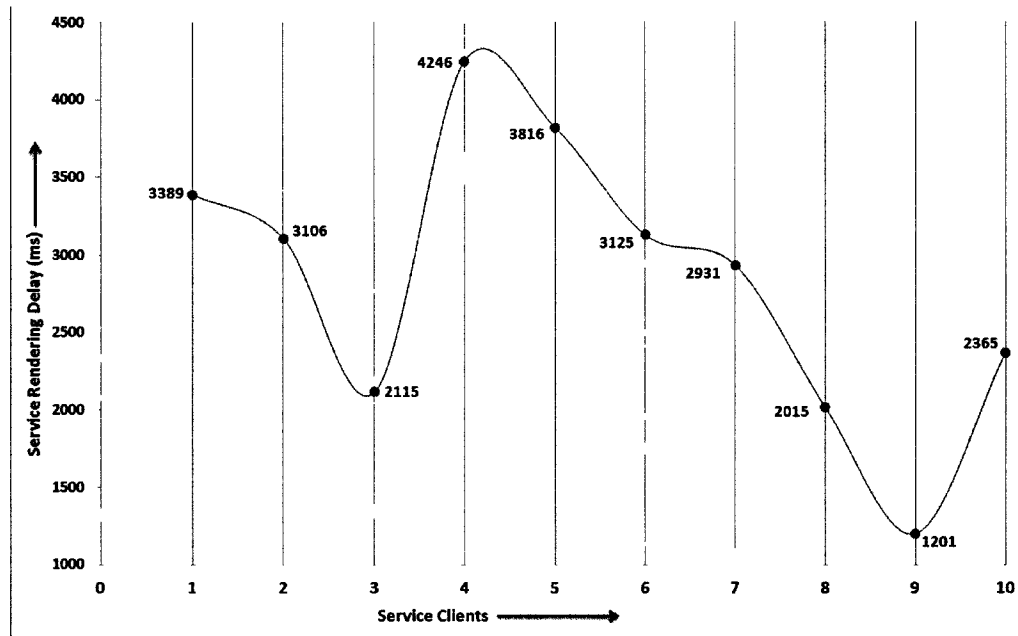
- 1) Open source mobile OS, such as Google Android, gives a developer greater control over underlying applications and their native access to phone hardware.
- 2) A pro-active location- and context-aware user interface can be embedded with the framework which will alert a subject when he/she is about to go somewhere with weak or no Wi-Fi or cellular data coverage.
- 3) Choosing better sensor hardware and spatial location with respect to the smartphone.

6.1.3 Dynamic Services and COI Rendering

Upon reception of the content from both BSN and PSN, the context extraction sub-system passes the content to right module, generates high-level context primitives from low level raw context data, aggregates them, and finally stores the context metadata in the context repository. This exerts a content-to-context processing delay. Once the user context is pushed to the *Ubiquity Stack*, it dynamically maps a subset of services and COI and assigns credits to each service to be used for portal layout purpose. The layout manager needs to normalize the credit values of each service and dynamically allocates each service a portlet with size and location. The following figure shows the end-to-end delay of 10 randomly selected context-aware services.

The delay shown in Figure 6.4 is calculated as follows:

Delay of Dynamically Adapting to Context-aware Services and COI = (The delay in initializing the appropriate service clients + instantiating appropriate protocol stacks from the open stack + negotiating with the actual service providers + extracting contents as payload + extracting context from the downloaded contents + inferring complete context and finding COI and services based on context + updating priorities of each service and generating dynamic portal layout).



LEGEND

1-Youtube, 2-Google Maps, 3-Google EHR, 4-Skype, 5-Facebook
 6-YahooMessenger, 7-Google Calendar, 8-Heart Beat, 9-ooVoo, 10-Wordpress

Figure 6.4 Average delays in milliseconds in rendering a typical instance of several services and associated COI using the proposed framework.

Figure 6.4 shows an instance of delay of rendering ten services to the iMoogle portal interface taking into account the above factors. A typical instance shown in the figure portrays that all the services and associated COI were dynamically derived within approximately five seconds. Run time test results presented above only considers that an individual user is interacting with the framework at any given moment. To make it more realistic and closer to practical assumption, we tested the framework with large number of users (synthetic) with many of them accessing the framework concurrently. This is particularly useful to test whether the framework can scale well and maintain user states, sessions, and personalized contexts. In the next section we provide the evaluation of the proposed framework with large number of users and simultaneous accesses.

6.2 CONCURRENCY

In this section, we provide the test results of concurrent access to the framework. Because social network frameworks are subjected to heavy and unpredictable concurrent access, we have laid out the following test bed. Up to 1,000 concurrent HTTP requests with session-alive features is posed to the SenseFace framework during different times of a day. Each instance of the test session does the following jobs consecutively while all the concurrent accesses are run in parallel: 1) it creates a new user to the database, 2) it creates some random contexts for the user, 3) it assigns some random number of services and members of COI to that user and saves to the database, and 4) it logs in the SenseFace framework with the user name and generates a log file.

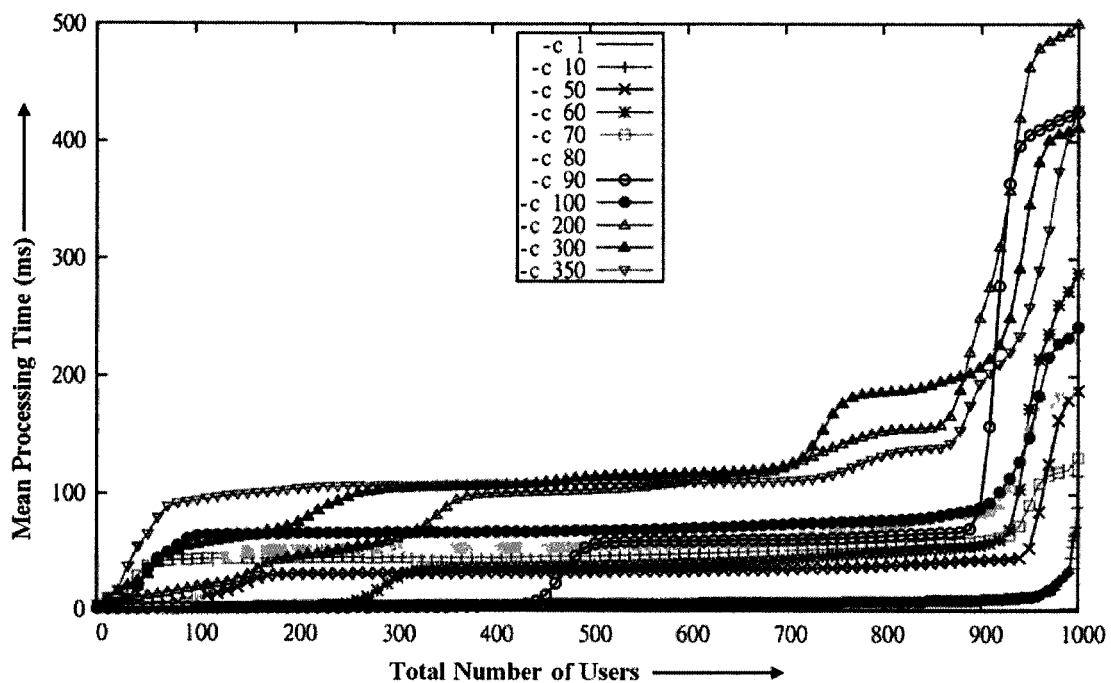


Figure 6.5 Mean processing time of accessing SenseFace framework by 1000 users for 49.320 seconds.

The script logs some important concurrency parameters. For example, when a physical user creates his/her context profile with SenseFace, the framework creates a separate folder in the server side which stores and physically separates the users preferences as well as personalized choices regarding services and COI in every unique contexts. The benchmarking concurrency test has been performed using Apache 2.2 HTTP benchmarking tool⁵⁹ with HTTP KeepAlive feature enabled. The results are meant to provide concurrent support of user sessions and context states. Figure 6.5 shows the test result of total 1,000 users with 1, 10, 50, 60, 70, 80, 90, 100, 200, 300, and 350 users concurrently accessing the framework without compromising the state, session and context of each user.

Figure 6.5 shows the mean processing time (waiting time + establishing connection + processing time) in milliseconds. The figure shows total 11 concurrency levels with the markers shown inside the top-center box where `-c` switch refers to concurrency value. For example, “`-c 350`” refers to the curve for minimum 350 simultaneous access to the framework for the duration of 49.320 seconds. Please note that Figure 6.5 does not actually take into consideration the delay of consuming service contents from their respective parent server. It only creates the sessions, contexts and PSN dataset of each user and logs in the framework as software agent/robot. From the observed data we conclude that for the no-concurrency level i.e. “`-c 1`”, the processing time is almost unnoticeable. As the number of concurrent accesses increases, the processing time increases accordingly. It is noteworthy that the sharp rise of mean processing time that is experienced after the concurrent number of users reaches more than 700 is due to limitation of the hardware test bed, operating system, processor speed and the benchmarking tool capability. In the presence of better simulation environment, the algorithm is expected to handle even larger number of concurrent users.

⁵⁹ <http://httpd.apache.org/docs/2.2/programs/ab.html>

However, the highest processing delay is seen to be below 500 milliseconds, which is very much appreciable for the framework to handle a large amount of traffic. Figure 6.5 only shows up to concurrency level “-c 350.” In order to see the effect of traffic for even higher concurrency levels and more data, we have captured them via Table 6.2. Table 6.2 shows details about seven distinct concurrency levels, starting from 400 to 1,000 simultaneous KeepAlive HTTP sessions to the SenseFace framework. For example, during test one, the tool continued to send a minimum of 400 requests to the framework for duration of 120 seconds with a mean value of 76.84 numbers of requests per second. During the 120 seconds testing period, 10 requests failed which is pointed out by failed request field. The total processing time in milliseconds is broken down into connection establishment with the framework, processing the request and waiting time followed by total time. Each time field is further categorized into minimum, mean, standard deviation, median and maximum values. The final column shows the number of requests in percentage value and the time in milliseconds took by the framework to handle that amount of request.

For the sake of brevity, we have provided the serving results of 50, 75, 95 and 100 per cent requests only. The results shown in Table 6.2 only consider accessing one service from the category of *Service without Social Ties*. The highest number of failed requests was recorded as 10, which is very positive. The minimum processing time for each of the seven concurrency levels of ascending order was gradually increasing from 92 to 164 milliseconds. The total mean values varied for test one to test seven from 5,021 to 8,412 milliseconds. The standard deviation, median and maximum valued varied from 2,175.9 to 4,173.2 milliseconds, 5,165 to 9,830 milliseconds and 7,683 to 12,973 milliseconds respectively.

Table 6.2: Concurrent access to one service without social tie class of service by 1000 users for 120 seconds.

Server hostname:	137.122.89.108
Server port:	80
Document Path:	/SenseFace/benchmark/index.php

Test#	Concurrency Level/Total Users	Failed Requests	Total Processing Time (ms) (Connection+ Processing+ Waiting)					Number of the requests served within a certain time		
			Min	Mean	SD	Median	Max	Request (%)	Time (ms)	
1	400/1000	10	Connect	0	3	35.3	0	502	50	5165
			Processing	92	5018	2177.2	5165	7683	75	7670
			Waiting	90	3642	1721.5	2616	7677	95	7680
			Total	92	5021	2175.9	5165	7683	100	7683
2	500/1000	9	Connect	0	4	47.3	0	994	50	5690
			Processing	104	6046	2556.3	5208	10249	75	7714
			Waiting	104	3918	1799.3	3160	7711	95	10232
			Total	105	6050	2556.4	5690	10249	100	10249
3	600/1000	9	Connect	0	4	47.3	0	1005	50	7685
			Processing	108	6739	2953.7	7685	10237	75	9720
			Waiting	108	5195	2375.1	5123	9739	95	10216
			Total	108	6742	2953.0	7685	10238	100	10238
4	700/1000	9	Connect	0	4	47.4	0	1000	50	7767
			Processing	119	7453	3458.0	7767	12843	75	10293
			Waiting	119	5316	2342.3	5182	7761	95	12835
			Total	119	7456	3457.1	7767	12843	100	12843
5	800/1000	9	Connect	0	4	47.5	0	1003	50	8811
			Processing	128	8038	3862.2	7836	12901	75	10368
			Waiting	128	5342	2353.9	5201	7831	95	12892
			Total	128	8041	3860.8	8811	12901	100	12901
6	900/1000	8	Connect	0	5	57.8	0	1006	50	9829
			Processing	144	8343	4121.9	9828	13019	75	12968
			Waiting	144	6544	3427.5	7240	10502	95	13013
			Total	145	8348	4118.7	9829	13019	100	13019
7	1000/1000	9	Connect	0	4	47.3	0	1000	50	9830
			Processing	163	8408	4174.8	9830	12972	75	12909
			Waiting	163	8367	4180.9	9784	12960	95	12955
			Total	164	8412	4173.2	9830	12973	100	12973

From the analysis, the highest processing delay was found to be 12,973 milliseconds in the worst case scenario when the server was fully loaded. The test results are encouraging in that they show promising prospect to be deployed to a real web environment as we tried to make the test environment as real as possible.

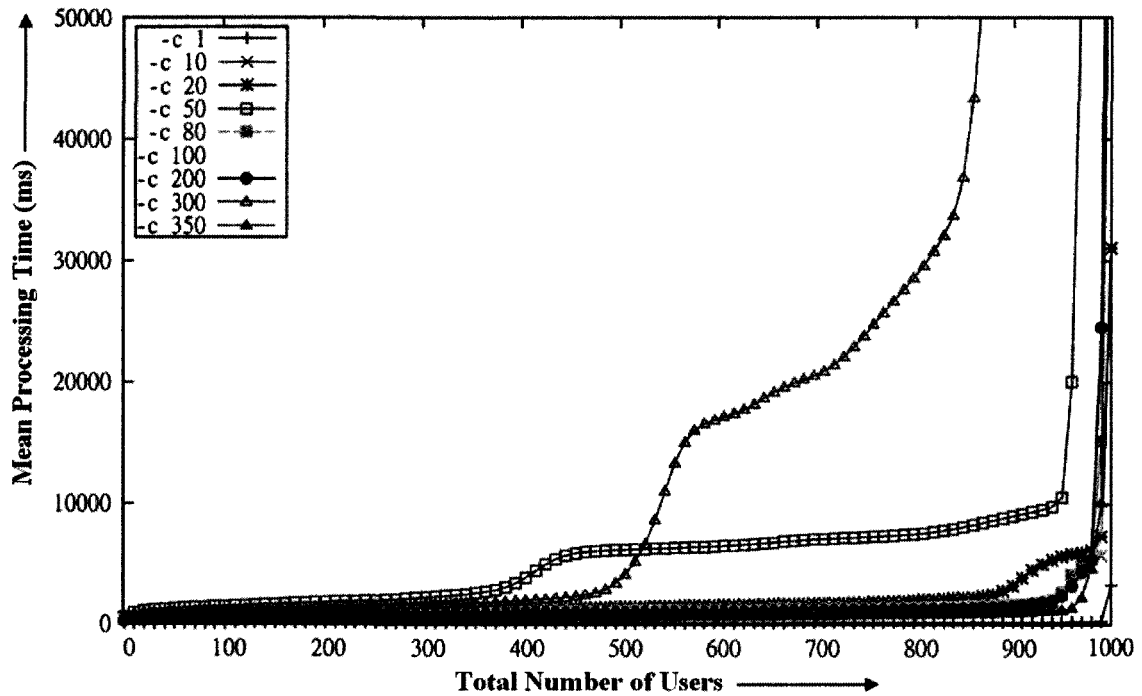


Figure 6.6 Result details of total 1000 users accessing up-to ten randomly selected services at any given moment for 53.16 seconds.

We then conducted a benchmarking test by choosing a random number of services (between one and 10 from a pool of 25 services as shown in Figure 6.2), where each request takes into account the delay in accessing content from a respective third party web servers and extracting context values. To minimize the impact on the server side scripts, each service is accessed in parallel using `curl_multi_init`⁶⁰ functionality. Figure 6.6 shows the usage data of 1,000 users accessing the framework

⁶⁰ <http://php.net/manual/en/function.curl-multi-init.php>

and consuming random number of services at any given context. Depending on the type of service used, either the service clients of the framework or the actual service provider may experience heavy bandwidth issue and might cause unpredictable waiting and processing time. For example, in the case of concurrency level 300 (“-c 300” in Figure 6.6), the framework shows dramatic increase in processing time, which is above 20 seconds, after the total number of users go beyond 500. The next case is “-c 50”, which shows increased delay after the total users cross 400. We cite the same explanation as deduced in the case of Figure 6.5 that by providing adequate hardware and software resources, the same algorithm can scale to larger number of concurrent user access. In the next section we provide the subjective evaluation of the proposed framework.

6.3 USABILITY

This section provides subjective measures, such as users’ feedback collected from questionnaires. The evaluation considered aspects such as framework access and layout, technical functionalities and user satisfaction. These measures were obtained by the combination of logging data as a result of user interaction with the framework and the subsequent completion of online questionnaires. The subjective evaluation was based on a sample of 35 test persons with about 25 per cent being female. The evaluation was conducted between May 15, 2010 and June 30, 2010. The majority of the users belong to an age group older than 25 with about 10 per cent under 25 years. All test users used various Internet-based services in different scale during the past four to six years. About 20 per cent of the test users were non-technical and they used the Internet sparingly only at their home desktops or laptops.

Table 6.3: Usability and user satisfaction questionnaire (adapted from ISO 9241-11⁶¹).

Questionnaires
<p>Q1. <i>I appreciate the concept of context-aware services and community of interest. If my privacy is protected, I might consider using the framework.</i></p> <p>Q2. <i>I communicate with my social ties through emails, instant messages, and mobile phones and so on.</i></p> <p>Q3. <i>I consume different categories of services such as emails (Hotmail, Gmail, Yahoo...), instant messaging (MSN, Skype...), social networks (Facebook, Twitter...), audio/video conferencing (Skype, Google Talk...), checking weather, checking bus schedule and so on.</i></p> <p>Q4. <i>I appreciate that I do not have to login at each service individually to check its current content.</i></p> <p>Q5. <i>The framework collects user context from different categories of services (e.g. email, calendar...) in real time. Do you agree with that?</i></p> <p>Q6. <i>The delay/waiting time in showing the services within the portal/home-page is within your tolerable limit.</i></p> <p>Q7. <i>The orientation/presentation of the dynamically available services based on the context value is quite interesting and I liked the idea.</i></p> <p>Q8. <i>I like the concept that the framework checks the content of all my registered services and extracts the context value.</i></p> <p>Q9. <i>Incorporating sensors to extract important context information (e.g. high heart rate, room and body temperature) looks appropriate to me.</i></p> <p>Q10. <i>I am happy with the idea of categorizing the services according to the type of services (e.g. email, video conferencing, calendar, and sensory data).</i></p> <p>Q11. <i>You are at your work and engaged in a meeting. You only allow certain services until your meeting is over or you manually permit a particular service or certain members of your COI. These kinds of scenarios portray the benefit of adding context-awareness to my services.</i></p> <p>Q12. <i>It is useful to see the list of all my social ties organized according to my social categories (e.g. family, kin, friends, colleague...).</i></p> <p>Q13. <i>I like the idea of organizing my social ties based on the services (e.g. Gmail, Yahoo, Skype, Facebook) through which I communicate with them.</i></p> <p>Q14. <i>It is interesting to show a subset of my social ties (community of interest) at any given context (e.g. location: home and/or event: meeting and/or time: evening).</i></p>

Before the test began, the subjects were first given a brief introduction of each of the functionalities they had to test and provide their feedback for. Then, each user

⁶¹ <http://zing.ncsl.nist.gov/iusr/documents/cifv1.1b.htm>

was requested to test the framework at different locations and time for the duration of their choice. Some took several days to test different contexts while some finished the test with one or two distinct contexts within a day. Finally, each subject was requested to complete an online web-based questionnaire containing 14 questions (shown in Table 6.3). Each question had five possible answers (“Strongly Agree,” “Agree,” “Not Sure,” “Disagree” and “Strongly Disagree”).

As the questionnaires were online, the users had the option to provide their feedback regarding their test experience at any moment of a day from their home or any other location over the Internet. Because user privacy is out of scope of this research, providing login information about services such as Facebook, Gmail, Skype and so on was kept optional for the users. To facilitate the testing process, we created demo accounts for each of the services and kept them operational for about a year prior to this usability test. This way, users could see the content of each service and its associated social ties. Some users provided their login information with the assurance that current testing infrastructure would not store any personal or login information in the server. Some users needed special assistance due to their age and inability to either see properly or read fonts or understand some user interfaces. Proper onsite help was arranged for those accessibility issues.

The approach of the usability test described below consists of capturing some variants of the proposed context-aware framework prototype. All the variants were tested in two modes: simple and advanced. In the simple mode, the portal showed only those services and COI that a user defined for a particular context a priori. However, the services were dynamically positioned on the portal interface based on user preference. In the advanced mode, not only the services chosen by the user at any particular context but also those services (from the pool of one’s all registered services) that have new content were also brought within the service dashboard. In this mode, the services having new or unread content were shown with some noticeable

size and graphical annotations, in addition to their spatial layout (as described in Section 3.3.3). The variants that we considered capturing through this usability test are following:

- *Variant-1: Overall user feedback regarding the framework*

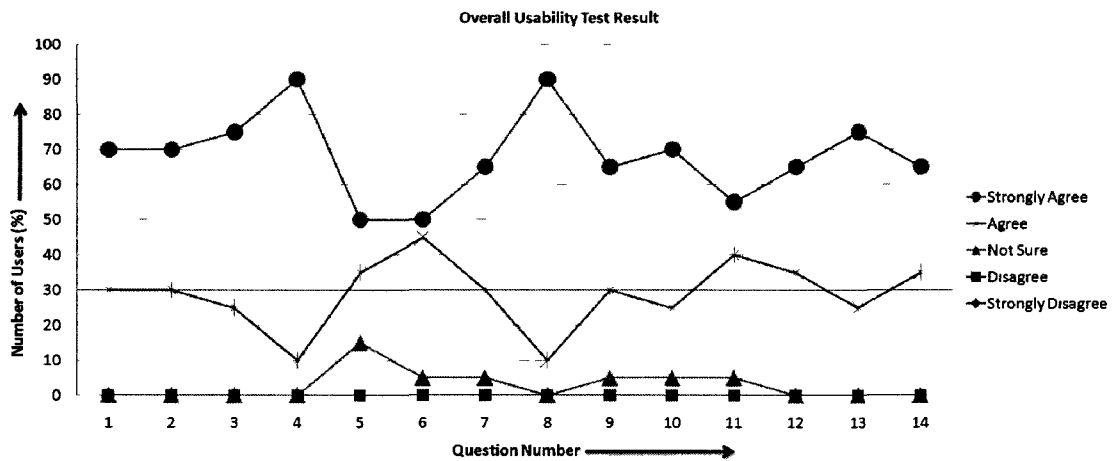


Figure 6.7 Integrated feedback results from all the test subjects regarding the framework.

Figure 6.7 shows the overall test results collected through the questionnaire initiative. This variant reflects the user feedbacks regarding the overall framework and the demonstrated features. The figure also portrays the integrated view of user satisfaction and experience. Question numbers four and eight scored the highest approval rating (90 per cent) under the category “Strongly Agree,” both of which showed the potential of the research value presented through the SenseFace framework. On the other hand, question numbers five and six of the same category earned the lowest score (50 per cent). In the case of question number five, we analyzed the subjects who did not approve strongly and found that they either belonged to non-technical user group or were those who only tested the framework once then provided

their feedback. In the case of question number six, user-tolerable delay is found to be vague and follows no pattern.

The mean value of subjective approval of “Agree” is approximately 30 per cent. The highest approval in the category “Not Sure” was given to question number five, which shows that users were skeptical about the idea of collecting user context from the analysis of content coming from heterogeneous services. One reason we unearthed was that many of the users do not use many Internet-based services on a regular basis, and so some could not pick up the concept of context extraction from service content. On the other hand, context extraction from sensors showed a better approval rating than its service counterpart, while both did the same job (providing context value to the framework). The average value of user ratings in “Not Sure” category is about 5 per cent. The most interesting information is that no user chose the categories “Disagree” and “Strongly Disagree.”

- *Variant-2: Overall user view about context-awareness*

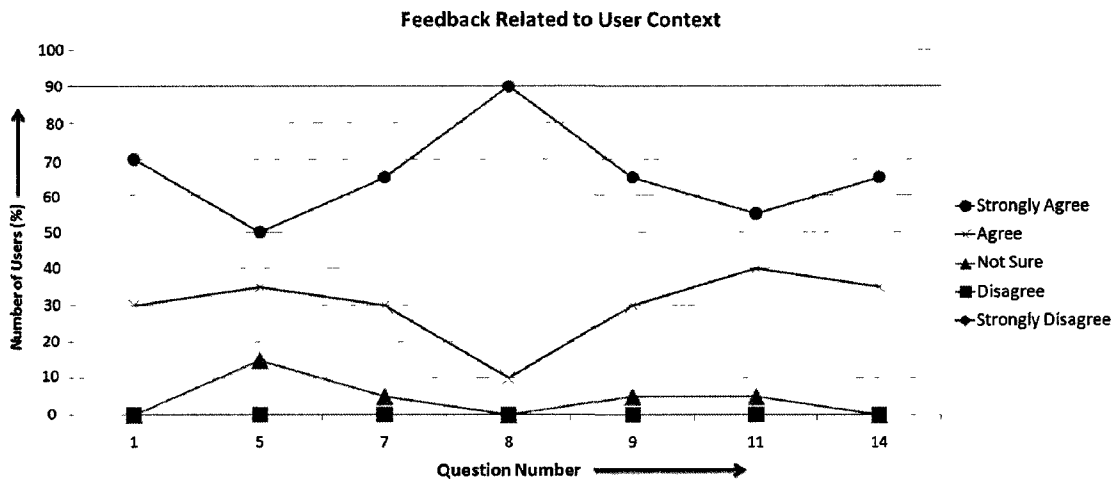


Figure 6.8 User feedbacks regarding global view of context-awareness.

Although different aspects of user context are evaluated through this subjective evaluation test, this variant provides the set of questionnaires that are fully or somewhat related to users' view of context. As shown in Figure 6.8, question numbers one, five, seven, eight, nine, eleven and fourteen were meant to collect context related user feedbacks. Around 70 per cent of users on average strongly accepted the notion about importance of user context, 25 per cent of users showed their positive view and the rest were undecided, leaving their answers empty for the category “Disagree” and “Strongly Disagree.”

- *Variant-3: Overall user view about the context-aware services*

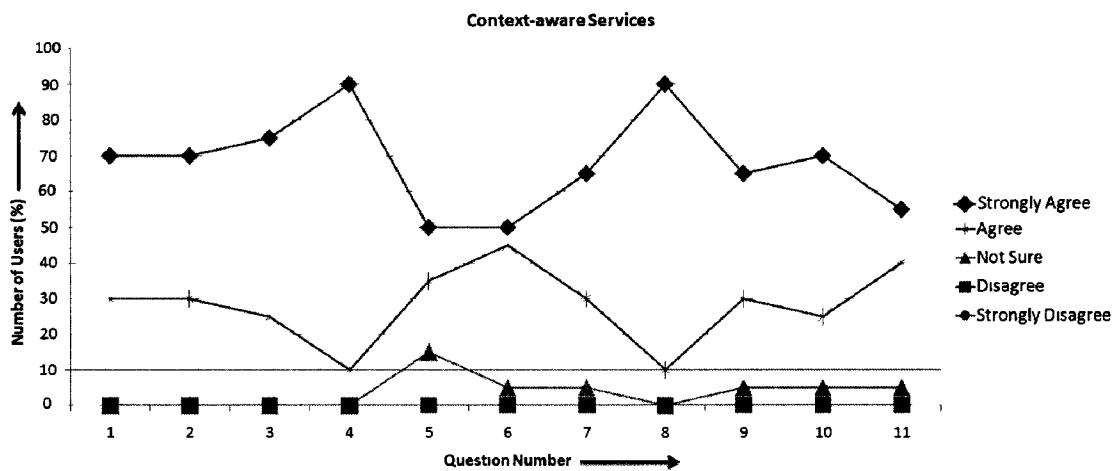


Figure 6.9 Users' view about providing services based on a given user context.

This variant helps us in collecting user feedback regarding the central idea and one of the most important contributions of this thesis — how interesting the idea of “context-aware services” is to users. Question numbers one to eleven either directly or indirectly brings forth user views regarding this important phenomenon. From the obtained test data, we observe that about 70 per cent of the users strongly agreed the concept of providing services based on user context while around 25 per cent of users

agreed upon the concept and the rest were unsure what to respond. No user completely disapproved of the idea.

- *Variant-4: User view about personal social networks*

The idea behind this variant spans to several questionnaires as shown Figure 6.10 (details of each can be seen from Table 6.2). Users were introduced to the term “personal social network (PSN)” to refer to all the personalized services and social ties that belong to a subject. Question numbers two, three, four, eight and ten through fourteen all share the concept of PSN, fully or partially. The mean value of the user choice regarding “Strongly Agree” is around 70 per cent with question numbers eleven, twelve and fourteen below this mean value. After analyzing these three questions, we concluded that all these questions refer to the features that were kept as “enhanced” options with respect to PSN. Hence, some liked the idea strongly while a mean value of approximately 25 per cent only liked the idea. The rest (5 per cent) could not decide.

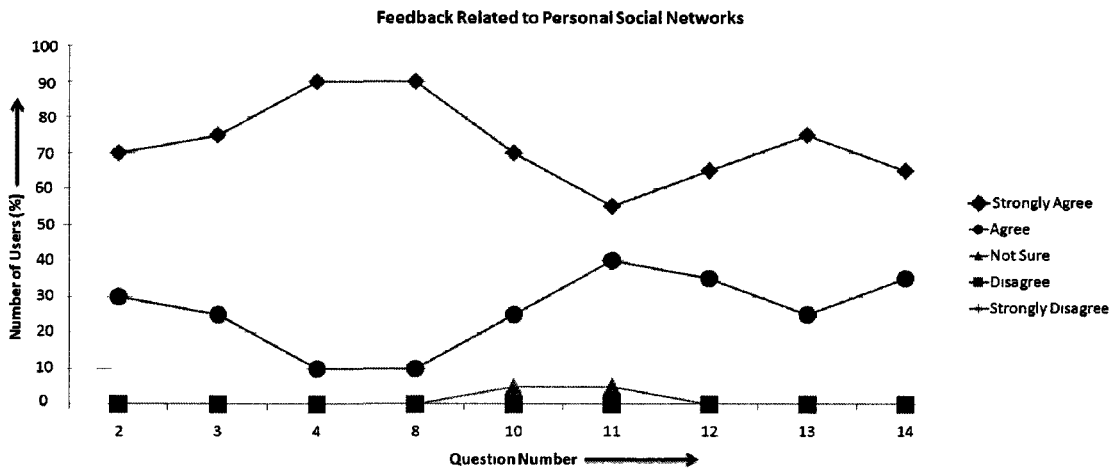


Figure 6.10 Users’ view regarding aggregation of one’s all the social ties and consumed services.

When we tried to analyze the questions that users answered “Not Sure,” it turned out that the scenarios may not be commonplace occurrences in the daily life of some of the subjects. Nonetheless, no person completely disagreed or strongly disagreed to any of the questions.

- *Variant-5: User view of community oriented services (Q13)*

The concept *Community Oriented Services* was introduced to the users in the context of collecting social ties’ information through those services. This is helpful in both aggregating one’s whole population of social ties as well as helping users in selecting COI from each of such services. As shown in Figure 6.11, most of the subjects (75 per cent) strongly liked the concept while the rest liked the idea. Same as before, most of the subjects were keen to see the idea operational with more services that were not shipped with the demo environment.

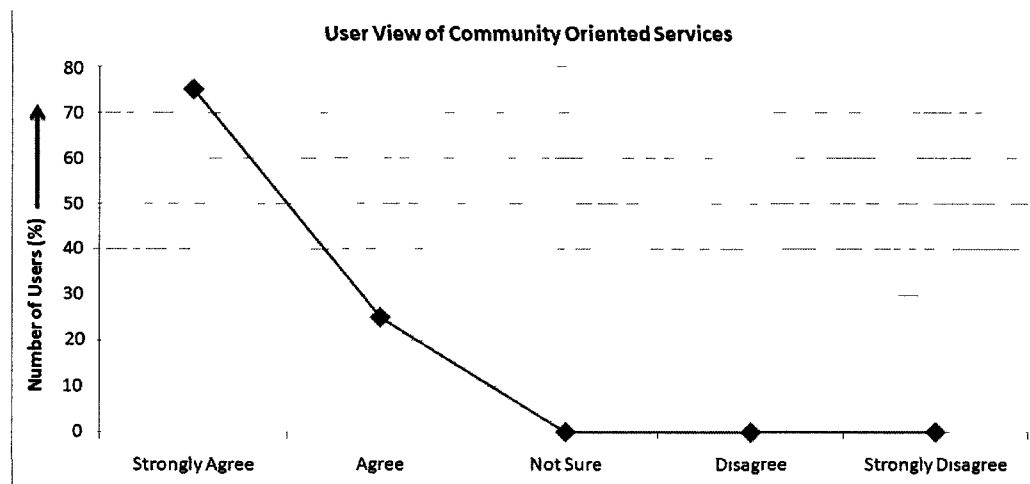


Figure 6.11 User feedbacks regarding community oriented services.

- *Variant-6: Collecting user context from sensors (Q9)*

The test environment was bundled with four sensor based services and users had the option to test a subset or all services. The four sample services were based on the 5 types of sensors shown in Figure 6.3. The first two services were to check user stress level using both heart rate and body temperature sensors. If a user was stressed for certain duration and outside weather was conducive to outdoor activities, then the framework suggested services that the user has configured for this context. Some popular services that people chose were location service, local bus schedule checker and local weather service. If the outside weather was too rough, then the users had the options to carry on indoor sensor-based games that emulate the outdoor activities. The rest two services are based on accelerometer and GPS sensors which tell the framework whether a user was doing some physical activities for a certain amount of duration (which a user can configure).

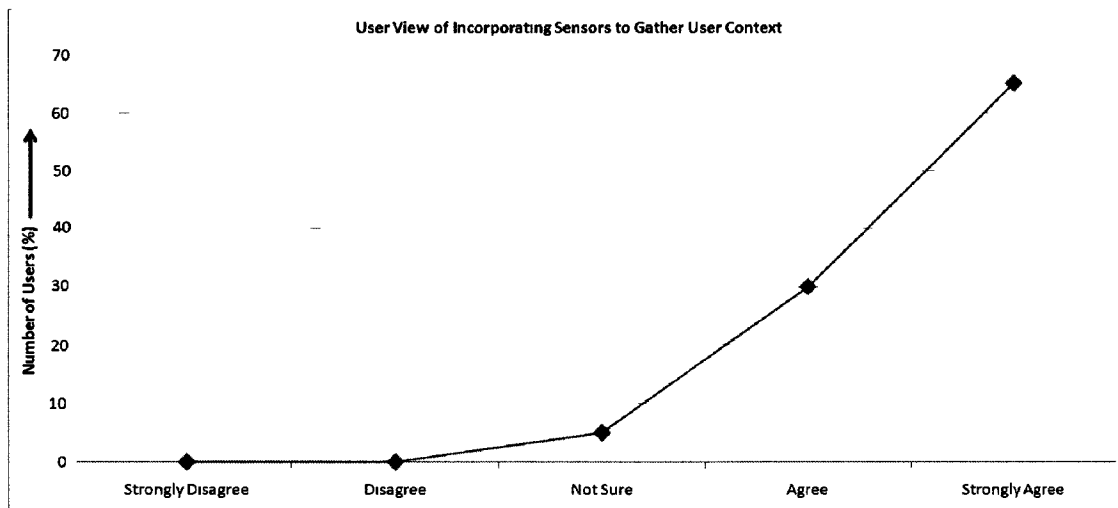


Figure 6.12 User view of collecting a user and his/her ambient information from body sensor network.

Figure 6.12 shows that 65 per cent of the subjects strongly supported the idea of incorporating sensors to monitor different vital parameters and advice them some suggested activities they prefer in those contexts. Thirty per cent of the subjects approved and 5 per cent did not have any clear opinion on incorporating sensors. Moreover, some users raised opinions regarding privacy issues while others were concerned about the prices of those sensors. Some showed doubt regarding carrying sensors constantly in order to access those services. The overall recommendations were very encouraging and optimistic.

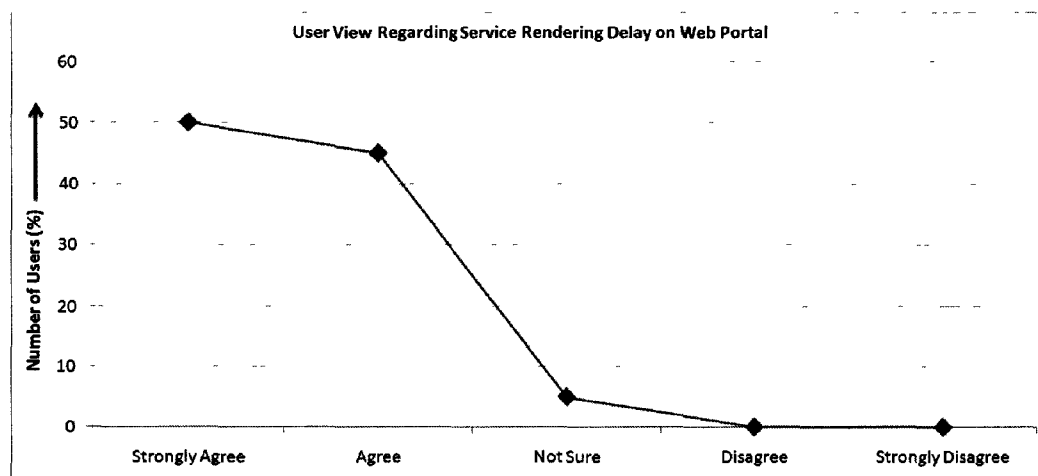


Figure 6.13 User feedbacks regarding waiting time tolerability in rendering services within the SenseFace portal.

- *Variant-7: User feedback regarding delay in rendering all the services within the portal (Q6)*

Each user was requested to comment on the waiting time starting from login to the full rendering of all the services that each chose for any given context. Although we previously presented such waiting time from an objective perspective (shown in Figure 6.4), this subjective questionnaire was intended to get the human version of a

tolerable delay. From the results shown in Figure 6.13, we can make several conclusions. First, half of the test subjects strongly agreed that the service rendering delay they observed during their testing period was satisfactory. Forty-five per cent said they were supportive with the delay while five per cent of subjects could not quantify the delay he/she observed. None showed frustration or disapproval about the delay value. However, the delay values that each user experienced might have been slightly longer if a large number of users (for example, 1,000) concurrently accessed the framework. Because of a lack of such numbers of volunteers, we already presented a software approach of artificially generating large number of users and benchmarked the delay and other similar metrics, which is already presented in Section 6.2.

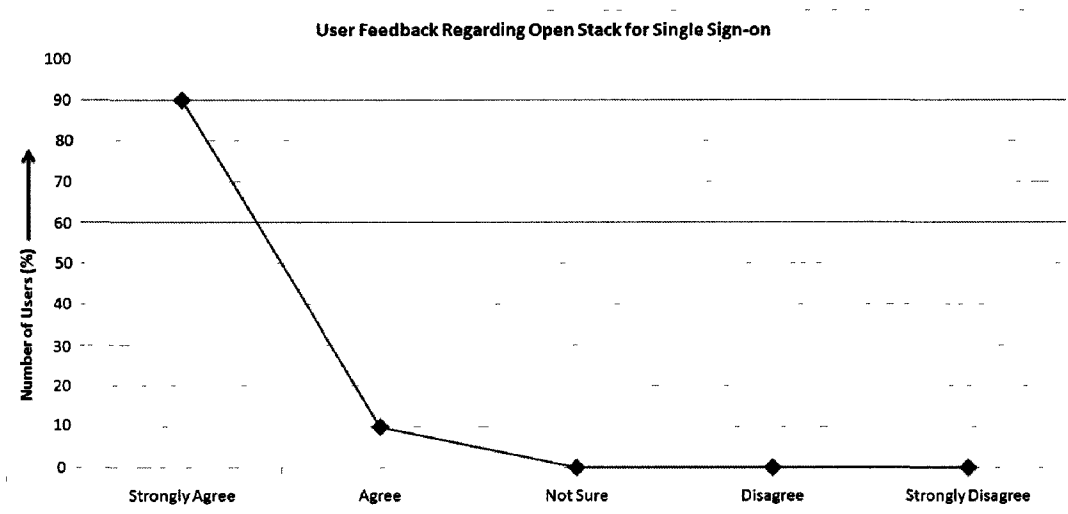


Figure 6.14 User view about benefit of single sign on for all the services.

- *Variant-8: Usefulness of single sign on using open stack (Q4)*

This variant points out two features of the framework. First, users voiced their opinion regarding memorizing the large number of login information to all the online services they use in their personal life. Thanks to the embracing of OpenID by a large number of existing Internet-based services, users liked the idea of logging into their

existing services using just one ID. Second, user opinions reflect the usefulness of open stack to seamlessly incorporating proprietary and legacy access control mechanisms through different layers of open stack. Ninety per cent of subjects showed strong support of this feature while the rest (ten per cent) were supportive (see Figure 6.14). Although most of the subjects just observed this feature by using the demo accounts, some users actually tested the feature with their own account information.

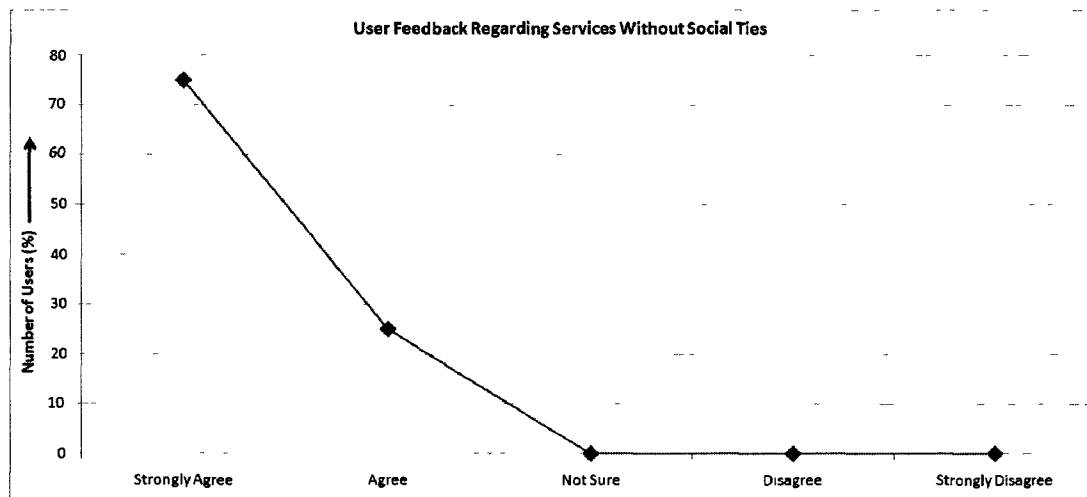


Figure 6.15 User view about extracting content from services without social tie categories.

- *Variant-9: User feedback regarding extracting content from services without social ties (Q3)*

Users were presented with several services as proof of concept to demonstrate the usefulness of extracting live content that might carry important context information such as weather data, location, addresses in maps, events from a calendar and sensory data, to name a few. As shown in Figure 6.15, seventy-five per cent of users showed strong support of this variant while twenty five per cent approved of the concept. Most of the users could distinctly identify these types of services from the one shown in

Figure 6.11 that these types of services are meant to consume information only, not to communicate with one's social ties.

- *Variant-10: Dynamic priority and layout management (Q7)*

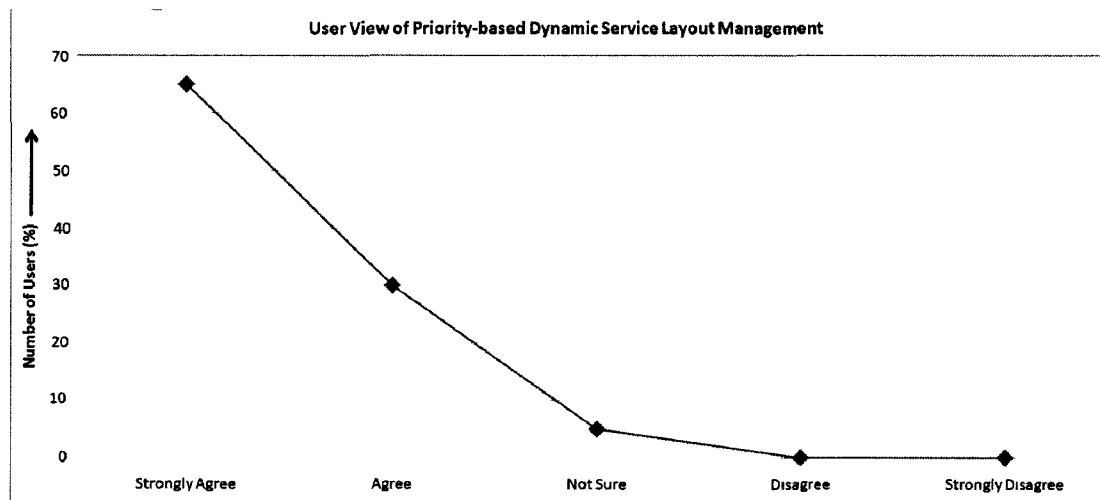


Figure 6.16 User responses on dynamic layout of the context-aware services and COI.

The idea behind this variant is to pull each service from the pool of context-aware services and tag each service with a use assigned priority value depending on the calculated context value that it contains at that given user context. These dynamically credited priorities are used by the portal to dynamically allocate the spatial layout. For example, a service having the highest priority will be allocated the middle portion of the portal grid as shown Figure 5.6. As discussed before, the demo can be experienced in two modes: simple and advanced. In simple mode this dynamic layout effect is not shown. Each user was requested to switch from simple to advanced mode to see the effect of dynamic tagging and layout of each service based on its earned priority value. The framework can be configured to augment each service with other visual and accessibility features such as adding emotional tags, varying size of each portlet, text

to speech and color variations. Sixty-five per cent of the users strongly accepted the idea, 30 per cent gave their consent and 5 per cent were unsure. Nonetheless, the results shown in Figure 6.16 are motivating.

- *Variant-11: Compound context generation from context primitives (Q11)*

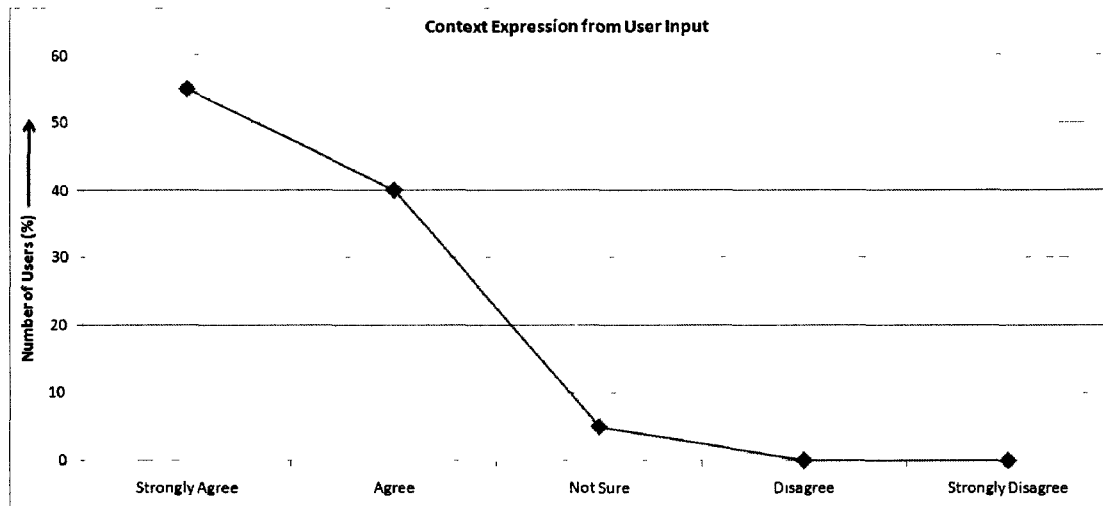


Figure 6.17 User comments about creating fine grained user context from context primitives.

Users were presented with interfaces such that they can make their context expressions in pursuant to the theories shown in Section 3.2. Initially, to assist mostly non-technical users, the framework showed some basic context primitives selected from the login information such as location of the user, semantic time (for example, time shown according to hourly time slice, portion of the day, day of the week and so on). The context primitives were categorized to reflect the context topology shown in Figure 3.3—for example, the context primitives were divided into four logical divisions: spatial, temporal, personal and task. A subject had the flexibility to add as many distinct context expressions into his/her profile. A user can visualize or edit the

context expressions that he/she has already defined in the past to facilitate the formation of new context expressions. For example, the framework provides the flexibility to express daily life and personalized context expressions such as the one shown in questionnaire number 11. The resulting user feedbacks related to this idea is shown in Figure 6.17. Fifty-five per cent of the users strongly accepted the proposition while 40 per cent showed their support and 5 per cent did not have either a positive or negative view.

The percentage of subjects who showed strong support for this variant is relatively less than other variants. We have pondered over this issue and found that the definition of context itself is vague and varies from subject to subject. Some subjects were somewhat confused about the way user interface was designed to facilitate the context expression. At the end of the test, many users gave their suggestions regarding this matter, which were very illuminative. We are constantly striving to accommodate these suggestions to improve the context expression capability.

- *Variant-12: User view regarding social tie visualization*

Users could observe the total number of their social ties sorted out by the types of services they communicate with as well as by some social categories they have defined such as family, relatives, friends, business, etc. The framework also presents a subset of social ties based on the observed context of a user. The context-aware social ties such as COI were also presented according to types of services that connect the subject with his/her social ties and the social category each social tie belongs to. Moreover, a subject had the option of sending a message using the particular service through which he/she is connected to the subject. For example, if a subject is connected with his family physician via Twitter health service, the user can send a short message to the family physician through the social tie visualization interface. Finally, a subject

had the option to see the online status of a particular social tie through the service the subject communicates with.

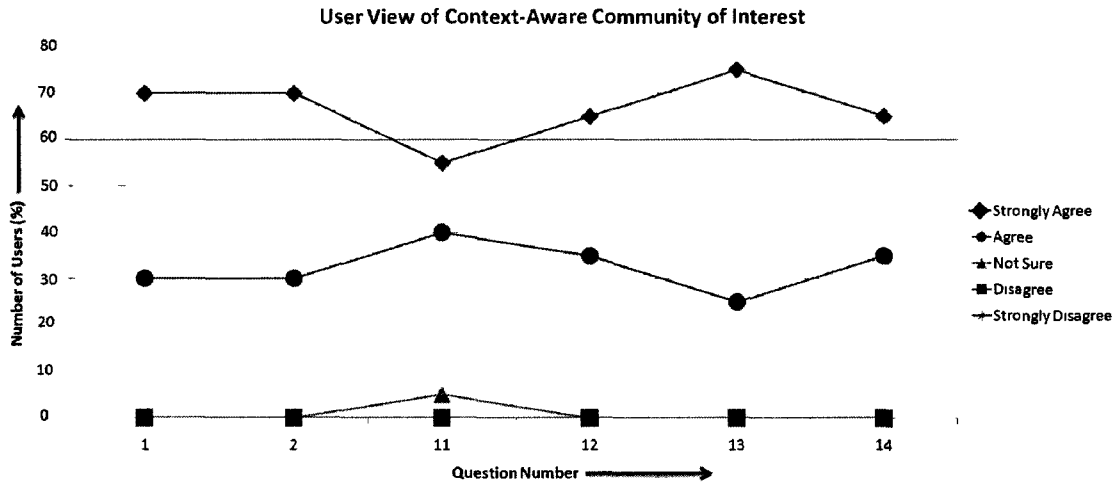


Figure 6.18 User view about presenting total and/or context-aware social ties.

For example, if a social tie of a subject is online in Skype instant messaging service, both the subject and the social tie can see their status, provided both of them set appropriate permissions. Figure 6.18 shows the outcome of the user opinions regarding this variant. Approximately 70 per cent of the subjects were very optimistic about the way SenseFace presented the social ties and the rest were positive about this idea. This shows the potential of incorporating social tie view within the framework. Many users expressed their enjoyment of seeing the whole population side by side with types of service and social categories that will enhance their communication capability and frequency with their social ties.

- *Variant-13: Non-technical users' view*

Figure 6.19 shows non-technical users' (homemaker, businessman, construction worker, etc.) views regarding all the aspects of the framework. This variant is intended

to capture how the people outside the university and research domain feel about the necessity of the concept proposed in this thesis. Non-technical users were first introduced to different key words and concepts through real life examples that were introduced through all 14 questions. Many users were not familiar or did not use some aspects of the framework in the past, which led to the “Not Sure” option. From the captured feedback, we were enlightened by their feedback and their urge in seeing the concepts serve their daily life. For example, services such as location, weather, monthly bill payment and local bus schedule based on user context such as location and time were very appealing to this group.

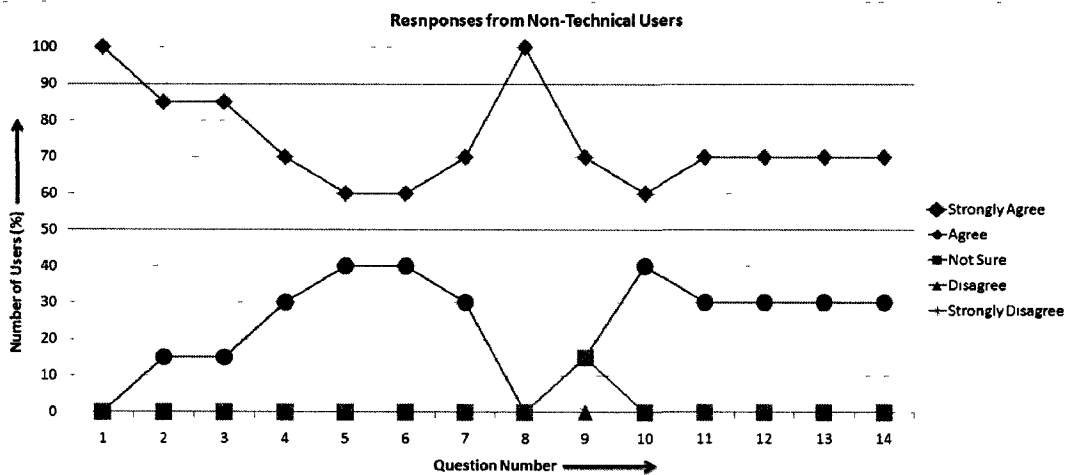


Figure 6.19 Non-technical users’ feedback about different aspect of the framework.

As seen in Figure 6.19, all the subjects in this category strongly supported the framework and its ability to check the content of each registered services (question numbers one and eight). On the average 75 per cent strongly supported the features provided by the framework while a mean value of 20 per cent showed their support. A mean value of about 5 per cent was undecided. We are in the process of doing a second round of usability tests in a large scale which will occur once we enhance both

interfaces and functionalities as suggested by the test subjects. The test results mentioned in this section only consider physical subjects, which might not portray the effect when a large number of users access all the features concurrently. The following section deals with the software emulation of large number of concurrent access to the framework.

6.4 SUMMARY

In this chapter we have tried to evaluate the framework from three different perspectives. In Section 6.1 we have presented run time performance of different subsystems. This includes the content extraction subsystem and dynamic adaptation of services and COI based on user context. We have elaborated the test results in terms of content extraction delay and missed attempts from both BSN and the Internet-based services. In Section 6.2 we showed how well the framework can scale by conducting a software-based benchmarking study where a large number of concurrent users (1,000) tried accessing the framework. The benchmarking test results are presented in three modes: first, without accessing any content from services, second, accessing only one service concurrently and third, randomly accessing one to 10 services by all the concurrent users. The first benchmarking test gives us a view how well the framework can handle concurrent user sessions, online profiles, and maintain separate states of PSN datasets. The second and the third benchmarking results actually show the real picture of the concurrency with taking into account the delay in collecting content from third party services.

Finally, Section 6.3 provides the details about subjective evaluation results. This was performed by conducting a usability test with real users. The users were asked to comment on different key concepts/variants that were introduced through this thesis. Detailed analyses emanate the fact that most of the users strongly supported the key concept areas.

CHAPTER 7: CONCLUSIONS AND FUTURE WORK

Due to the recent popularity of social networks in particular, heterogeneous Internet-based services in general and usage of body sensor network, we recognized the need for a context-aware social network framework. Existing literature reveals the potential of both sensory media as well as multimedia content carried by heterogeneous services to provide important contextual information of a user. To attain this, the following contributions to the field of Social Networks have been made as part of this thesis:

- A framework, called SenseFace, which bridges body sensor network and Internet-based services, including social network, to dynamically capture user context.
- Recommendation of context-aware services and social ties, called community of interest.
- Detailed modeling and design of the framework.

The proposed framework provides the following features. First, it is capable of extracting one's personal social network from the Internet by leveraging open stack of the Internet. Second, it can extract content from both body sensor network and diversified Internet-based services. Third, it can extract context information from the received sensory data and the multimedia content of heterogeneous services and create high level context primitives. Fourth, the framework can combine the generated context primitives to define a user context. Fifth, it employs a novel *Ubiquity Stack* that can map each user context with a subset of services and social ties. Sixth, the framework can dynamically assign a priority to each context-aware service so that the visualization interface can dynamically maintain the layout of each service based on

its earned credit. Finally, from the conducted performance evaluation, we conclude that the framework can scale well and support a large number of concurrent users.

We have described the mathematical background to model such context-aware framework. First, we elaborate how to express the personal social network of a user using standard mathematical set theory notations. Second, we model user context in two aspects. One is how to express raw context parameters and the other is how to combine them to produce complete context. Third, we elaborate the modeling of associating user context with one's personal social networks to map user context with a subset of services and social ties. Fourth, the modeling is wrapped up with the expression of dynamic service and social tie layout management system.

In an initiative to realize the framework modeling, we have proposed the design of the framework. The framework contains three high-level subsystems. The first subsystem is called PSN extraction subsystem which is responsible to extract personal social network of a subject from the Internet. The second subsystem is named *Context Extraction Subsystem* and employs six components to extract context information by analyzing available content from the BSN and services. The third subsystem is called dynamic service and COI adaptation subsystem, which is responsible to map user context with a subset of services and social ties that are already available in the personal social network database. We have identified the necessary protocols, system and communication architecture, salient components and layers of each subsystem. We have illustrated the implementation details of the framework. The BSN has been implemented using off the shelf sensors and open source smartphone operating system. The subsystems have been implemented using open source service modules available either as third party web services or developed as part of SenseFace framework initiative.

In summary, the design, modeling and implementation of the framework were meant to provide an insight of three important aspects that were targeted as the main

contributions of this research. First, we have shown how to extract user context dynamically by analyzing contents obtained from one's body sensor network and from heterogeneous Internet based services. Second, we have presented the algorithms to design and model context-aware services in a personalized fashion. Finally, we have illustrated how to model and design community of interest of a user at any given moment.

The framework has been evaluated through three modes of testing. During the first mode of testing, which was conducted by physical users, we have captured two performance metrics to understand the viability of the framework for subsequent usability tests: end-to-end delay of each subsystem, which gives us an idea about the overall run-time performance of the framework and number of missed attempts to access the raw input from both BSN and the services. During the second mode of testing, the framework was tested with large number of users having up to 1,000 users simultaneously accessing the framework. This shows the scalability of the framework. This mode of testing gives us a view of traffic and processing time as number of users increases. Finally, we have carried out a usability test by deploying the framework with some popular services to obtain feedback regarding different subjective measures.

The test results are encouraging and show that the observed metrics are within tolerable limits for such distributed and Internet-based best efforts systems [Gartrell, C. M. (2010)]. Moreover, user feedbacks show very promising prospect of the proposed framework. In the following, we present our future vision rooted around the kernel of this thesis that we have identified while conducting the usability and run time performance testing.

FUTURE PLAN

1. Preserving user privacy

One of the earnest concerns echoed from most of the users is to add proper privacy model so that user has the full control over how the framework uses the sensitive and personalized data originating from sensors and Internet-based services. However, this very important aspect was kept out of the scope of this thesis. However, we will be investigating existing and state of the art privacy models that would fit the SenseFace framework and will adopt a suitable privacy model.

2. Adding emotion and haptics to augment user context

We are contemplating adding new media with the SenseFace framework in addition to sensory and Internet-based multimedia. Two media candidates that we are planning to incorporate are haptics and emotion. For example, adding emotions can be translated into two phases. First, capturing a subject and his/her social ties' emotion occurs by analyzing states/contents provided by sensors and multimedia information carried by heterogeneous Internet-based services. Second, adding the emotional state with each service will enable users to semantically understand the importance of each service. Then, through haptic devices, we could feel the emotions tagged with each service. This would enhance the semantic visualization metaphors.

3. Measuring more metrics of context

In this thesis we have tried to answer the quality of context and dynamic priority allocations to the services from the usability test data obtained from human subjects. However, we want to propose suitable mathematical models to quantify the context values and the priority assignment scheme. This would help us in answering the two metrics quantitatively. We are investigating to measure the following metrics in near future: a) cost, b) response time, c) reliability, d) availability, e) reputation, f) accuracy of service prediction, g) user satisfaction in

terms of Mean absolute error (MAE), h) compound context error rate, i) quality of context, j) significance of context, k) confidence on context information, l) precision, m) context-weight gain, n) probability of correctness (PoC), threshold level for confidence on context information, and o) uncertainty of context information.

4. *Extending the application horizon*

At its current stage, SenseFace is designed to be a generic context-aware framework that can be extended for any particular domain. Because of its support of both sensors and Internet-based services, it can be adopted to a large number of applications such as e-Health, e-Learning, e-Business, serious games, smart homes, ambient intelligence and any application that needs to support user context. We will investigate the applicability of SenseFace framework to those domain specific applications in near future.

REFERENCES

- An, J., Pack, S., An, S., Kim, M., and Jeon, Y. (2009). "A novel service architecture for personalized context aware services," in Proc. 11th Int. Conf. on Advanced Communication Technology, 15-18 February, vol. 1, pp. 554 – 559.
- An, X., and Wang, J. (2006). "Opt: online person tracking system for context awareness in wireless personal network," in ACM REALMAN, Florence, Italy, pp. 47 – 54.
- Aziz, O. et al. (2007). "A pervasive body sensor network for measuring postoperative recovery at home," in ACM IPSN, Ser. Journal of Surgical Innovation, vol. 14, no. 2, pp. 83–90
- Bahri, A., Bouaziz, R., and Gargouri, F. (2009). "Fuzzy ontology implementation and query answering on databases," 28th North American Fuzzy Information Processing Society Annual Conference, June, pp.1-6.
- Balasundaram, B. (2008). "Cohesive subgroup model for graph-based text mining," IEEE Int. Conf. on Automation Science and Engineering, pp. 989 - 994.
- Baldauf, M., Dustdar, S., and Rosenberg, F. (2007). "A survey on context-aware systems," Int. Journal of Ad Hoc and Ubiquitous Computing, June, vol. 2, no. 4, pp. 263-277.
- Bardram, J. E. (2005). "Activity-based computing: support for mobility and collaboration in ubiquitous computing," Pers. and Ubiquitous Comp., Springer-Verlag, vol. 9, no. 5, pp. 312–322.
- BB Health Blog (2010). [Online]. Available: <http://www.bblfinal.com/make-your-doctors-life-easy-twitter-about-yourillness.html>.
- Beutel, J., Kasten, O., and Ringwald, M. (2003). "Btnodes a distributed platform for sensor nodes," in ACM SenSys. Los Angeles, California, USA: ACM, pp. 292 – 293.
- Berkman, L. (2000). "Social support, social networks, social cohesion and health," Social Work in Health Care, vol. 31, no. 2, pp. 3-14.
- Bhardwaj, S., Lee, D-S., Mukhopadhyay, S. C., and Chung, Wan-Y. (2008). "Ubiquitous healthcare data analysis and monitoring using multiple wireless sensors for elderly person," Sensors & Transducers Journal, vol. 90, pp. 87–99.
- Bolchini, C., Curino, C., Quintarelli, E., Schreiber, F. A., and Tanca, L. (2007). "A data-oriented survey of context models," SIGMOD Record, vol. 36, no. 4, pp. 19-26.

- Bottazzi, D. et al. (2003). "AGAPE: A location-aware group membership middleware for pervasive computing environments," Proc. 8th Int. Symp. Comp. and Commun., IEEE Press, pp. 1185–1192.
- Bottazzi, D., Corradi, A., and Montanari, R. (2006). "Context-aware middleware solutions for anytime and anywhere emergency assistance to elderly people," IEEE Communications Magazine, IEEE Press, pp. 82-90.
- Bottazzi, D., Montanari, R., and Toninelli, A. (2007). "Context-aware middleware for anytime, anywhere social networks," IEEE Intelligent Systems, Sept.-Oct, vol. 22, no. 5, pp. 23-32.
- Broens, T., Sinderen, M. van, Halteren, A. Van, and Quartel, D. (2007). "Dynamic context bindings in pervasive middleware," 5th Annual IEEE Int. Conf. on Pervasive Computing and Communications Workshops, pp. 443-448.
- Chao, W., Satuluri, V., and Parthasarathy, S. (2007). "Local probabilistic models for link prediction," 7th IEEE Int. Conf. on Data Mining, pp. 322 - 331.
- Christopoulou, E. (2008). "Context as a necessity in mobile applications," Handbook of Research on User Interface Design and Evaluation for Mobile Technology.
- Dey, A. K. (2001). "Understanding and using context," Personal Ubiquitous Comput., Springer-Verlag, vol. 5, no. 1, pp. 4-7.
- Eisenman, S. B. et al. (2007). "The bikenet mobile sensing system for cyclist experience mapping," in ACM SenSys. Sydney, Australia: ACM, pp. 87 - 101.
- El Helou, S., Salzman, C., Sire, S., and Gillet, D. (2009). "The 3A contextual ranking system: simultaneously recommending actors, assets, and group activities," Proc. of the 3rd ACM Conf. on Recommender Systems, New York, USA, Oct. 22-25, pp. 373-376.
- Fielding, R. T. (2000). Architectural styles and the design of network-based software Architectures. Ph.D. dissertation, University of California, Irvine.
- Ganti, R. K., Yu-En, T., and Abdelzaher, T. F. (2008). "SenseWorld: towards cyber-physical social networks," Int. Conf. on Information Processing in Sensor Networks, pp. 563-564.
- Gao, T., Greenspan, D., and Welsh, M. (2005). "Improving patient monitoring and tracking in emergency response," Proc. of the Int. Conf. on Information Communication Technologies in Health, July.
- Gaonkar, S., Li, J., Choudhury, R. R., Cox, L., and Schmidt, A. (2008). "Micro-Blog: sharing and querying content through mobile phones and social participation,"

- Proc. of the 6th Int. Conf. on Mobile systems, applications, and services, Breckenridge, CO, USA, pp. 174-186.
- Gartrell, C. M. (2010). SocialAware: context-aware multimedia presentation via mobile social networks. University of Colorado. Masters Thesis. [Online] Available:
http://www.cs.colorado.edu/~rhan/Papers/Mike_Gartrell_CU_MS_thesis-final.pdf.
- Gifford, D. K. et al. (1991). "Semantic file systems," Proc. of the 13th ACM Symp. on operating systems principles, ACM Press, pp 16–25.
- Gopal, B., and Manber, U. (1999). "Integrating content-based access mechanisms with hierarchical file systems," Proc. of the 3rd Symp. on operating systems design and implementation, USENIX Association, pp 265–278.
- Gu, T., Wang, X. H., Pung, H. K. and Zhang, D. Q. (2004). "An ontology-based context model in intelligent environments," Proc. of Communication Networks and Distributed Systems Modeling and Simulation Conference.
- Hamdan, S. (2009). Context aware layout manager for video conferencing systems. Master's thesis, Dept. Electrical Engineering, University of Ottawa, ON, Canada.
- Hanson, M. A. et al. (2009). "Body area sensor networks: challenges and opportunities," IEEE Computer, vol. 42, no. 1, pp. 58-65.
- Helou, S. El, Gillet, D., Salzmann, C., and Rekik, Y. (2009). "Software for sustaining interaction, collaboration and learning in communities of practice", Solutions and Innovations in Web-Based Technologies for Augmented Learning: Improved Platforms, Tools, and Applications, Advances in Web-based Learning (AWBL) Book Series. pp. 300-316.
- Huang, Q. et al. (2004). "Relying on safe distance to achieve strong partitionable group membership in Ad Hoc networks," IEEE Trans. Mobile Comp., vol. 3, no. 2, pp. 192–205.
- Huebscher, M. C., and McCann, J. A. (2006). "An adaptive middleware framework for context-aware applications," Personal and Ubiquitous Computing, vol. 10, no. 1, February, pp. 12 – 20.
- Hull, B. et al. (2006). "Cartel: a distributed mobile sensor computing system," in ACM SenSys, Boulder, Colorado, USA, pp. 125 - 138.
- Hung, K., and Zhang, Y. T. (2003). "Implementation of a WAP-based telemedicine system for patient monitoring," IEEE Trans. Inform. Technol. Biomed., June, vol. 7, no. 2, pp. 101-107.

- Joly, A., Maret, P., and Daigremont, J. (2009). "Context-awareness, the missing block of social networking," *Int. Journal of Computer Science and Applications*, Technomathematics Research Foundation, vol. 6, no. 2, pp. 50-65.
- Jovanov, E. et al. (2003). "Stress monitoring using a distributed wireless intelligent sensor system," *IEEE Eng. Med. Biol. Mag.*, vol. 22, no. 3, pp. 49-55.
- Kaenampornpan, M., and O'Neill, E. (2004). "An intergrated context model: bringing activity to context," *Proc. Workshop on Advanced Context Modeling, Reasoning and Management*, Nottingham, England, 7-10 September.
- Kalasapur S., Song, H., and Cheng, D. (2009). "Extracting co-locator context," 6th Annual Int. Mobile and Ubiquitous Systems: Networking & Services, pp. 1-7.
- Kapadia, A., Henderson, T., Fielding, J., and Kotz, D. (2007). "Virtual walls: protecting digital privacy in pervasive environments," *Proc. of the 5th Int. Conf. on Pervasive Computing*, LNCS, Springer-Verlag, vol. 4480, pp. 162-179.
- Kawahara, Y., Kurasawa, H., and Morikawa, H. (2007). "Recognizing user context using mobile handsets with acceleration sensors," *Proc. IEEE Int. Conf. on Portable Information Devices*, 25-29 May, pp. 1-5.
- Killworth, P., Eugene, J., Russell, B. H., Gene, S. A., and Christopher, M. (1990). "Estimating the size of personal networks," *Social Networks*, vol. 12, pp. 289-312.
- Kong, H., Xue, G., He, X., and Yao, S. (2009). "A proposal to handle inconsistent ontology with Fuzzy OWL," *WRI World Congress on Computer Science and Information Engineering*, vol. 1, pp. 599-603.
- Koolwaaij, J., Wibbels, M., Böhm, S. and Luther, M. (2009). "ContextKing: virtual kingdoms in real life," *Proc. of the 1st IEEE Int. Conf. in Games and Virtual Worlds*, March 23 - 24, Coventry, UK, pp. 61-67.
- Krco, S. (2004). "Health care sensor networks-architecture and protocols," *Ad Hoc and Sensor Wireless Networks*, August, vol. 1, pp. 1-25.
- Lam, T.H.W., and Liu, J.N.K. (2008). "Modeling a personal profile by Fuzzy ontology map and its applications," *IEEE World Congress on Computational Intelligence*, June, pp.414-421.
- Lee, C.-S., Wang, M.-H., Acampora, G., Loia, V., and Hsu, C.-Y. (2009). "Ontology based intelligent fuzzy agent for diabetes application," *IEEE Symp. on Intelligent Agents*, pp.16-22.
- Leimeister, J. M., Daum, M., and Krcmar, H. (2002). "Mobile communication and computing in healthcare - designing and implementing mobile virtual communities for cancer patients," *Proc. of the Tokyo Mobile Business Roundtable*, Tokyo.

- Leong, L. H., Kobayashi, S., Koshizuka, N., and Sakamura, K. (2005). "CASIS: a context-aware speech interface system," Proc. of the 10th Int. Conf. on Intelligent User Interfaces, pp. 231-238.
- Lorincz, K. et al. (2007). "Wearable wireless sensor network to assess clinical status in patients with neurological disorders," in ACM IPSN, Cambridge, Massachusetts, USA, pp. 563 – 564.
- Luo, J. et al. (2004). "Pilot: probabilistic lightweight group communication system for Ad Hoc networks," IEEE Trans. Mobile Comp., vol. 3, no. 2, pp. 164–179.
- Mika, P. (2007). "Ontologies are us: A unified model of social networks and semantics," Journal of Web Semantics, vol. 5, no.1, pp. 5-15.
- Mikalsen, M. et al. (2006). "Putting context in context: the role and design of context management in a mobility and adaptation enabling middleware," Proc. of the 7th Int. Conf. on Mobile Data Management, pp. 9-12.
- Munoz, M.A. et al. (2003). "Context-aware mobile communication in hospitals," IEEE Comp., vol. 36, no. 9, pp. 38–46.
- Nordström, E. et al. (2007). "Experiences from measuring human mobility using bluetooth inquiring devices," ACM MobiEval, San Juan, Puerto Rico, USA, pp. 15 – 20.
- Noury, N. et al. (2000). "Monitoring behavior in home using a smart fall sensor," IEEE-EMBS Special Topic Conf. on Microtechnologies in Medicine and Biology, October, pp. 607-610.
- Nurmi, P., Kukkonen, J., Lagerspetz, E., Suomela, J., and Floreen, P. (2010). "Betelgeuse a tool for bluetooth data gathering," Helsinki Institute for Information Technology (HIIT) [Online]. Available: <http://www.cs.helsinki.fi/group/acs/betelgeuse/articles/betelgeusebodynets07.pdf>.
- Ocampo, R., Galis, A., Todd, C., and De Meer, H. (2008). "Reasoner-Based flow context aggregation," 2nd Int. Conf. on Communications and Networking in China, 22-24 August, pp. 98 – 103.
- OpenStack (2010). [online]. <http://blog.broadbandmechanics.com/2009/05/02/open-stack-v2/>.
- Park, S. et al. (2002). "Design of a wearable sensor badge for smart kindergarten," Proc. of the 6th Int. Symp. on Wearable Computers, pp. 231 – 238.
- Perer, A., and Shneiderman, B. (2006). "Balancing systematic and flexible exploration of social networks," IEEE Transactions on Visualization and Computer Graphics, vol. 12, no. 5, pp. 693-700.

- Perer, A., and Shneiderman, B. (2008). "Integrating statistics and visualization: case studies of gaining clarity during exploratory data analysis," SIGCHI Conf. on Human Factors in Computing Systems, pp. 265–274.
- Phung, D., Adams, B., Tran, K., Venkatesh, S., and Kumar, M. (2009). "High accuracy context recovery using clustering mechanisms," Proc. IEEE Int. Conf. on Pervasive Computing and Communications, March, pp.1-9, 9-13.
- Preuveneers, D. et al. (2004). "Towards an extensible context ontology for ambient intelligence," Proc. 2nd European Symp. Ambient Intelligence, LNCS 3295, pp. 148-159.
- Quan, T. T., Hui, S. C., and Fong, A.C.M. (2006). "Automatic fuzzy ontology generation for semantic help-desk support," IEEE Transactions on Industrial Informatics, August, vol. 2, no. 3, pp. 155-164.
- Rahman, Md. A., El Saddik, A., and Gueaieb, W. (2009). "SenseFace: a sensor network overlay for social networks," Proc. of the IEEE International Instrumentation and Measurement Technology Conference (I2MTC 2009), Singapore, May 5-7, pp. 1031-1036.
- Rahman, Md. A., El Saddik, A., and Gueaieb, W. (2009). "Data visualization: from body sensor network to social networks," Proc. of the IEEE Int. Workshop on RObotic and Sensors Environments, Lecco, Italy, pp. 157-162.
- Rahman, Md. A., Alhamid, M. F., El Saddik, A., and Gueaieb, W. (2009). "An ambient intelligent body sensor network for e-Health applications," Int. workshop on Medical Measurements and Applications, Cetraro, Italy, pp. 22-25.
- Rahman, Md. A., El Saddik, A., and Gueaieb, W. (2010). "Building dynamic social network from sensory data feed," IEEE Transactions on Instrumentation and Measurement, vol. 59, no. 5, pp. 1327-1341.
- Rahman, Md. A., El Saddik, A., and Gueaieb, W. (2010). Augmenting context awareness by combining body sensor networks and social networks. IEEE Transactions on Instrumentation and Measurement, (Accepted).
- Rialle, V., Noury, N., and Herve, T. (2000). "An experimental health smart home and its distributed internet-based information and communication system: first steps of a research project", Medinfo, vol. 10, no. 2, pp. 1479-1483.
- Ryu, H. et al. (2007). "A task decomposition scheme for context aggregation in personal smart space," Software Technologies for Embedded and Ubiquitous Systems, Lecture Notes in Computer Science, Springer, pp. 20-29.

- Sanchez, E. (2006). *Fuzzy logic and the semantic web (capturing intelligence)*. Elsevier Science Inc., New York, NY, USA, ISBN-0444519483.
- Santana P.C. et al. (2005). "Supporting emotional ties among Mexican elders and their families living abroad," *Conf. Human Factors in Comp. Sys.*, ACM Press, pp. 2099–3003.
- Sarkar, P., and Moore, A. W. (2007). "Dynamic social network analysis using latent space models," *ACM SIGKDD Explorations Newsletter*, vol. 7, no. 2, pp. 31-40.
- Sarikaya, S., Alim, M. A., and Rezaei, S. (2006). "Integrating wireless EEGs into medical sensor networks," *ACM IWCMC*, Vancouver, British Columbia, Canada, pp. 1369 – 1373.
- Schmitt, L., Falck, T., Wartena, F., and Simons, D. (2007). "Novel ISO/IEEE 11073 standards for personal telehealth systems interoperability," *Joint Workshop on High Confidence Medical Devices, Software, and Systems and Medical Device Plug-and-Play Interoperability*, pp. 146 – 148.
- Shtykh, R.Y., and Qun, J. (2008). "Capturing user contexts: Dynamic profiling for information seeking tasks," *Proc. 3rd Int. Conf. on Systems and Networks Communications*, 26-31 October, pp. 365-370.
- Silva, B. De, Natarajan, A., and Motani, M. (2009). "Inter-user interference in body sensor networks: Preliminary investigation and an infrastructure-based solution," *6th Int. Workshop on Wearable and Implantable Body Sensor Networks*, pp. 35-40.
- Singh, L., Beard, M., Getoor, L., and Blake, M. B. (2007). "Visual mining of multi-modal social networks at different abstraction levels," *11th Int. Conf. Information Visualization (IV'07)*, pp. 672-679.
- Smith, H., Rogers, Y., and Brady, M. (2003). "Managing one's social network: Does age make a difference," *Proc. of INTERACT'03*, Kluwer Academic Publisher, pp. 551-558.
- Smith, K. P. and Christakis, N. A. (2008). "Social networks and health," *Annual Review of Sociology*, August, vol. 34, pp. 405-429.
- Smithson, M. J., and Verkuilen, J. (2006). "Fuzzy Set Theory: Applications in the Social Sciences," *Sage Publications*, Thousand Oaks, vol. 147.
- Snijders, T. A. B. (2009). "Longitudinal methods of network analysis," *Encyclopedia of Complexity and System Science* (Ed. Bob Meyers), Springer Verlag.
- Sriram, J. (2009). *Activity-aware electrocardiogram-based passive ongoing biometric verification*, Thesis (Masters), Dartmouth College, Hanover, New Hampshire. [Online] Available: <http://www.cs.dartmouth.edu/reports/TR2009-655.pdf>.

- Sun, F., Fok, C.-L., and Roman, G.-C. (2007). "Schat: A group communication service over wireless sensor networks," ACM IPSN, Cambridge, Massachusetts, pp. 543 – 544.
- Sun, S. (2009). "Stably extracting text contents from email messages with Python," 2nd Int. Conf. on the Applications of Digital Information and Web Technologies, pp. 199-203.
- Sunyaev, A., Kaletsch, A., and Krcmar, H. (2010). "Comparative evaluation of Google Health API vs. Microsoft Healthvault API," Proc. of the 3rd Int. Conf. on Health Informatics, January 20-23, Valencia, Spain, pp. 195-201.
- Talzi I. et al. (2007). "PermaSense: Investigating Permafrost with a WSN in the Swiss Alps," ACM EmNets, Cork, Ireland, pp. 8 – 12.
- Tapia, E. M. (2008). Using machine learning for real-time activity recognition and estimation of energy expenditure. Thesis (Ph. D.)--Massachusetts Institute of Technology, School of Architecture and Planning, Program in Media Arts and Sciences. [Online] Available: <http://dspace.mit.edu/handle/1721.1/44913>.
- Teixeira, T., Jung, D., Dublon, G., and Savvides, A. (2009). "Recognizing activities from context and arm pose using Finite State Machines," 3rd ACM/IEEE Int. Conf. on Distributed Smart Cameras, 30 August - 2 September, Como (Italy), pp. 1-8.
- Upendra, R., and Max, O. (2008). "Predicting network availability using user context," Proc. of the 5th Annual Int. Conf. on Mobile and Ubiquitous Systems, pp. 1-8.
- Velardi, P., Navigli, R., Cucchiarelli, A., and Antonio, F. D. (2008). "A new content-based model for social network analysis", IEEE Int. Conf. on Semantic Computing, pp. 18-25.
- Wang, X., Dong, J. S., Chin, C. Y., Hettiarachchi, S. R., and Zhang, D. (2004). "Semantic Space: An infrastructure for smart spaces," IEEE Pervasive Computing, vol. 3, no. 3, pp. 32-39.
- Wark, T. et al. (2007). "The design and evaluation of a mobile sensor/actuator network for autonomous animal control," ACM IPSN, Cambridge, Massachusetts, pp. 206 – 215.
- Wasserman, S., and Faust, K. (1994). "Social network analysis: methods and applications," Cambridge University Press, Cambridge.
- Wu, W. et al. (2008). "Medic: Medical embedded device for individualized care," Journal of Artificial Intelligence in Medicine, Elsevier, vol. 42, no. 2, pp. 137-152.

- Wyatt, D. et al. (2008). "Towards the Automated Social Analysis of Situated Speech Data," Proc. of 10th Int. Conf. UbiComp, Seoul, South Korea, pp. 168-171.
- Xu, X.-M., Zhan, J., and Zhu, H.-t. (2008). "Using social networks to organize researcher community. Intelligence and security informatics," Lecture Notes in Computer Science, Springer Berlin / Heidelberg, vol. 5075, pp. 421-427.
- Yager, R. R. (2008). "Granular computing for intelligent social network modeling and cooperative decisions," 4th IEEE Int. Conf. Intelligent Systems, vol. 1, pp. 1-7.
- Yang, B. H. and Rhee, S. (2000). "Development of the ring sensor for healthcare automation," Robotics and Autonomous System, vol. 30, pp. 273-281.
- Yao, J., and Warren, S. (2005). "Applying the ISO/IEEE 11073 standards to wearable home health monitoring systems," Journal of Clinical Monitoring and Computing, Springer, vol. 19, pp. 427 – 436.
- Zhdanova, A. V., Zoric, J., Marengo, M., Kranenburg, H. Van, Snoeck, N., and Sutterer, M. (2006). "Context acquisition, representation and employment in mobile service platforms," Proc. 15th IST Mobile & Wireless Communications Summit, 4-8 June 2006, Mykonos, Greece, pp. 64-68.
- Zhan, Y., Wang, S., Zhao, Z., Chen, C., and Ma, J. (2009). "A mobile device oriented framework for context information management," Proc. IEEE Youth Conf. on Information, Computing and Telecommunication, pp. 150 – 153.
- Zhang, J., Karim, M., Akula, K., and Ariga, R. (2008). "Design and development of a university-oriented personalizable Web 2.0 mashup portal," Proc. IEEE Int. Conf. on Web Services, 23-26 September, pp. 417-424.
- Zhang, P., Sadler, M. C., Lyon, A. S., and Martonosi, M. (2004). "Hardware design experiences in Zebranet," Proc. ACM 2nd Int. Conf. on Embedded networked sensor systems, Baltimore, Maryland, USA, pp. 227 – 238.