# RAIMA: a Framework for the Design and Analysis of Self-Adaptive Egocentric Social Networks 

Hossein Rahnama<br>Ryerson University

Follow this and additional works at: http:// digitalcommons.ryerson.ca/dissertations
Part of the Digital Communications and Networking Commons

## Recommended Citation

Rahnama, Hossein, "RAIMA: a Framework for the Design and Analysis of Self-Adaptive Egocentric Social Networks" (2010). Theses and dissertations. Paper 1584.

# RAIMA: A FRAMEWORK FOR THE DESIGN AND ANALYSIS OF SELF-ADAPTIVE EGOCENTRIC SOCIAL NETWORKS 

By Hossein Rahnama (B.Sc., M.A.Sc.)<br>Department of Electrical and Computer Engineering<br>Ryerson University

[^0]I hereby declare that I am the sole author of this thesis.
I authorize Ryerson University to lend this thesis to other institutions or individuals for the purpose of scholarly research.

## Hossein Rahnama

I further authorize Ryerson University to reproduce this thesis by photocopying or by other means, in total or in part, at the request of other institutions or individuals for the purpose of scholarly research.

## Hossein Rahnama

## ABSTRACT <br> RAIMA: A Framework for the Design and Analysis of Self-Adaptive Egocentric Social Networks, Doctor of Philosophy, 2010, Hossein Rahnama, Department of Electrical and Computer Engineering, Ryerson University

An emerging research area in pervasive computing is the inference of social context in order to facilitate and mediate communications among collocated people. Understanding users' needs through information reasoning and leveraging principles of social networks plays an important role in the emergence of innovative computer-mediated social networks. This thesis introduces a generic social networking framework for the design, analysis and visualization of opportunistic social networks. The proposed framework is capable of analyzing social similarities in order to provide decision support to users in the form of ego-centric social graphs. Using opportunistic data networks, a distributed inference model is introduced to provide multi-criteria attribute matching in an ad hoc computing environment. Enhancing communications protocols to deal with real-time analysis of dynamic data, generation of spontaneous semantics, and introducing efficient social visualization techniques are salient goals of this research. Efficient pattern matching algorithms in mobile ad hoc networks can have significant benefits in generating real-time context and eliminate the need for a centralized arbiter. In our research, we demonstrate a generic and customizable software architecture for achieving efficient pattern matching in mobile ad hoc networks. In this research we present a novel design for the development of a generic matching engine that is customizable to changing social scenarios. We show how customizable semantics can play an important role in decision-making, selection of a desired attribute, and notifying users with messages in a volatile mobile network.
Table of Contents
Chapter One: INTRODUCTION ..... 1
1.1 Motivation ..... 1
1.2 Objective ..... 5
1.3 Methodology ..... 8
1.4 Thesis Outline ..... 11
Chapter Two: Literature ..... 12
2.1 Background ..... 12
2.2 Social Networking Systems vs. Recommender Systems ..... 14
2.3 Literature ..... 16
2.3.1 Survey of Related Social Driven Systems ..... 16
2.3.2 Survey of Social Sciences Literature ..... 20
2.4 System Design Input from Social Sciences ..... 26
2.5 System Design Specifications ..... 33
Chapter Three: METHODOLOGY ..... 37
3.1 Introduction to the framework ..... 37
3.2 Recursive Multi-Criteria Matching Algorithm in RAIMA ..... 43
3.3 Generation of Social Context ..... 56
3.4 Pairing and Discovery Modules ..... 62
3.5 Visualization ..... 66
3.6 A RAIMA Case Study ..... 73
Chapter Four: RESULTS AND SIMULATIONS ..... 78
4.1 Simulation Environments ..... 79
4.1.1 Analysis of RAIMA with increasing profile size attributes ..... 80
4.1.2 Analysis of RAIMA with 50 users, Profile Size of 10 Attributes ..... 83
4.1.3 Analysis of RAIMA with 50 users and 50 Profile Attributes ..... 84
4.1.4 Third Case Study: A Small Social Network with big profile size ..... 85
4.2 Comparative Results ..... 85
4.2.1 Comparative Results Based on 10 Profile Attributes and 50 Users ..... 87
4.2.2 Comparative Results Based on 50 Profile Attributes and 50 Users ..... 88
4.2.3 Comparative Results Based on 100 Profile Attributes and 50 Users ..... 88
4.2.4 Comparative analysis in clustering a socio-centric social network using RAIMA ..... 89
Chapter Five: CONCLUSIONS ..... 102
5.1 Summary ..... 102
5.2 Contributions of the Research ..... 105
5.3 Suggestions for Future Research ..... 107
REFERENCES: ..... 109
APPENDIX A. RAIMA PROTOTYPE ..... 121

## List of Figures

Figure 1.1: Research Areas ..... 7
Figure 1.2: Research Methodology ..... 10
Figure 2.1: Process flow within an abstract Social System ..... 15
Figure 2.2: Use Case diagram representing the user in a social network ..... 26
Figure 2.3: Reliability factors in a social networking system ..... 33
Figure 3.1: Structure of a Stable Marriage Algorithm ..... 39
Figure 3.2: The RAIMA Framework ..... 42
Figure 3.3: Structure of RAIMA ..... 44
Figure 3.4: Nodes states in the RAIMA Framework ..... 54
Figure 3.5: Context Generation for distinct social scenarios ..... 57
Figure 3.6: A context template based on three social attributes ..... 58
Figure 3.7: RAIMA Deployment Model for three distinct social context ..... 59
Figure 3.8: User interface for the context generator ..... 60
Figure 3.9: User interface for a RAIMA user with 3 attributes ..... 62
Figure 3.10: Ad Hoc pairing using regular expressions ..... 64
Figure 3.11: Pairing Process in the Client/Server mode ..... 65
Figure 3.12: Egocentric vs. Socio-centric Social Networks ..... 66
Figure 3.13: A Cartesian Social Decision Space with 8 Context Templates ..... 68
Figure 3.14: Conceptual Distance between two users in one social context ..... 69
Figure 3.15: Conceptual Distance between two users in 3 context domains ..... 70
Figure 3.16: Conceptual Distance between two users in 3 context domains ..... 71
Figure 3.17: Pluggable ontologies used to create different semantics in one social structure. ..... 72
Figure 3.18: Generated Egocentric Graph for the flight scenario case study ..... 76
Figure 3.19: Conceptual distances between 8 users in the flight scenario case study ..... 77
Figure 4.1: RAIMA Control Panel ..... 79
Figure 4.2: Developed Architecture of the RAIMA Framework ..... 80
Figure 4.3: Structure of the RAIMA test ..... 83
Figure 4.4: Probability of Good Matches with $\mathrm{a}=10$ ..... 83
Figure 4.5: Probability of Good Matches with a = 50 ..... 84
Figure 4.6: Probability of Good Matches with $\mathrm{a}=100$ ..... 85
Figure 4.7: Comparisons between RAIMA and SMA based on 10 profile attributes ..... 87
Figure 4.8: Comparisons between RAIMA and SMA based on 50 profile attributes ..... 88
Figure 4.9: Comparisons between RAIMA and SMA based on 100 profile attributes ..... 89
Figure 4.10: Comparison between RAIMA and SMA in a socio-centric network ..... 90
Figure 4.11: 11 clusters based on using Context Template A with R from 0 to 1 ..... 92
Figure 4.12: 8 clusters based on Context Template B with R from 0 to 1 ..... 93
Figure 4.13: 8 clusters based on Context Template $C$ with $R$ from 0 to 1 ..... 94
Figure 4.14: Search space for Person 1 ..... 96
Figure 4.15: Search space for Person 4 ..... 97
Figure 4.16: Search space for Person 8 ..... 97
Figure 4.17: RAIMA Visualization of the sailing club simulation ..... 99
Figure 4.18: RAIMA Visualization of the academic conference simulation ..... 101
Figure 5.1: Interaction and Processes within the RAIMA framework ..... 104
Figure 5.2: Structure of RAIMA's modular design ..... 107
List of Tables
Table 3.1: User's Profile and Search Criteria ..... 45
Table 3.2: Nodal Categories ..... 48
Table 3.3: RAIMA Comparison Algorithms ..... 49
Table 3.4: Definitions ..... 50
Table 3.5: Recursive calculation of conceptual distance between two users ..... 52
Table 3.6: RAIMA Service matching with alternation ..... 55
Table 3.7: Multi-Criteria Social Matching with Three Attributes ..... 55
Table 3.8: Flight Scenario Data and Scores (best scores highlighted) ..... 74
Table 3.9: Time Range Distribution ..... 76
Table 3.10: Flight Scenario attributes scoring algorithms and sampled calculations ..... 77
Table 4.1: Test case 1 ..... 83
Table 4.2: Test case 2 ..... 84
Table 4.3: Test case 3 ..... 85
Table 4.4: Comparative test case 1 ..... 87
Table 4.5: Comparative test case 2 ..... 88
Table 4.6: Comparative test case 3 ..... 88
Table 4.7: Performance of RAIMA in a socio centric network ..... 89
Table 4.8: Search space and scores for person 1 ..... 95
Table 4.9: Search space and scores for person 4 ..... 96
Table 4.10: Search space and scores for Person 8 ..... 96
Table 4.11: Usage of RAIMA in a sailing club simulation ..... 98
Table 4.12: Usage of RAIMA in an academic conference ..... 100

## Chapter One: INTRODUCTION

### 1.1 Motivation

People look for other people for many reasons: dating, finding a shared interest, addressing economic issues, solving challenging problems, or just to have a conversation. Sometimes, people rely on particular people to help them in finding a person. The notion of an old lady who can find the right man for someone's daughter is familiar to many people. Computer-Mediated Social Networks may help in bringing people together in both physical and virtual spaces and raise new possibilities with respect to automating this process. Over the past ten years, commercial and research prototypes have explored the space of computer-mediated social networks extensively. Websites like match.com apply algorithms to users' profiles to introduce potential romantic partners. The LoveGety device in Japan comes in male and female versions and can be set in one of a few modes to introduce a match. The device broadcasts its user's mode so that when a male and female are within a five-meter reach of one another, both their devices beep and flash. Social networking tools like Facebook.com and MySpace.com let users create intricate friendship networks and use these networks for communications and meeting people. Meetup.com allows people with shared interests to meet in person in public areas. Researchers have also explored many other innovative applications such as matching people as they browse the web [1], introducing professionals based on skills [2] and matching people who are collocated in a geographical location [3]. Although there have been several studies by social psychologists on what attracts people to each other, there
has not been a well-defined study that incorporates these findings into the design principle of computer-mediated social networks. For example, in LoveGety, being of the opposite sex and being in proximity is considered sufficient to create a match. In research projects like the one proposed by Lowet et al. [4], browsing the same Web page may result in a match, as may [5] seeking the same question on the Internet. This suggests that in many social networks the idea of matchmaking has tried to mimic human intuition without relying on the definition of a "match" in social sciences literature.

In most systems, social semantics such as "matching rules" are defined during the design phase of the social system and not during the run time. This type of design is evident in Nitwho [6] and the work of Thorn-Santelli et al. [7]. This design model makes social systems less adaptive and adversely affects their scalability, especially when the social context changes from one domain to another. Social context can be defined as set of rules, semantics, and paradigms that represent a user and his or her preferences in a social system. Following this conventional design model, each social setting requires a separate system design requirement from the architecture level. Providing a simple but adaptive framework that allows users to define social semantics during the run time can significantly increase the practicality and intuitiveness of real-time social networking systems. Currently, a social networking Website designed for dating purposes is only useful for finding dates and not necessarily effective when social needs are different, such as when users want to find business customers or academic partners in their vicinity. To provide such features, system designers must provide run-time adaptation capabilities in social networking frameworks.

Current design models have proven to be less adaptive in mobile settings where users are constantly roaming and their needs change in real time. In current practices in the design of social networking applications, a separate design is required for each social scenario. However, if each social setting can provision its semantics to a generic framework that allows its users to apply roaming profiles in different settings without the need to redesign or recompile their software, the scalability and usability of mobile social networking systems can be enhanced. This can also result in a step forward towards the standardization of protocols and paradigms in the design of mobile real-time social networks. Run-time adaptation of social context is an important system feature and has been a research motivation in the design of our framework.

Another identified challenge in the current design practices of social networks is that many of these systems are designed based on the principles of recommender systems. These systems, such as Newsweeder [8], Conversation Map [9] and GroupLens [10], work like movie or book recommender systems except that instead of recommending items such as books or movies to their users, they recommend people to each other. We believe that the design of social networking systems should be different from recommender systems due to the sensitivity and nature of the data that they process. In social networking systems, some amount of personal information is usually revealed to obtain a match. This makes the interaction more risky and raises issues such as privacy, trust, personal attraction, and reputation to a greater importance. Revealing detailed
information about books or movies is not as sensitive a process as revealing demographic characteristics and background information about a user.

Due to the sensitivity of such data, system design guidelines for social networking systems can be described differently in comparison to recommender systems. This is one of the reasons we represent social similarity spatially and not textually. In our framework, we introduce a visualization engine that represents collocated users in the form of egocentric social graphs. A short distance between two nodes in the graph indicates a higher similarity degree between users. This model conveys the degree of similarity without revealing the exact profile information associated with a user. Protecting users' privacy in social networks is an important guideline in the design of our framework. After reviewing available literature in the design of social systems, we were inspired to create a paradigm that could introduce people to each other and indicate their similarities without revealing explicit private information during early encounters. As the trust levels between users increase, the levels of information release could be set and determined by the users themselves. This has been a challenge in system design as a compromise between ambiguous social systems such as SecondLife versus very open systems such as Google Orkut. In this research we try to introduce a framework that is less ambiguous but more inferable by its users. Our motivation is to introduce a development framework that makes social software more friendly, scalable, adaptive, and usable. One of our main inspirations in this work is to enable software developers to build social software that can enhance interpersonal communications, not replace it. We believe that the habits and behaviors of users should be minimally affected while using mobile devices as social
decision support tools and that these devices should be well integrated in the ergonomics of human interpersonal interactions.

### 1.2 Objective

The objective of this research is to introduce, design, and analyze a software development framework to build social decision support systems. The framework can have an impact in the emergence of adaptive social decision support systems, which are optimized for ubiquitous computing environments. The framework is proposed to address users' needs during social introduction and encounter, orientation, and social feedback. We have identified five key areas to be considered and analyzed in our research objectives. These objectives are targeted to address the topics and concepts discussed in section 1.1. Objectives of our research are addressed in the following research domains:

- Matching Algorithms: Matching algorithms in ubiquitous environments should be relative and multi-criteria. Implementation of Fuzzy Logic and Bayesian Belief Networks in ubiquitous computing environments is still not feasible. This is due to their excessive demands on computing resources that are currently unavailable on existing mobile computing platforms. The matching algorithm that we propose in our framework is linear yet multi-criteria.
- Run-time adaptation: The proposed framework is adaptive to different social environments, and the social semantics can be defined during the software run
time, not compile time. This means that each social setting can be treated and represented as a distinct social context in the form of a unique pluggable template/ontology that is utilizable by generic social networking frameworks.
- Service Discovery and Pairing: Selection and optimization of service discovery algorithms in ubiquitous networks has a direct impact on the longevity and performance of the network. Since structures of social networks can be complex, selection of discovery algorithms should be dynamic and based on the type, size, and structure of the social network. In chapter 2, we demonstrate our candidate discovery/pairing algorithm, which was used in the development of our framework for medium-sized spontaneous social network structures.
- Visualization: The framework provides a visualization model to represent and generate real-time social information in form of egocentric graphs. We have selected this model primarily due to the amount of time needed to process social data in textual formats. When social data is represented spatially, inference and understanding in social networks is significantly improved. This becomes more valuable in small user interfaces like mobile phones. Research works like SimBetAge [11] indicates how representing social data in egocentric form is easier to process in comparison to textual/sequential representation of social data.
- Feedback: Ad hoc and spontaneous networks disappear after interactions between the nodes are finished. However, since the proposed framework proposes matches, it is valuable that users are capable of storing encounters of interest and
providing feedback on the relevance of the proposed matches back to the framework. This feedback can be used to make the system more relevant and finely tuned for future encounters in each social context domain.

Research areas discussed in this thesis are summarized in Figure 1.1.


Figure 1.1: Research Areas

Research in the analysis of relational matching allows us to introduce an algorithm specifically designed for ad hoc social networks. The algorithm should be designed to handle profiles with multiple attributes and should handle them safely due to the private nature of social data. Analysis of different schemes for run-time adaptation of social semantics allows us to propose a model that makes the social networks scalable and adaptive. Since social networks are complex and very dynamic, different discovery algorithms may be utilized due to the nature of the social network. The visualization model allows us to make the social networks more usable on constrained devices such as mobile phones. Our work on feedback allows us to introduce a model in which users can
bookmark and record encounters of interest for future use despite the ad hoc nature of the social network.

### 1.3 Methodology

We have investigated the requirements for building a software development framework that is useful for processing real-time computer-mediated social networks. Since the framework requires various services and software components to work coherently, a middleware design approach has been adopted, following the guidelines of Nakajima et al. [12]. Our methodology is set to address the objectives presented in section 1.2. To address the requirements of suitable matching algorithms, we have performed a series of comparative analyses on current exact matching algorithms that are widely used in mobile social networks. We have identified their weaknesses in spontaneous social networks and have set the improvement of these weaknesses as requirements for our new proposed matching algorithms. We have demonstrated the structure of our algorithm and have provided comparisons on how our algorithm outperforms the principle of current matching algorithms with different numbers of profile attributes. We have provided grounds for how run-time adaptation can help the scalability of mobile social networks and have introduced an ontology template and a provisioning model that is capable of customizing the matching algorithm in different social settings. Based on the results provided in Chapter 2, we have indicated that in addition to the algorithm being multicriteria, the results from the algorithm should be presented in multi-dimensional context domains. This is presented in our proposed visualization model, which is representing social data in the form of egocentric social graphs. We have presented our visualization model based on the findings from social sciences and focus group studies.

There is research literature available on representing social data spatially in which social data is represented in form of socio-centric social graphs. These graphs mainly represent social ties using Friend of a Friend Ontology. We discuss why presenting social information in an egocentric format is preferable in mobile social networks. We have used our framework in three different social settings and present our results. Although the focus of our research has been on egocentric social networks, we have also provided results showing the benefit of the algorithm in socio-centric social networks. The methodology used in our research is summarized in Figure 1.2.

In summary, our framework has an advantage over existing systems because it provides a complete set of tools for building and maintaining adaptive mobile social networks by its users. Our motivation in introducing this framework is to address three important requirements in the design of social networking systems: 1) run-time adaptation of the system in accordance with changing social context, 2) a linear and multi-criteria matching algorithm that is optimized for ubiquitous environments in which techniques such as Fuzzy Logic are not feasible to use, and 3) real-time visualization of social context to enhance the intuitiveness and privacy of real-time social networks.

Requirements, Specifications and Literature Review

Analysis and Development of a Relational Matching Algorithm: RAIMA

Introduction and Design of a Discovery Algorithm suitable for Ad hoc Social Networks

Introduction and Design of a schema to make Social Networks Reconfigurable, Adaptive and Scalable

Introduction and Design of a Visualization Model suitable for ad hoc Social Networks

Development of a feedback loop suitable for ad hoc social networks

Delivery of the Middleware Framework

Development of trial applications on the provided middleware

Performance and Usability Analysis of the applications

Feedback from the focus groups

Figure 1.2: Research Methodology

### 1.4 Thesis Outline

In chapter 2, we provide a background on the emergence of computer-mediated social networks. We have provided our findings by reviewing literature in the social sciences and have indicated how social science principles can be incorporated into the design of social networking platforms. In chapter 3, we discuss available matching algorithms and their shortcomings and introduce our algorithm and its supporting development framework, called RAIMA. We also present our results for service discovery techniques and pairing models in ad hoc social networks.

Chapter 3 includes the design of our visualization engine, which is capable of representing users with different social profiles in a multi-context social domain. In Chapter 4, we provide results on the performance of the RAIMA algorithm and compare it with the widely used "Stable Marriage Algorithm". We also present three scenarios in this chapter to validate our comparative analysis. Chapter 5 includes our conclusions and insights into points of improvement for our work. Appendix A shows how RAIMA is used as a prototype and includes the user interfaces created for the prototype.

## Chapter Two: Literature

### 2.1 Background

Software suites installed on personal computers changed how documents are created beginning in the 1980s. The emergence of electronic mail changed how people communicate and how these documents are shared. The availability of the Internet and the World Wide Web to the public and its increasing ubiquity has had an impact on how people communicate with each other. The Internet is acting as a medium in the emergence of computer-mediated social networks. In these networks, people create relationships with each other in much the same way as they do in real-world models, in which communities and relationships follow sociological, psychological, and even political patterns. Can we assume that these computer-driven networks running on our personal information can impact our interpersonal communications?

People are fundamentally social beings. People look for other people for many reasons: dating, finding a shared interest, addressing economic issues, solving challenging problems, or maybe just to have a conversation. Sometimes, people rely on particular people to help them find a person. Teenagers have also been enthusiastic to use technology, from landlines to mobile phones, text messaging, and online chat-rooms [13] for social interaction. Computer-Mediated Social Networks may help bring people together in both physical and online spaces, and raises new possibilities in enhancing the process of social interaction.

Over the past ten years, commercial and research prototypes have explored the space of computer-mediated social networks. Websites like match.com apply algorithms to users' profiles to introduce potential romantic partners. The LoveGety device comes in male and female versions and can be set in one of a few modes. The device broadcasts its user's mode so when a male and female are within five-meter reach of one another, both their devices beep and flash. Social networking tools like Facebook.com and MySpace.com let users create intricate friendship networks and use these networks for communications and meeting people. Meetup.com allows people with shared interests to meet in person in public areas. Researchers have also explored many other innovative applications, such as matching people as they browse the Web [14], introducing an expert to an information seeker [15], and matching people who frequently visit a physical location [16].

Although there have been several studies by social psychologists on what attracts people to each other, there has not been a well-defined study that incorporates these findings into the design principle of computer-mediated social networks. For example, in LoveGety, being of the opposite sex and being in the proximity is considered sufficient to create a match. In [14] browsing the same Web page may result in a match and in [15], so may seeking the same question on the Internet. This suggests that in many computer-mediated social networks the process of matchmaking has tried to mimic human intuition without relying on social science results. We believe that social science results can provide a valuable foundation and a series of design guidelines for computer-mediated social networks. In this chapter, we explain how social networking systems are different from
other computer-driven systems such as the recommender systems. Then we propose a system design approach deduced from the social sciences literature.

### 2.2 Social Networking Systems vs. Recommender Systems

Simplifying the definition of social networking systems, one can say they are recommender systems that recommend people instead of goods and items. The principal design of a movie recommender system may not be very different from that of a book recommender system or a music recommender system. In this research, we plan to explain why we consider social networking systems as a system class of their own. We explain how social systems are different from recommender systems from the design point of view.

Recommender systems help users choose from large sets of items that are normally abundant and time-consuming to analyze. Recommender systems normally process users' preferences to create a smaller subset of items and then recommend this subset to the user. GroupLens [17] and MovieLens [18] are examples of collaborative recommender systems in which a filtering technique is used to match a target user with other users who have similar preferences, and then the systems recommend items that these neighboring users rated highly and that the target user has not rated. On the other hand, [19] and [20] are examples of content-based recommender systems. They typically apply models from machine learning to learn users' preferences and then use information retrieval techniques to select similar items to recommend. Social Networking Systems recommend people
instead of items. Recommending people is a salient specification that makes such systems different from other recommender systems. In these systems, a certain amount of personal information is revealed to obtain a match. This makes the interaction more risky and raises issues such as privacy, trust, personal attraction, and reputation to a greater importance.

Therefore, a basic social networking design model should address the privacy and validity of this personal information and introduce a data flow where such information would not be misused. To address these issues, we have divided the data flows in social networking systems into four arts: 1) Design, 2) Matching, 3) Recommendation, 4) Interaction, and 5) Feedback. Figure 2.1 shows this process model.


Figure 2.1: Process flow within an abstract Social System

Each step of this model raises different questions. Answers to these questions may lead to some useful design factors applicable in our proposed framework - for example, in the design phase, how the users should be profiled and what information a system represents about its users. In the matching process, how should the system compute the matching profiles? During the recommendation phase, how should the matching users be brought together? How much of the users' information should be revealed? In the feedback phase, how should the result of an interaction be represented to the users? Should the users provide explicit feedback about their encounters? If so, how is such feedback interpreted by the system and its users?

We analyze these questions by looking at the literature from social sciences and social system engineering. In section 2.3.1, we review current social networking systems from the system design perspective. In 2.3.2, we study the work of researchers on how they have used their findings from social sciences in designing social networking systems. Using these studies, we introduce a design guideline in section 2.4 that can address both technical and sociological specifications of a reliable opportunistic social networking system.

### 2.3 Literature

### 2.3.1 Survey of Related Social Driven Systems

Based on our review, we have categorized current social networking systems into five categories. Our classification is based on the different types of social networking systems
that we have reviewed in both commercial and research/prototype forms. These categories are as follows:
I) Social Systems for Information Seekers
II) Information Systems with Hidden Social Networking Paradigms
III) Opportunistic Social Networks
IV) Social Awareness Systems
V) Social Visualization Systems

## I) Social Systems for Information Seekers

Works such as ReferralWeb [15] and Expertise Recommender [21] are examples of this kind of social network. Both systems need two types of profiles, one about the user's expertise and the other about the user's social relationships. Both systems obtain the expertise profile by data mining. Referral Web mines Web documents for information about field experts. Then content analysis is used to associate topics to experts and build social relationships based on the cooccurrence of names on Web documents. While this information can be obtained from the Internet, Referral Web obtains it from document co-authorship, meaning that writing a paper together denotes a social relationship. Social Network information for Expertise Recommender is collected through a pile sort method and observation techniques. These systems introduce and connect people based on similar information interests.

## II) Information Systems with Hidden Social Networking Paradigms

The focus of these systems is on navigating information repositories to find facts. The repositories are constructed so that when users need information beyond that which is already recorded, pointers are provided to people who can help. PHOAKS [22] first lets users find Web pages on topics they are interested in, and then, if they are particularly interested, they can explore further to find and contact the person who recommended the Web page. The Designer Assistant [23] organizes software design knowledge as a hierarchical series of questions. Designers browse the hierarchy to get advice about their projects. Each piece of advice is tagged with an owner, which the users can contact if they cannot find an answer to their question. Answer Garden [24] is a similar research project built around users' questions and answers. The common characteristic of these systems is that there is no explicit profile of the users. Instead users are matched through a shared interest around a particular type of information.

## III) Opportunistic Social Networks

These systems match users opportunistically without a need for a specific user request. Results of the matching can be based on the users' shared interests, which are inferred by the system from users' current activity or records of past activity. I2I [14] is an example of opportunistic social networks. I2I matches users by applying text similarity metrics to cluster users who are browsing similar Web pages on the Internet. An interesting feature of I2I is that users can maintain their
anonymity while getting involved in peer-to-peer communications with other users over instant messaging and news groups.

Yenta [25] is a distributed agent-based system. In Yenta, each user's individual agent identifies interesting topics to the user by analyzing the user's profile. SocialNet [16] matches users based on recurring presence at certain physical locations. Each user's device stores identification of user's friends and notices recurring patterns of co-presence with unknown others. SocialNet also uses a recommendation model in which every time two friends meet, their devices exchange lists of unknown people who are in within their proximity. If friend X is a friend of friend Y's unknowns, then friend X will be prompted to introduce friend Y and the unknown.

## IV) Social Awareness Systems

[26], [27], and [28] are examples of socially aware systems. In these systems, users are allowed to maintain social awareness and to communicate with friends, family members, and colleagues. For example, instant messaging systems let users specify buddies, keep users aware of their status (e.g., available or idle), and let users engage in text chats with people on their buddy lists. Socially Aware Systems differ in two fundamental ways. 1) They can introduce people who don't already know each other but have shared interests and 2) They can suggest specific opportunities to collaborate even among people who already know each other.

## V) Social Visualization Systems

Reviewing these systems has provided us with some useful insights about how our system can visualize social information. These systems generally convey information that users can use to decide whether a community is one they would like to join and to identify which members they would like to communicate with. Research works such as the one by Donath et al. [29], [30] and Smith et al. [31] are examples of visualization systems that are used for representing online communities. These systems are used to mainly to identify communications patterns between people. In general, social visualizations support the goal of finding someone to communicate with, and social matching systems partially automate this process. Social visualization systems provide graphical representations of a social activity and leave the comprehension and decisionmaking to the user. Social matching engines, on the other hand, ease the decisionmaking process by identifying specific people a user may wish to communicate with and creating introductions to facilitate interactions.

### 2.3.2 Survey of Social Sciences Literature

Social science literature consists of research works focusing on the relationships between individuals and the individual in relation to a group. Researchers have studied topics including interpersonal attraction, friendship, dating, and mating. In this section, we emphasize works that offer information about how and why people come together, interact, and form relationships. Some literature concerning social attractions provided us
with insights about how user profiles should be designed. Some of the work focusing on social communities helped us in designing the social matching models.

We reviewed our social science literature in order to address requirements that a social networking system should have from a sociological point of view by seeking specific goals. We are interested to know how to model social interests, how to structure and analyze social communities, and how to understand the reasons users join a particular community. Understanding these principles has helped us to design a data structure suitable for representing users' profiles. This structure is proposed based on our findings from social sciences using the following five categories that are explained below:
A) Social Interest
B) Importance of the Social Context
C) Structure of the Social Community
D) Reasons for Joining a Community
E) Social Conditions for Co-operative Behavior

## A) Social Interest

In our review, we have identified three factors that can predict social attractions: i) user characteristics, ii) user demography, and iii) user familiarity with the community setting.

## i) User Characteristics

Examples of characteristics that make users interesting to each other are friendliness, trustworthiness, sense of humor, and physical attractiveness. Also, it is expected that individuals with similar personal characteristics are likely to be attracted to each other. This means that from the system design perspective, personal characteristics function as preferences that one individual may have about another. Although there are regularities within a culture, one person's profile attribute such as sense of humor could be interpreted as silliness to another user. Early works by Waller [32] investigated attraction in dating scenarios. He describes the complex cues used by humans to consider whether another individual is a suitable date or not. This can have a valuable input to our system design. It reinforces the point that user goals and attributes must be considered in the matching process and have to be weighted differently.

## ii) User Demography

Ethnic background, gender, marital status, profession, and annual salary are examples of demographic information. They are more objective than characteristic information-for example, either you are within an age group or not. Such features can be used as visible signals to introduce people. Works by [33] shows that demographic features correlate with interpersonal attractions. An experiment performed by Cosley et al. [34] shows that users search and discloses demographic information in a social network. It also shows that
users are sensitive to some of their demographic information being disclosed. Whyte [35] have presented results on the role of proximity in friendship development. He suggests that people who are physically proximate are likely to meet frequently and interact with each other more efficiently.

## iii) User Familiarity with the Community Setting

Whyte has shown that families who live near each other are likely to develop friendships. Kraut et al. [36] showed that proximity of offices strongly correlated with better collaborations. In [37], authors show that in online communities it takes longer for relationships to develop. In general, it is inferred from our studies that the more users are familiar with their interaction settings, the more the possibility of finding a good match may increase.

## B) Importance of the Social Context

An environment around a relationship sometimes influences the development of the relationship. Environments influence people's behavior, and behavior influences the assumptions people make about each other. In [38], Goffman shows how one role can be dominant in a particular setting, but people may flexibly adopt other roles at any time. For example, a professor may need to play the role of a father at home; in another case an office manager can play the role of a colleague in an informal setting. In [39], Jackson found that the context in which a friendship is formed influences the dimensions of similarity. In other words, people are attracted to different types of people in different settings. In [40], Gabarro shows that there is evidence that the more two persons are
similar in their backgrounds, the more easily a task-based relationship can be formed between them.

## C) Structure of the Social Community

An area of sociology called structural analysis has formalized the representation of social communities. People are represented as nodes, and connecting vertices show their relationships. Mathematical graph theory is used to represent and analyze concepts such as strength of social ties, central and peripheral social roles, information flow and access within a community. These are evident in [41], [42], and [43].

Geographic limitations, participation in online communities, kinship, and working relationships all have been used to define groups for social network studies. Barry Wellman [44] has analyzed computer-mediated communications to show that social ties over computer networks are similar to the ones established in face-to-face meetings. In [45], authors show how network analyses have been used to identify patterns of information seeking between normal users and very well-connected users (gatekeepers) within a social network.

## D) Reasons for Joining a Community

Research such as [46] and [47] shows that simply bringing people together in a group does not necessarily result in effective participation. An important objective within this research domain is to identify factors that motivate people to actively participate in a group or community. The work presented in [37] summarizes a model that can be used to
predict various motivational factors such as making group members care more about the group. In [48], Kraut et al. shows how such models are applicable to Computer Supported Cooperative Work (CSCW). He found that telling users something unique about their relationship to a topic motivated participation in a social network.

## E) Social Conditions for Co-operative Behavior

Kollock [49] explains findings and requirements for designing co-operative systems. He suggests that a good co-operative system should have the following features: 1) people must be able to meet again, 2) people should be able to identify each other, and 3) people should have access to information about how others have behaved in the past in order to be able to believe or trust them. An example of the last feature is eBay.com. eBay can be considered a visible reputation-driven system where users trust each other based on past experiences. Users of the system must be able to trust each other even if they have never traded with each other before. Resnick et al [50] suggests the structure of these recommender systems encourages good behavior.


Figure 2.2: Use Case diagram representing the user in a social network

The use case diagram presented in Figure 2.2 follows the UML notation and represents social factors affecting the user in social networking systems.

### 2.4 System Design Input from Social Sciences

Insights derived from our social science literature review in section 2.3 can be used as series of design guidelines in our proposed system. Following these, we plan to propose a social system based on the guidelines described in Section 2.3. We deduce a series of system specifications from these guidelines and define system requirements and specifications in section 2.4.

Personal information such as personal attributes, religious beliefs, and ethnic background are sensitive. People are concerned about how these types of information are handled by social software systems. In this thesis, we argue that personal information is in most cases necessary for a social networking system. There is evidence that users will be willing to provide such information if the social system is considered trusted, reliable, and effective. Based on the social science literature reviewed in section 1.3.2, we can conclude that users will be willing to provide personal data to the extent that they receive benefits from doing it. Work by [51] about disclosing personal data in e-commerce transactions gives more support to this point.

In [42], authors surveyed four hundred online users and their concern for online privacy. Authors identified three levels of concerns: 1) concerns from privacy fundamentalists ( $17 \%$ of sample size), who were generally very conservative in disclosing personal data, 2) concerns from a pragmatic majority ( $56 \%$ of sample size), who had very specific concerns and associated strategies for addressing their concerns, and 3) concerns from the marginally concerned ( $27 \%$ of sample size), who were willing to provide personal information in almost all circumstances. This study suggests that a reliable and trusted system may attract many users and would become acceptable to a large segment of users; Facebook.com is a good example of this observance. One question that we have addressed in our system design is defining that which specific personal data raise the largest user privacy concerns? For example a user's birth date has higher privacy significance than the user's hobby or education level. Another question is that how to define a suitable match depending on the context and users' goals? The design model of
our system should be explicit and should be based on empirical results from the social sciences where possible. This will make the system more understandable to the users and help them to understand in which situations the system is useful.

Another design factor derived from the literature [7] implies that people would like to meet others with whom they have been physically collocated. Research by Verbrugge [52] suggests that people who encounter each other in their daily rounds are likely to be quite similar. As an example, if you share a train with someone, there is a chance that you live in a similar neighborhood, have similar work and career types, and have comparable education and socioeconomic status. Concerns we need to address are how to design different models for different social settings and how these adaptive models can be matched to the task and be made available to users in an understandable and comprehensible form.

Visualization of a social network can also be an important system feature for the user, as it can increase the user's inference from a social setting. Research works in the structural analysis of social networks divide such visualizations into socio-centric (populationbased) and egocentric (user-centered). We need to address what model is more applicable in our system design. Criteria for selection are based on the user's feedback and how much information they infer from their social surroundings using these types of graphs. Socio-centric approaches represent all relations between individuals in a population such as members of an organization. Work by [52] demonstrates that socio-centric representations diverge from individual users' expectations. [43] suggests that users
prefer personal social network visualizations that show how similar they are to other users. From this we conclude that a visualization model targeted to a user should be egocentric than socio-centric. By an egocentric network, we mean that one that represents data about the relationships of a single person with respect to others. Contact Map [53] and [54] are examples of such systems. We plan to use egocentric networks to record user relationships and provide functions to browse the network and allow communications within the network. To achieve this, we could use either an explicit entry model that relies on users to provide information or approaches such as data mining to mine patterns of communications between users. Since we are proposing our system for dynamic social networks, we would like to rely on user entry instead of relationship mining. Using data mining in spontaneous networks is a difficult if not impossible task due to the lack of existing patterns and the volatile nature of the network and its nodes/users. Data mining is an interesting approach in organized and infrastructure-based social networks, but is beyond the scope of our research.

For making effective introductions, we need to design our system around an algorithm that can represent and introduce users to each other effectively. We plan to utilize egocentric graphs in building user relationships; therefore, we need to show how different tie strengths can be represented and shared among different users. We also need to address how users make inferences from social network visualizations. What types of social network visualizations are effective, and for what type of tasks?

Introducing users in a social network raises two challenges: the technical challenge of computing effective introduction information and the social challenge of maintaining sensitive personal user information. In order to create meaningful introductions, the system also needs to access and process context information from available raw data such as the location coordinates of a community. To address these challenges, we need to analyze what types of information are important for crafting an effective introduction between two people within a geographically bounded community. Can users judge or predict the consequences of the matching rules built into the system for introduction? That is, when the rules may impact users' privacy, how can users remain confident about the validity of those rules? We also need to address whether a system can use mutual acquaintances as an effective way of introducing users to each other.

The size of the social network is also an important concern. There are two points where the number of users in a social network is of concern: first, the total users of a system (potential matches), and second, the number of users brought together in a match. Typically, more total users are better, since this makes it more likely that good matches can be found for any given user in the network. Research works by Ackerman [24] show that the efficiency of recommender systems is better when the scope of the match is limited appropriately. Co-locating users based on similar interest can act as a filter to bring similar people together. Another design factor is that users do not want to be matched with lots of others. This may indicate that although a large number of potential matches can increase the probability of finding a match, limiting the scope of a social network can have an equally beneficial effect $n$ the efficiency of the system. Other issues
regarding the size of a social network include identifying in what cases the social networks require large user populations to be effective and when can they work effectively within a limited population. What factors such as user goals, prior relationships, and overlapping interests can influence the effective size for a social network?

Understanding the context of interaction is also important for creating an effective social networking system. This is due to the fact that user goals for a social interaction can differ. These types of interaction differ from settings to setting and from online virtual spaces to real settings. Since users may share personal information while roaming between social settings, an important factor to consider is that a system should not reveal any personal information about a person without that person's consent in each setting. This means that privacy policies should be defined separately for each social network.

User feedback to a social networking system can help the system to introduce better matches the next time users use the system. However, obtaining such feedback is harder than getting user ratings for books or music. The dataflow between expressing preferences, building profiles, computing recommendations, receiving recommendations, and rating more items is essential to the success of recommender systems. Social matching systems may benefit from using a similar feedback loop. After users are introduced by the system, they can indicate their satisfaction with the match so that the system can update user profiles and social models. However, the contextual nature of human activity raises some complications. There are different ways that a person can be
judged. A single rating of a person would not be sufficient in a social networking system. A person can be better or worse, more or less trustworthy in different ways and in different settings. For example, when looking for a partner to play basketball with, factors such as skill and agility are considered. The same person may be able to find better matches if he uses different set of attributes in a conference settings.

Creating a model that allows users to change their set of profile attributes in different contexts is an interesting and challenging design problem. The evaluation of a person from one context to another dynamically changes, and ratings and reputations can roam between different social settings. This raises the issue that users should be ranked only within a particular context, and the score they obtain should not influence their scores in other settings.

One design factor to keep in mind is whether reputations should be portable; when obtaining a feedback about a match, the system should know which aspects of the match deserve credit or blame. What should be the system's response to a user's feedback? For example, should profile attributes be updated or should a different social model be used? In this section, we presented our findings from the reviewed literature in distinct categories. In summary, we presented that privacy, context of interaction, size of the social network, representation of the social data, and user feedback play important roles in the successful design of a reliable social network. These factors are depicted in Figure 2.3.


Figure 2.3: Reliability factors in a social networking system

### 2.5 System Design Specifications

We plan to set our system specifications based on the input from the system design literature, described in section 1.3, and the social sciences literature, presented in section 1.4. We set specifications for a useful, reliable, and trustworthy social networking system and realize the system design architecture based on these specifications. We introduce a middleware framework with which programmers can develop socially driven software applications. The framework provides software utility tools that programmers can use to customize their applications for a particular design scenario. Our framework provides these design patterns in the form of software components, and programmers can use them for building effective social networking systems.

We have summarized the specifications of our framework into five distinct category groups and seventeen specifications in total based on the reliability factors presented in section 1.4. The specifications and groups are described as follows:

## - System Specifications Group I: Privacy

1 In many cases, social software need access to sensitive personal information. If we build a trusted system, users may be willing to share the information with the proposed system.

2 The system should be able to create effective introductions between users while maintaining the privacy of personal data.

3 Interacting physically offers greater rewards and risks than interacting in a virtual space; when this is an option, the system must support users in exercising this option safely.

4 Levels of privacy shall be defined so that users can exchange valid profile information when a particular trust level is reached. For instance, this means that pseudo-names can be transformed to real identities in an online community when a certain level of trust is reached between two users.

## - System Specifications Group II: Structure of the Social Network

5 Users of the system should have a clear idea of the models that the software uses to match them with other users.

6 Demographic information must be kept protected in the matching process and becomes available only upon users' explicit request. However, this information can be used securely to compute the possibility of a match within a community.

7 The system should be designed in a way to encourage collaboration between its users.

8 The system should distinguish users' connectedness. Very well connected users can be detected within a community and may be used as gatekeepers within a community. This may have a significant effect on the efficiency of searching and recommendations features within a social group.

9 The system should be able to utilize social structures and connections between the users.

## - System Specifications Group III: Size of the Social Network

10 The system's efficiency should not rely on the size of the social network.
11 Users' location can be used as a filter to increase the possibility of a match. For instance, users gathered in a church may have higher possibility of a match in comparison to people walking randomly in a street.

## - System Specifications Group IV: Context of Social Interaction

12 The system should be capable of providing egocentric and socio-centric information to the user depending on the context.

13 User goals must be considered in the social system in the form of context information.

14 User profiles should be dynamic and adaptive in different settings for optimized social experience and increasing the possibility of a match.

## - System Specifications Group V: Feedback Loops in Social Networks

15 The system should be reputation-driven to increase the trust level of its users.
16 The system should be able to leverage users' feedback to introduce better recommendations as the user uses the system. The feedback should also be relative to a specific role or context.

17 The system should be designed in such a way that users know about a possibility that they may encounter or meet again. According to the literature, this encourages good behavior between users in a virtual community.

We have addressed these system specifications in our design guidelines and propose our framework in Chapter 3. We introduce a multi-criteria matching algorithm to address specification groups I and II. We introduce a discovery model integrated with RAIMA to connect users to each other in an optimized way; this addresses specification groups I, II, and III. RAIMA and the discovery and pairing services are described in detail in Chapter 3. In Chapter 3 we also introduce a visualization engine to address specification group IV. We show how we have used a pluggable ontology model to address users' contexts of interaction and target specification groups IV and V. In chapter 4, we demonstrate our results and provide some analytics about the performance of two sample applications built on top of our framework. In chapter 5, we provide our conclusions and prospects for future work.

## Chapter Three: METHODOLOGY

### 3.1 Introduction to the framework

In this chapter, we propose a framework that includes a multi-criteria matching algorithm called RAIMA which is capable of providing optimal matches to its users. To increase the efficiency of our proposed algorithm in spontaneous social networks, we also introduce an alternative service discovery mechanism that can distribute profiles among social nodes more efficiently. We demonstrate how our algorithm can change from one social context to another. We achieve this goal by separating the social context from the algorithm itself. This allows social context providers such as venue organizers to use simple tools to customize the algorithm for their needs. We then represent our visualization engine, which can provide the algorithm's space graphically as an enhanced usability factor to our system.

Efficient pattern matching algorithms in mobile networks can have considerable benefits in generating supportive information and reduce the significance of a centralized communications arbiter, used in many mobile social networking systems. Also, social science results provided in chapter two can present valuable foundations and design guidelines for the design of a computer-mediated social system.

Based on our findings in chapter 2, we have categorized these factors in a social networking system as follows:

- Importance of the social context
- Importance of the weight of demographic attributes in different social interactions
- Selection process in a social structure is multi-criteria and relatively optimal, not ideal

Importance of the social context is presented in social science literature such as [55], [56] and [57]. These works signify the importance of the interaction context. They demonstrate that an environment around a relationship is conducive to the development of a good relationship. Environments influence people's behavior, and behavior influences people's attributions towards each other. In [58], Reinhard shows that there is evidence that the more two people are similar in their backgrounds, the more easily a task-based relationship can be formed between them. In system design, this indicates that the algorithm used for social matching should be adaptive to the social context, which implies that the social context should be defined on a layer separate from that of the algorithm itself. Therefore a model should be provided so that the context can influence how the algorithm executes.

The second design guideline is about the importance of attribute weights in social networking systems. In different social contexts, different attributes should have different weights. For example, at a research conference, the research interest and university affiliation can carry greater weight in comparison to the gender or age of the attendees.

On the other hand, if the context of interaction is set around a dating scenario, gender and age may be assigned greater weight compared to research interests. This indicates that each context imposes different weights on the context's associated attributes.

The third design guideline for the algorithm is its ability to perform a multi-criteria selection process. It is unlikely that the matching in a social network can perform well based on exact matches. It is evident that human selection and preference models are usually relative. We all have preferences, and our preferences also have alternatives. Despite this, many of the social networking systems that we surveyed such as [59]-[61] are basing their design on exact matching and "Stable Marriage" matching algorithms. These algorithms do not consider the user's alternative selection. A pseudo example of the Stable Marriage algorithm is shown in Figure 3.1. As can be seen in the Figure, Stable Matching algorithms are not capable of performing matching based on multiple criteria.

```
Method Aggressive MatchMaking
while there is an unpaired male do
pick unpaired man M and the first woman
f on his list
remove f from his list so it won't be
picked again
if f}\mathrm{ is paired then
if f prefers M more than her current
partner B then
set M-f as married
set B-f as unmarried so now B is
unpaired
else
M is still unpaired since f is happier
with B
end if
else
woman f is not paired
immediately M-f, as married
end if
end while
```

Figure 3.1: Structure of a Stable Marriage Algorithm

Using stable or exact matching algorithms generally imposes three restrictions on a social networking system:

1) Users need to provide many attributes to be able to find an optimum match.
2) Such systems perform poorly with a small number of users and do not give any supporting information about the social structure and the proposed matches.
3) The waiting time for each query to be matched is relatively high and is proportional to the number of active users within a system.

In this chapter, we present a framework that can provide similarity scores (Conceptual Distance) between collocated users, allowing each user to analyze a social network effectively based on their needs and context. Later, we show how we use these scores as the basis of our social visualization engine. A good network visualization engine can give important information to the user about his or her social surroundings. Another feature of our system is that it represents similarity or differences between users spatially rather than textually. We believe that users can infer more about their social surroundings using spatial visualization compared to textual models, mainly due to the time that is required to process textual information. Also, since privacy is a significant concern in social networks, abstracting user information in the form of a spatial distance and not revealing the string-based data can be considered an innovative feature of our framework. Although visualization of social networks has been a topic of study during the past decade, no literature has presented user similarity by integrating an algorithm within the visualization engine. Vizster [62] and Cavalier [63] present interesting works and focus on a socio-centric model in which the whole community is presented to the user and the
ontology used in both systems are mainly based on a Friend of a Friend (FoAF) relationship and not the similarity between the users. Reviewing these systems has provided us with useful insights about how a social system can visualize societal information using a multi-criteria matching algorithm. Social visualization systems generally convey information that users can exploit to decide whether a community is the one they would like to join and to identify members they would like to communicate with. NodeTrix [64] and works by Sack [65] and Smith et al. [66] are examples of visualization systems that are used for representing online communities. Social visualization systems provide graphical representations of a social activity and leave the comprehension and decision-making to the user. Social matching systems, on the other hand, ease the decision-making by identifying specific people whom a user may wish to communicate with and creating introductions to facilitate interactions.

Our research goal is to integrate these two systems into a single framework and focus on real-time social networks where users are usually co-located physically and can use this framework to obtain more information about their social surroundings. Another important feature of our proposed framework is its ability to provide supporting societal information to proximate users while maintaining their privacy.

The framework has a multi-layer and modular architecture to increase its customization and scalability. The structure of the framework and its relationship to other layers is shown in Figure 3.2. The framework uses the core functionality of operating systems for resource management and communications. Sets of utilities are selected to form the lower
layer of the framework. This is done in separate layers to allow system users to customize the framework for their different scenarios. For example, the TCP protocol and the support for IEEE 802.11X can be utilized for a centralized network scenario, and the Bluetooth protocol can be replaced for a more spontaneous and ad hoc social network. The openGL engine performs most of the visualization process. The upper layer of the framework includes the ports for connection to internal software services such as the context templates and the utility functions. Also, this layer hosts the matching algorithm, RAIMA, and the Discovery/Migration services.


Figure 3.2: The RAIMA Framework

The algorithm and service discovery components are responsible for communications between nodes and they include a pairing mechanism suitable for volatile mobile
networks. The first layer represents the communications components. The second layer within the matchmaking/discovery component includes the integrated matching algorithm to search and retrieve prioritized results in the mobile network. The third layer includes a set of template parsers, responsible for the adaptation of the matchmaking algorithm. The fourth layer stores the templates, attributes and their associated weights for specific settings. The relevancy scores generated by the framework are then sent to a visualization engine responsible for generating egocentric social graphs using a graphical engine. The generated social graphs play an important role in helping users to better understand their social surroundings and, as a result, make optimum decisions. This visualization engine is discussed in Section 3.5. In general, the framework provides the following three main functionalities:

1) Multi-hop link support to proximate nodes
2) Matchmaking functionalities to connect and notify relevant nodes
3) Automatic customization and adaptation of the matching criteria based on the user's contextual and situational information

### 3.2 Recursive Multi-Criteria Matching Algorithm in RAIMA

The algorithm and discovery components are responsible for the communications between the social network nodes, and they include a pairing mechanism suitable for volatile mobile networks. The structure of these services is shown in Figure 3.3. The first layer represents the communications components. The second layer within the
matchmaking/discovery component includes the integrated matching algorithm for searching and retrieving prioritized results in the mobile network. The third layer includes a set of template parsers, responsible for the adaptation of the matchmaking algorithm. The fourth layer stores the templates, attributes, and their associated weights for specific settings.


Figure 3.3: Structure of RAIMA

Matchmaking in RAIMA is performed through processing of profiles with weighted attributes. These profiles include attributes, which are predefined by a context template generator and are stored in the system as markup schemas. A linear scoring model operating on the profile attributes performs a similarity analysis between the profiles. For example, a profile can be composed of "Name, Gender, Age Group, and Hobbies." Other users can use the same template criteria, searching their proximate nodes based on "Gender, Age Group, and Hobbies." Each attribute is associated with a weight score that
changes the impact of the attribute in the selection and ranking process. This is shown in Table 3.1.

Table 3.1: User's Profile and Search Criteria

| User's Profile |  |  |  |  |
| :--- | :--- | :--- | :--- | :--- |
| Name | Gender | Age Group | Hobbies | Image |
| John | M | $18-24$ | A,B,C | John's Image |


| Search Criteria |  |  |
| :--- | :--- | :--- |
| Gender | Age Group | Hobbies |
| Female | $18-24$ | A,B,C,D |

A profile allows the communicating nodes to calculate a numeric measure called the conceptual distance. This is a score relating to common elements in users' profiles. The higher the conceptual distance, the more those nodes have in common.

Equation 3.1 shows how profile attributes are stored in a profile matrix, P .

$$
\begin{equation*}
P=\left[x_{1}, x_{2}, \ldots, x_{n}\right] \tag{3.1}
\end{equation*}
$$

where, $x_{i}$ is a profile attribute represented by the context generator.
Weight vector of $n$ profile attributes in equation (3.2) is given by a one-column $W_{p}$ matrix as follows:
$\boldsymbol{W} \boldsymbol{p}=\left[\begin{array}{c}w_{1} \\ w_{2} \\ \vdots \\ w_{n}\end{array}\right]$

The multiplication of the one-row profile matrix of every person $D_{s}$ by a one-column weight matrix yields the result of the weighted attributes for calculating the conceptual distance. This is shown in equation

$$
D_{s}=P . W p=\left[x_{1}, x_{2}, \ldots, x_{n}\right]\left[\begin{array}{c}
w_{1}  \tag{3.3}\\
w_{2} \\
\vdots \\
w_{n}
\end{array}\right]
$$

The conceptual distance calculation is the result of the multiplication of the Weight matrix $(W)$ and the Profile Matrix $(P)$. The weight Matrix is the importance of each attribute in that particular analysis. For example, consider a profile comparison in a conventional dating scene, where finding a person with a particular gender is more significant than finding someone with the same age bracket.

Therefore the weight matrix associates a higher weight to "gender" compared to "age." This decision matrix is shown in (3.4). The matrix resolves a conceptual distance for any profile matrix combination whether the elements of P or W are static or varying.

$$
\left[\begin{array}{c}
C D_{1}  \tag{3.4}\\
C D_{2} \\
\vdots \\
C D_{n}
\end{array}\right]=\left[\begin{array}{cccc}
P_{11} & P_{11} & \ldots & P_{1 m} \\
P_{21} & P_{22} & \ldots & P_{2 m} \\
\vdots & \vdots & & \vdots \\
P_{n 1} & P_{n 2} & \ldots & P_{n m}
\end{array}\right]\left[\begin{array}{c}
W_{1} \\
W_{2} \\
\vdots \\
W_{n}
\end{array}\right]
$$

Each node's profile details its relevant attributes. These can be of many forms including numeric, descriptive, or abstract.

If the comparison between two attributes is a numeric comparison, the conceptual distance is calculated by:
$A(t)=1-\frac{\left(a_{n}(t)-a_{m}(t)\right)}{f s}$
In many cases the attribute is a numeric range to ensure the profile's consistency. Age groups and salary ranges are examples of numeric ranges. For these attributes, the conceptual distance is calculated as follows:
$B(t)=1-\frac{\left(b_{n}-b_{m}\right)}{N}$

If the attributes are selected from a predefined exclusive lists or a predefined mutually exclusive menu structure, the conceptual distance is calculated as follows:
$C(t)=c_{n}(t) \pm c_{m}(t)$

If the attributes are selected from an array-based or linked list structure with no exclusivity, the conceptual distance is calculated as follows:
$D(t)=1-\frac{\sum_{n-1}^{N} d_{n} \wedge d_{m}}{N}$

Descriptions and examples of the RAIMA formulas and their structure are explained and summarized in Table 3.2 and 3.3. Each attribute is fitted into a number of categories defined during the profile generation. This is shown in Table 3.2.

Table 3.2: Nodal Categories

| Category | Example | Description |
| :--- | :--- | :--- |
| Numeric | $1,2,3 .$. Etc | A numeric manipulation based off a <br> single value is performed |
| Numeric Range | $1-2021-23$ etc | A numeric manipulation based off a <br> range of values is performed |
| List | Artificial <br> Intelligence, <br> System Design | A flexible list of interests is <br> compared for similarities |
| Exclusive List | Male to Female | A predefined list of traits is <br> compared for converse similarities |

Profile selection is an important part of the process as it involves explicitly defining the importance of each context to the overall system and defining the weights for the $W$ matrix (see equation 3.4). Each context that affects the system's decision needs to be considered. If any factors are overlooked or not included, then the system will ignore them as being not applicable in making a determination. The P matrix is resolved by comparing each node's profile with all discovered trusted nodes present in the system.

Each node covers a large range of comparisons in order to calculate the conceptual distance effectively. Each profile attribute that plays a role in score calculation is considered in the P matrix.

Table 3.3 shows the definitions associated with each profile's attribute and the appropriate comparison algorithm. For example if first weight $W_{1}$ was an age comparison, the profile would define it as a type Numeric or Numeric range; therefore the results in the P matrix for column one would be the results of equation $\mathrm{A}(\mathrm{t})$ or $\mathrm{B}(\mathrm{t})$ where if $\mathrm{W}_{2}$ was a comparison of gender, the P matrix second column would be the results of $\mathrm{C}(\mathrm{t})$. Each column in the P matrix represents a profile attribute. The attribute type can be numeric, a numerical range, an exclusive list, or a list. This is shown in Table 3.3.

Table 3.3: RAIMA Comparison Algorithms

|  | Type | Comparison Algorithm |
| :---: | :---: | :---: |
| A | Numeric | $A(t)=1-\frac{\left(a_{n}(t)-a_{m}(t)\right)}{f s}$ |
| B | Numeric Range | $B(t)=1-\frac{\left(b_{n}-b_{m}\right)}{N}$ |
| C | Exclusive List | $C(t)=c_{n}(t) \pm c_{m}(t)$ |
| D | List | $D(t)=1-\frac{\sum_{\mathrm{n}-1}^{\mathrm{N}} \mathrm{~d}_{\mathrm{n}} \Lambda d_{\mathrm{m}}}{N}$ |

$A(t)$ is the final score of two numeric values: $a_{n}(t)$ and $a_{m}(t) . a_{n}(t)$ is the main value, which is used to divide $a_{n}(t)-a_{m}(t)$ and obtain the relative difference. The absolute value is then subtracted from 1 to obtain a percentage of the similarity, compared to the main value $\mathrm{a}_{\mathrm{n}}(\mathrm{t}) . \mathrm{B}(\mathrm{t})$ is the final score of two numeric values, defined by their index in a predetermined range of numeric values. The index of the first value $\left(b_{n}\right)$, is subtracted from the index of the second value $\left(\mathrm{b}_{\mathrm{m}}\right) . \mathrm{N}$ is the total number of partitions in the range. Each index ( $\mathrm{b}_{\mathrm{i}}$ ) will be $[0 \leq \mathrm{bi} \leq \mathrm{N}-1]$. The relative difference and percentage is calculated similarly to $\mathrm{A}(\mathrm{t}) . \mathrm{C}(\mathrm{t})$ is the score of two Boolean values, with a predetermined score with opposite signs. The calculations of $\mathrm{C}(\mathrm{t})$ returns the absolute value of the difference between the two values, based on their score representations. In our example presented
below in Table 3.4, we chose 0.5 for Male and -0.5 for Female values, resulting in a score of 1 for opposite values, and a score of 0 for matching values. $\mathrm{D}(\mathrm{t})$ is a direct comparison of string elements. The score is a result of counting how many of the primary list's elements $\left(d_{n}\right)$ exists in the secondary list $\left(d_{m}\right)$. The relative difference is calculated by dividing the number of matches by the number of elements in the primary list. The final score is produced by multiplying each individual score by its predetermined weight, as shown in equation (3.4).

Table 3.4: Definitions

| Definitions |  | Example |
| :--- | :--- | :--- |
| $\mathrm{a}_{\mathrm{n}}$ | $\begin{array}{l}\text { Is the numeric value or function associated } \\ \text { with the attribute }\end{array}$ | e.g. Age $=12$ |
| $\mathrm{~b}_{\mathrm{n}}$ | $\begin{array}{l}\text { Is a numeric value associated with the range } \\ \text { set the numeric value fills }\end{array}$ | $\begin{array}{l}\frac{\{. .20\}}{b=\mathbf{0}} V \frac{\{21 . .30\}}{b=1} V \frac{\{31 . .40\}}{b=2}\end{array}$ |
| $\mathrm{c}_{\mathrm{n}}$ | Is a single value tied to a binary comparison | $\begin{array}{l}\text { Male }=0.5 \\ \text { Female }=-0.5 \text { or } \\ \text { Mobile }=0.5 \\ \text { Immobile }=-0.5\end{array}$ |
| $\mathrm{~d}_{\mathrm{n}}$ | $\begin{array}{l}\text { is a range of skills/likes associated with the } \\ \text { node }\end{array}$ | $\begin{array}{l}\text { e.g Hobbies, }\end{array}$ |
| N | This is the number of objects in the data set | $\begin{array}{l}\text { e.g. the Numeric range above has an } \\ \text { example } \mathrm{N}=3\end{array}$ |
| n | Is the node doing the comparing of skills for D |  |$\}$

A social context can be represented by a set of categories $A=\left\{A_{1}, A_{2}, \ldots, A_{m}\right\}$, where each category contains number of subcategories. In RAIMA, each category can be assigned a
priority in accordance with user's choices. Since the algorithm is recursive, subcategories are also associated with a weighted priority as described in Equation 3.2.

A set $\left\{A_{1}, A_{2}, \ldots, A_{m}\right\}$ has $2^{m}$ subsets( provable by induction), which can be grouped in a new combined set $\left\{F_{1}, F_{2}, \ldots \ldots, F_{2}{ }^{m}\right\}$. This new set can be divided into two other subsets $F_{1}$ and $F_{2}$, where $F_{1}$ is a set of categories to which priorities were assigned by the first user and $F_{2}$ is the set of categories that were not selected.

Therefore: $B_{1}=\left\{B_{1}, B_{2}, \ldots, B_{n} \mid n \leq 2^{m}\right\} \subseteq\left\{F_{1}, F_{2}, \ldots \ldots, F_{2}{ }^{m}\right\}$ and $\left\{B_{1} \cap B_{2}\right\}=\varnothing$. The attribute with the lowest priority within the context receives the weight of $S_{n} x$, where $x \in(0,+\infty)$ and $S_{n}=\left|B_{n}\right|$. Starting from the lowest priority within the set, we assign weights by the increment of $x$ (the attribute above the lowest priority have the weight $2 S_{n-1} x$ ).

Let $\mathcal{W}=\left\{W_{1}, W_{2}, \ldots W_{o}\right\}$ be a set of weights obtained by comparing two users' priorities, which can be rearranged in the sequence $W$ of increasing weights due to better matches, where $c_{i}$ is the number of subcategories in each $i$ 's category. Note, if two priorities differ by $k$ orders of magnitude, then their weights differ by $k x$.

Therefore,
$\sup (W)=\sum_{i=1}^{n} i S_{i} x$

Continuing the recursion, the set $\left\{F_{1}, F_{2}, \ldots \ldots, F_{2}^{m}\right\}$ can be divided into subsets $H_{1}$ and $H_{2}$, where $H_{1}$ is a set of categories to which priorities were assigned by the first user and $H_{2}$ is the set of categories that were not selected.
$H_{l}=\left\{H_{1}, H_{2}, \ldots ., H_{p} \mid p \leq 2^{m}\right\} \subseteq\left\{F_{1}, F_{2}, \ldots . ., F_{2}^{m}\right\}$ and $\left\{H_{1} \cap H_{2}\right\}=\varnothing$. If $\left\{B_{1} \cap H_{1}\right\} \neq \varnothing$, the elements of $F_{1}$ and $H_{1}$ will be ordered according to their priorities. This model is shown in Table 3.5.

Table 3.5: Recursive calculation of conceptual distance between two users

| Priority 1 | $1 .$ <br> 2. $\mathrm{S}\|\mathrm{~B} \sigma(1)\|$ |  | Priority 1 | 1. <br> 2. $\mathrm{S}\|\mathrm{H} \sigma(1)\| .$ |
| :---: | :---: | :---: | :---: | :---: |
| Priority 2 | 1. <br> 2. $\mathrm{S}\|\mathrm{~B} \sigma(2)\|$ |  | Priority 2 | 1. <br> 2. $\mathrm{S}\|\mathrm{H} \sigma(2)\| .$ |
| $\ldots$ | $\ldots$ |  | $\ldots$ | $\ldots$ |
| Priority of $B \sigma(n)$ | 1. <br> 2. $\mathrm{S}\|\mathrm{~B} \sigma(\mathrm{n})\|$ |  | Priority of $\mathrm{H} \sigma(\mathrm{p})$ | 1. <br> 2. $\mathrm{S}\|\mathrm{H} \sigma(\mathrm{p})\| .$ |

From here on we find $\left\{B_{1} \cap H_{l}\right\}=\left\{G_{1}, G_{2}, \ldots, G_{l} \mid \leq_{2}^{m}\right\}$. Elements that are not in the set $H_{l}$ but are in $\beta_{1}$ are giving a conceptual score zero. The elements that constitute the union of
$F_{1}$ and $H_{1}$ are assigned scores relative to what priority they have in $F_{1}$ and $H_{1}$. The calculation of scores is given by the formula:
$\frac{i x}{2^{a}}=$ attribute weight
Where $i x$ is the score assigned to each element in a priority and $a$ is the number of the positions by which this element lost its priority with respect to the choice of the second person. Since the algorithm runs recursively, the same procedure will be applied to every set of categories, so that the elements of it with the lowest priority will be assigned the value of $y$. Each $y$ will be calculated in terms of $x$, where the maximum score is
$\sup (Y)=\sum_{i=1}^{\boldsymbol{k}} \boldsymbol{i} \boldsymbol{S}_{\boldsymbol{i}} \boldsymbol{y}=j_{x}$
where $Y$ is the sequence of scores obtained matching $k$ subcategories and $j_{x}$ is the points value that can be got from comparison of categories, where $j \in \boldsymbol{R}$. RAIMA divides the network nodes into consumers and providers. A consumer node is a node in the state of calculating its conceptual score against other nodes. A provider node is a node in the state of provisioning its profile to other consumer nodes.

Now all weights that are attained by comparison can be converted to the conceptual distance (how similar two users are based on their preferences).

$$
\begin{equation*}
\frac{1}{W_{i}}=D_{i} \tag{3.12}
\end{equation*}
$$

RAIMA changes the state of the nodes randomly to ensure the availability of all profiles within the network. A node is moved to a terminated state if it is removed from a context template. The context is defined by the context generator, which is defined in Section 3.3.


Figure 3.4: Nodes states in the RAIMA Framework

Table 3.5 provides an example of how matching is performed between consumers and providers in the RAIMA Framework. Consumer nodes (Node 1 and Node 2) request the same service type (Casual Relationship) concurrently. Preference of Node 1 about Casual Relationship is higher than Node 2. In this case, RAIMA pairs Node 1 with the provider Node 11 since Node 11 is providing the required service attribute more than other provider nodes. The numbers in parentheses are service types, which indicate the desire of the consumer node for a particular attribute, in this case the "Relationship Type." Because the service attribute of Node 1 has its score as 95 , Node 11 matches Node 1 . After that Consumer Node 2 and Node 11- are matched because peer 2 has an alternative preference for the Age Group 25 to 30.

Table 3.6: RAIMA Service matching with alternation

| Consumer |  |  |  |
| :--- | :---: | :---: | :---: |
| Node | Service | Alternation |  |
| 1 | Casual <br> Relationship(95) | AgeGroup- <br> 25to30() |  |
| 2 | Casual <br> Relationship(90) | Being-Male() |  |
| 3 | AgeGroup- <br> 25to30(80) | Hobby- <br> RubyOnRails |  |
| 4 | Hobby-Hiking(75) | Casual <br> Relationship() |  |
| Node | Provider |  |  |
| 10 | Service | Alternation |  |
| 11 | Casual Relationship <br> (100) | AgeGroup- <br> 25 to30() |  |
| 12 | AgeGroup-25to30 <br> (70) | Hobby- <br> RubyOnRails(60) |  |
| 13 | Being-Male() |  |  |

Service types used in the previous example are shown in Table 3.6. The example shows how two attributes, "Age Group and Relationship Types" results in alternative matching when the ideal match is not available. These two are distinct attributes used for the searching.

Table 3.7: Multi-Criteria Social Matching with Three Attributes

| Node | Service | Age | Relationship <br> Type |
| :---: | :---: | :---: | :---: |
| 1 | Being_Male(95) | $18-$ | Casual |
|  |  | 24 |  |
| 2 | Being_Male(90) | $24-$ | Casual |
|  |  | 30 |  |
| 3 | Being_Male(75) | $30-$ | Serious |
|  |  | 40 |  |
| 4 | Being_Female(87) | $70-$ | Serious |
|  |  | 100 |  |

While stable marriage algorithms just select a service by using consumer attributes, the multiple matching shown in RAIMA uses all attributes of both the consumer and the provider nodes. Profile selection is an important part of the process as it involves explicitly defining the importance of each context to the overall system and defining the weights for the W matrix (see (3.4)). Each context that affects the system's decision needs to be considered.

If any factors are overlooked or not included, the framework ignores it as being not applicable in the scoring process. The P matrix is resolved by comparing each node's profile with all discovered trusted nodes present in the system.

### 3.3 Generation of Social Context

One of our design goals was to adapt the matchmaking algorithm based on the user's social setting (e.g., the location of the user). This means that the attributes used in the matchmaking process should dynamically change when the user is located in different social settings. For instance, a student user may use attributes such as Research Interests and Study Major in an academic setting and use attributes such as Age, Hobbies, and Relationship Status while in a social setting. Using a markup-based template generator enables the framework to customize these attributes upon the detection of the users' context. Profile templates can be created using a Web service for different settings and events. This is shown in Figure 3.5.


Figure 3.5: Context Generation for distinct social scenarios

Figure 3.6 shows how the attributes are associated with a weighted value. For example, the Age attribute is associated with a weight of 25 and Gender is associated with a weight of 75 . Event organizers can use this web service to customize their own templates and associated attributes. Event organizers prioritize the weights by selecting priorities such as Highest, Medium, or Lowest from a drop-down menu. The template generator will associate a weight to the attribute based on the number of inputted attributes. The output of this web service is an XML file capable of customizing the matchmaking algorithm for different scenarios. The provisioning model of these schemas is shown in Figure 3.7.

```
<?xm7 version=1.0>
<profile>
    <id>B1ank1</id>
    <Numeric>
                        <id>Age</id>
                <weight>25</weight>
                            <item><value>25</value></item>
    </Numeric>
    <Exclusive List>
                <id>Gender</id>
                <Weight>75</Weight>
                        <item><Name>M;F;</Name></item>
    </Exclusive List>
</profile>
```

Figure 3.6: A context template based on three social attributes

Figure 3.6 shows the overall structure of the context-provisioning model in RAIMA. Each event is connected through a proxy to a static context generator. Event organizers are granted access to the context generator through the proxy and can create the event template using simple web interfaces. Events, each represented as a cloud in Figure 6, are enabled with a Wi-Fi or Bluetooth access point responsible for the provisioning of event templates to attendees. This ensures consistency in scoring and calculation, as it is ensured that everyone will use the same template at the event. As users enter each event, a social graph is visualized on their mobile devices representing their similarities with proximate nodes. Templates discussed in this section play an important role in the consistency of these graphs.


Figure 3.7: RAIMA Deployment Model for three distinct social context

Figure 3.8 shows how the context provisioning model is mapped on a user interface. The user can select a geographical area and associate it with a context template as shown in Figure 3.6. This way, each zone can be associated with a unique context template, and the user within the range receives a copy of the context template. Figure 3.8 shows how a host user can select a location and create a template for it.


Figure 3.8: User interface for the context generator

The context semantics are generated through information elicitation for each domain. The provisioned context template includes a series of questions, which are dynamically weighted by the template generator. Users' responses to questions result in the generation of matchmaking semantics in a particular context domain. The questions may have a single or multiple answers. The following example shows how a context semantic is generated in a scenario where the users are matched based on their interests. For
attributes with a single possible response where $n$ is the priority of the attribute in order of importance, $r$ is the number of possible responses and $W$ is the associated weight with the attribute.
$\frac{\boldsymbol{W}(\boldsymbol{r}-\boldsymbol{n})}{\boldsymbol{r}}$
For attributes with two possible responses:

$$
\begin{equation*}
\frac{w_{s_{1}}\left(r-n_{1}+1\right)+w_{s_{2}}\left(r-n_{2}+1\right)}{2 r-n_{1}+2-n_{2}} \tag{3.14}
\end{equation*}
$$

For attributes with $k$ possible responses,
where $R$ is list of chosen responses (i.e. $R=\{b, f, w, x\}$ and k is size of $R(i . e ., k=4)$ ):

$$
\begin{equation*}
\frac{\Sigma_{i=1}^{k} w_{s_{i}}\left(r-n_{i}+1\right)}{k(r+1)-\left(\Sigma_{i=1}^{k} n_{j}\right)} \tag{3.15}
\end{equation*}
$$

The user inputs profile attributes and ranks its references based on the provisioned template. Figure 3.9 shows how this model is mapped on a user interface for a RAIMA user.


Figure 3.9: User interface for a RAIMA user with 3 attributes

### 3.4 Pairing and Discovery Modules

In order to achieve optimum usability in a mobile setting, preference was given to a pairing process not requiring excessive user involvement, especially during the handshaking process. To demonstrate the functionality of the framework and its pairing process, we have selected protocols capable of forming ad hoc network meshes which also have standard programming interfaces in Java to make them reconfigurable in our framework. For this reason, we have chosen IEEE 802.11X and Bluetooth as the
protocols of choice to be able to demonstrate the performance of the framework in a volatile mobile setting. The standard protocol's pairing process enforces the use of a key for ad-hoc pairing. This implies that in settings with multiple nodes, a user has to continuously approve and monitor connection requests and share a key with all the requestors.

This scheme is not favorable in our framework from a usability perspective, as there is usually a large number of users exchanging profiles and users tend to not want to spend excessive time pairing devices with other proximate users. In order to make this process seamless and yet maintain security in the communications, we have modified the authentication process by overriding the Bluetooth pairing and instead developed a hashkey generator to broadcast a shared key to proximate users who are using the same software client. Searching involves using L2CAP [67] as the physical layer and combination of SDP [68] and TCP/IP for higher layers. Figure 3.10 shows how the proposed pairing process works in an ad hoc setting.


Figure 3.10: Ad Hoc pairing using regular expressions

Initially, the client retrieves the UID (Unique Identifier) of all proximate devices through the protocol's discovery layer. This step does not require any handshaking between devices and does not impose a large latency on the network. The template generator, which is usually stored on a fixed terminal, generates a regular expression and a hashkey, which is used as a common key between all the clients running the software. Devices sharing the same key override the Bluetooth pairing process and enable trusted communications among each other. In case the communications are in a centralized network, the same pairing process can be used between the clients and servers in the network. The server calculates the conceptual distance, and the results are passed to each
client. In this case, visualization is the only resource-intensive process executed on the clients. This is shown in Figure 3.11.


Figure 3.11: Pairing Process in the Client/Server mode

During the last step of pairing, all profile attributes are distributed between trusted nodes, and each client calculates its relative score against other users. We refer to this inferred score as the Conceptual Distance, which is discussed in Section 3.2. The Conceptual Distance is a linear distance between two nodes representing their overall similarity. In the social graphs discussed in Section 3.5, a long conceptual distance indicates less
similarity and a short conceptual distance indicates common attributes between the profile nodes. The pairing module plays a key role in connecting trusted nodes to each other and eliminating the non-trusted ones by verifying the shared regular expression. It is an important module in the framework, since it performs these tasks without relying on a centralized communications network. This increases the practicality of the framework in ad hoc and volatile mobile networks.

### 3.5 Visualization

Visualization of a social network is an important feature for users, as it increases the users' ability to make inferences from the social network. Research works in the structural analysis of social networks divide social networks into socio-centric (population-based) and egocentric (user-centered). In this section we address which model is more applicable in our system design. Figure 3.12 shows the difference between a socio-centric network and an egocentric network.


Figure 3.12: Egocentric vs. Socio-centric Social Networks

Criteria for the selection of these graph representations are based on the users' feedback and how much they infer from their social surroundings by analyzing these graphs. Socio-centric approaches represent all relations between individuals in a population such as members of an organization. Based on research works such as [69], it is demonstrated that socio-centric representations diverge from individual users' expectations. [69] suggests that users prefer personal social network visualizations that show how close or far other users are with respect to the user him- or herself. This suggests that egocentric rather than socio-centric visualization models are preferable in mobile social networks.

Structural analysis of social networks is based on graph theory, in which people are shown as nodes and their connections as vertices. The RAIMA visualization engine uses spring embedding techniques in a Cartesian coordinate system to represent social graphs in egocentric formats. In egocentric social networks, relationships are shown with respect to the focus user, and the graphs are processed for each user separately. Like most social networks, circles represent profile nodes, and vertices show the relationship between the users. In RAIMA, the vertices between nodes denote the inversed conceptual distance between users. A short distance indicates higher similarities between nodes and long ones indicate less similarity and more differences.

The focus user is typically illustrated at the center of the graph, usually on coordinate $(0,0)$. The relationships can be presented both in two-dimensional and three-dimensional spaces. 2D representation is used for one interaction context, and 3D graphs are used to analyze conceptual distances in different contexts. This means that the user can analyze
other people based on different criteria by processing each pane of the graph separately. Figure 3.13 depicts how a social decision space is visualized in RAIMA. The frame boundaries are the threshold of the conceptual distance, indicating the highest conceptual score possible in a social context. Each plane of the decision space represents a context of interest for the user.


Figure 3.13: A Cartesian Social Decision Space with 8 Context Templates

We note that in this model no privacy information is revealed about the user. The system uses personal information to calculate the distance, but this information is not available to users in the form of raw data. Figure 3.14 shows a social structure with two nodes. The user of the system is shown on the center of the graph as the focus user, and his
conceptual distance is shown against another user within a context. Both nodes are represented in form of a Cartesian triplet pair. Figure 3.14 shows how social similarity is shown between the focus user and a proximate user in one context. If the proximate user provides more profile templates, the visualization engine can present these in up to 8 context-based decision domains.


Figure 3.14: Conceptual Distance between two users in one social context

The RAIMA visualization engine can draw the social networks nodes in their context domains using Cartesian geometric surfaces. In a Cartesian form, nodes are presented in a 3-Dimensional Cartesian Space with paired triplets as follows:

## $\left[\begin{array}{l}x \\ y \\ z\end{array}\right]$

where, $\sqrt{x^{2}+y^{2}+z^{2}}$ is the distance from the focus user to the target user.
Figure 3.15 shows how a focus user can analyze one proximate user using three different context templates. The same representation can be made using Cylindrical or Spherical equations in which the social context is implied by the radial distance and angular position. In our simulations we represent each user with a unique color and use the Cartesian planes as context templates Figure 3.16 shows how the same process applies to multiple users in one social network with multiple context domains. The vertices represent the conceptual distance.


Figure 3.15: Conceptual Distance between two users in 3 context domains


Figure 3.16: Conceptual Distance between two users in 3 context domains

RAIMA's architecture supports pluggable ontologies. The default ontology used in RAIMA that calculates similarity between users can be replaced by other socio-centric ontologies such as FoAF. This means that the RAIMA users can select different inference models, represent them as a supported ontology, and use RAIMA's visualization engine without excessive integration. For example, Figure 3.17 shows how two ontologies are used to represent two types of inferences from a social network. Figure 3.18a shows how node 3 has more in common with node 8 and 10 and less in common with nodes 7 and 4 . The user can see the social structure in real time and see his or her best match in the social setting. Figure 3.18 b shows how a different ontology is being used to show
available and eager-to-talk users within a network. This is a presence ontology that users can choose to approach nodes who are willing to participate in a social interaction.

The calculation of scores and generation of the graphs are processed locally on the device to reduce network latency. This means that the user can see some similarities between him- or herself and proximate users, but no explicit information is revealed unless the users agree on a more trusted level of communications. Users can click on each profile's object (picture or graphical representation) to initiate a chat session, and they can increase the level of trust to access more information while they are in the social setting. The right portion of the figure shows how this is represented to the user. The user can click on a button to traverse the graph and see the next available matches. The graphs are generated using Scalable Vector Graphics (SVG) and openGL engines available on most computing platforms.

(b)

Figure 3.17: Pluggable ontologies used to create different semantics in one social structure.

### 3.6 A RAIMA Case Study

One case study is presented in this chapter to better explain the RAIMA process for the reader, and a complete set of analysis is provided in Chapter 4. All three provided case studies have been simulated in our lab, and their results are shown. The following case study shows how the process shown in this chapter works in a simulated scenario.

The following scenario is based on a fictional character, John, who travels alone from Toronto to New Zealand to attend a conference. First, John starts his journey on a plane, where he uses the software in a travel scenario while on board an airplane, then at a conference, and at the conclusion of his trip in a social setting, a cafe, to perform different social interactions.

John is on board an Air Canada flight from Toronto to London to catch a connecting flight from London to Otago. Despite the fact that there is no cellular coverage inside the plane, the air crew will allow John to use his PDA using Bluetooth. John activates his matching engine, which performs a search for new profiles available for this flight. The application software notifies John's device that the Air Canada template is available for flight AC994. John retrieves the profile using an OBEX file transmission and starts customizing it. John is interested in finding people on the flight who have a connecting flight in London and with whom he can interact with during his transit time in London. John saves his search criteria and sees that there are 8 active profiles on board, of which 4 have a short conceptual distance to him, meaning matching his criteria. Each mobile node can specify if it wants to be notified in case there are any matches and interested parties,
so John does not disturb anyone who matches his profile but does not want to be disturbed. Table I shows the 8 people with active profiles, of which 4 have a relatively small conceptual distance to John. John can see each best match, generate a chat session, and, if appropriate, approach the person.

Although personal information is used to calculate the conceptual distance, this information is not available to other users in form of raw data. This means that the user can see some similarities between him/her and proximate users, but no explicit information is revealed, unless the users agree on a more trusted level of communications.

Table 3.8: Flight Scenario Data and Scores (best scores highlighted)

| Name | City | Date/Time | Duration | Activities | Score |
| :--- | :--- | :--- | :--- | :--- | :--- |
| Weight | $\mathbf{3 0}$ | $\mathbf{2 0} / \mathbf{1 0} / \mathbf{0 8}$ | $\mathbf{1 0}$ | $\mathbf{1 0}$ |  |
| Hosein | London | $320 / 20: 00$ | 6 | Show |  |
| Stephan | London | $320 / 20: 00$ | 5 | Dinner, Cafe | $\mathbf{1 . 4 6}$ |
| Steve | Paris | $319 / 11: 00$ | 2 | Café | $\mathbf{3 . 4 7}$ |
| Maggie | London | $320 / 20: 00$ | 7 | Show | $\mathbf{1 . 2 8}$ |
| Willy | Hong Kong | $319 / 07: 00$ | 7 | Dinner | $\mathbf{3 . 1 7}$ |
| Ralph | Paris | $319 / 11: 00$ | 1 | Café | $\mathbf{3 . 6 8}$ |
| Chris | London | $320 / 19: 00$ | 5 | Show | $\mathbf{1 . 2 8}$ |
| Tobi | London | $320 / 20: 00$ | 4 | Show, Dinner | $\mathbf{1 . 3}$ |
| Sophia | Paris | $319 / 11: 00$ | 1 | Dinner | $\mathbf{3 . 6 8}$ |

The shortest conceptual distance occurs between John and Maggie, with 1.28, followed closely by Tobi, with 1.4. The longest conceptual distance for John occurs between Sophia and Ralph, with 3.68. For time comparison, a numeric range is used to calculate their relevance. The range is composed of 10 segments of the 24 -hour time range, as indicated in Table II. Because the time attribute's relevance is directly impacted by the Best Date score, the Best Time score is weighed with the Best Date score, along with a
weight of 10, or half of the Best Date weight. Table III explains the calculations in detail. In order to determine if two passengers are going to be in one location during a stopover, we could have used the flight number and obtained a highly weighted score, but that would offset any other flights with a very similar stopover schedule but having a low score due to a different Flight \#. Instead, any Flight \# calculation was given a weight of zero and is not included in the description. The actual date, time, place and duration were used to determine the conceptual score of this relationship and to give an approximate score for the relationship.

The Best Date value d 1 is the date that suits you best. It is then compared with d 2 , which is the date that suits the other person best. Best Time attributes t 1 and t 2 work in the same manner but are multiplied by the Best Date score, since the time is relevant only if the Best Date calculation returns a high conceptual score. M1 is 9, the number of 2.3 hour time segments as described in Table 3.9 , which make up the 24 -hour day. The relative difference between the index of t 1 and t 2 in the range is used to calculate the score. Duration attributes d1 and d2 indicate the user's own duration at the location, in hours, and the other person's, respectively. The relative difference between the two values is calculated and weighed against its associated weight of 10 , as well as the Best Date's score. The City attributes $\mathrm{C}_{\mathrm{x}}$ and $\mathrm{C}_{\mathrm{y}}$ are actually a list of cities where a particular person will be on the given date and time. In this scenario, since a person can be in only one location at a time, the list has a single element in set $\mathrm{C}_{\mathrm{x}}$, which is compared with the single element in set $\mathrm{C}_{\mathrm{y}}$.

Table 3.9: Time Range Distribution

| $\mathbf{1}$ | $\mathbf{2}$ | $\mathbf{3}$ | $\mathbf{4}$ | $\mathbf{5}$ |
| :--- | :--- | :--- | :--- | :--- |
| $0.0-2.3$ | $2.3-4.6$ | $4.6-6.9$ | $6.9-9.2$ | $9.2-11.5$ |
| $\mathbf{6}$ | $\mathbf{7}$ | $\mathbf{8}$ | $\mathbf{9}$ |  |
| $11.5-13.8$ | $13.8-16.1$ | $16.1-18.4$ | $18.4-20.7$ | $20.7-24.0$ |

The Activities sets Ax and Ay are list of attributes that represent the user's own activities against other people. Because someone can have more than one activity attribute, this score represents how many of the user's activities match other users. $n(X)$ gives the cardinality of X , representing all the profile elements in X . This corresponds to the equation $n(A x U A y)$, that represents the number of elements that $A x$ and Ay have in common. We then calculate the relative difference and conceptual distance between the number of common elements between the two sets.


Figure 3.18: Generated Egocentric Graph for the flight scenario case study

Table 3.10: Flight Scenario attributes scoring algorithms and sampled calculations

| Name | Weight <br> (w) | Formula | Maggie | Tobi | Sophia |
| :---: | :---: | :---: | :---: | :---: | :---: |
| Best Date <br> $\mathrm{b}_{1}=$ self <br> $\mathrm{b}_{2}=$ other | 20 | w * $\left\{1-\left[\left(\mathrm{b}_{1}-\mathrm{b}_{2}\right) / \mathrm{b}_{1}\right]\right\}$ | $\begin{aligned} & 20 *\{1-[(320- \\ & 320) / 320]\} \\ & =20 \end{aligned}$ | $\begin{aligned} & 20 *\{1-[(320- \\ & 320) / 320]\} \\ & =20 \end{aligned}$ | $\begin{aligned} & 20^{*} \\ & \{1-[(320-319) / 320]\} \\ & =19.94 \end{aligned}$ |
| $\begin{aligned} & \text { Best Time } \\ & \mathrm{b}_{3}= \\ & \text { date score } \\ & \mathrm{t}_{1}=\text { self } \\ & \mathrm{t}_{2}=\text { other } \\ & \mathrm{M}_{1}= \\ & 9=n \text { (segments) }-1 \end{aligned}$ | 10 | $\begin{aligned} & \mathrm{w}^{*} * \mathrm{~s}_{3} * \\ & \left\{1-\left[\left(\mathrm{t}_{1}-\mathrm{t}_{2}\right) / \mathrm{M}_{1}\right]\right\} \end{aligned}$ | $\begin{aligned} & 10 * 20 *\{1-[(9-9) / 9]\} \\ & =10 \end{aligned}$ | $\begin{aligned} & 10 * 20 *\{1-[(9-9) / 9]\} \\ & =10 \end{aligned}$ | $\left\{\begin{array}{l} 10 * 19.94 * \\ \{1-[(9-5) / 9]\} \\ =5.538 \end{array}\right.$ |
| $\begin{aligned} & \text { Duration } \\ & d_{1}=\text { self } \\ & d_{2}=\text { other } \end{aligned}$ | 10 | $\begin{aligned} & \mathrm{w}^{*} * \mathrm{~b}_{3} * \\ & \left\{1-\left[\left(\mathrm{d}_{1}-\mathrm{d}_{2}\right) / \mathrm{d}_{1}\right]\right\} \end{aligned}$ | $\begin{aligned} & 10 * 20 *\{1-[(6-7) / 6]\} \\ & =8.333 \end{aligned}$ | $\begin{aligned} & 10 * 20 *\{1-[(6-4) / 6]\} \\ & =6.667 \end{aligned}$ | $\begin{array}{\|l\|} \hline 10 * 19.94 * \\ \{1-[(6-1) / 6]\} \\ =8.333 \end{array}$ |
| $\left\lvert\, \begin{aligned} & \text { City } \\ & C_{x}=\text { self } \\ & C_{y}=\text { other } \end{aligned}\right.$ | 30 | $\mathrm{w}^{*}\left(\mathrm{n}\left(\mathrm{C}_{\mathrm{x}} \cup \mathrm{C}_{\mathrm{y}}\right) / \mathrm{n}\left(\mathrm{C}_{\mathrm{x}}\right)\right.$ ) | $\begin{aligned} & 30=30 *(1 / 1) \\ & =30 \end{aligned}$ | $\begin{aligned} & 30=30 *(1 / 1) \\ & =30 \end{aligned}$ | $\begin{aligned} & 30=30 *(0 / 1) \\ & =0 \end{aligned}$ |
| $\begin{aligned} & \text { Activities } \\ & \mathrm{A}_{\mathrm{x}}=\text { self } \\ & \mathrm{A}_{\mathrm{y}}=\text { other } \\ & \mathrm{n}(\mathrm{X}) \\ & =\text { cardinality of } \mathrm{X} \end{aligned}$ | 10 | $\mathrm{w}^{*}\left(\mathrm{n}\left(\mathrm{A}_{\mathrm{x}} \cup \mathrm{A}_{\mathrm{y}}\right) / \mathrm{n}\left(\mathrm{A}_{\mathrm{x}}\right)\right)$ | $\begin{aligned} & 10 * 10 *(1 / 1) \\ & =10 \end{aligned}$ | $\begin{aligned} & 10 * 10 *(1 / 1) \\ & =10 \end{aligned}$ | $\begin{aligned} & 10 * 10 *(0 / 1) \\ & =0 \end{aligned}$ |
| Score |  | 100 / total | $\begin{aligned} & 100 / 78.333 \\ & =1.28 \end{aligned}$ | $\begin{aligned} & 100 / 76.666 \\ & =1.3 \end{aligned}$ | $\begin{aligned} & 100 / 27.14 \\ & =3.68 \end{aligned}$ |



Figure 3.19: Conceptual distances between 8 users in the flight scenario case study

## Chapter Four: RESULTS AND SIMULATIONS

In this chapter, we demonstrate our results and findings on the performance of RAIMA. RAIMA is designed to be an effective framework in opportunistic and unstructured social networks in which selection of the users are multi-criteria and the social context is dynamically changing. Complying with these specifications, we have created a social network of 50 random users with diverse profile sizes. This is presented in the selection of our sample size to be 50 social users, which satisfies the size of a spontaneous social network according to [59]. In section 4.1, we describe the simulation platform that we have developed to analyze RAIMA. In Section 4.2 we show the performance of RAIMA in recalling matches when the size of the profile attributes increase. In Section 4.3 we compare RAIMA with the stable matching algorithms, which are widely used in computer-mediated social networks. In all test cases, a good RAIMA match is considered a match in the third quartile of the social network's recall. The same threshold has been used for evaluating the performance of SMA (Stable Marriage Algorithm). In Section 4.5, we show the results of RAIMA in three distinct scenarios and the subsequent output of RAIMA's visualization engine. In Section 4.5 two case studies and their results are presented to demonstrate the multi-criteria selection process of RAIMA. The multicriteria selection of RAIMA as discussed in section 3 is based on multiple attributes and multi-objective decision-makings. Support for Multi-Attributes decision-making is provided by RAIMA and the support for Multi-Objective decision making is provided by the context generator component.

### 4.1 Simulation Environments

The simulation environment for RAIMA has been structured in a way to support both centralized and ad hoc social networks. The RAIMA controller interface allows the user to select the attributes for the social network using a control panel. The control panel allows the Focus Node, Context Template, Precision/Recall, and Threshold to be selected for evaluation of RAIMA's performance. This control panel is shown in Figure 4.1.


Figure 4.1: RAIMA Control Panel

The overall structure of the RAIMA framework is shown in Figure 4.2. The platform is divided into five sections as described in section 3. The bottom lower layer is the MultiCriteria matching algorithm, which is connected to the filtering layer capable of changing
the precision and recall of the social network dynamically. The scoring model is shown on the left, which is connected to the graphical visualization engine on the right.


Figure 4.2: Developed Architecture of the RAIMA Framework

### 4.1.1 Analysis of RAIMA with increasing profile size attributes

In our tests, we consider a recall in the 75th percentile of the matching threshold a Good
Match. The formula for the locator of the position of the observation at a given percentile, $y$, with $n$ number of users sorted in ascending order, is:
$L_{y}=(\boldsymbol{n})\left(\frac{\boldsymbol{y}}{\mathbf{1 0 0}}\right)$
where $y=3$ for third quartile match, 2 for median, and 1 for the first quartile.

Also, we consider recall to be the fraction of the matches that are relevant to the context that are successfully retrieved, and the precision is the fraction of retrieved matches that are relevant to the context. We note that the Precision and Recall in all context domains are defined as follows:
precision $=\frac{\mid\{r m\} \cap\{r d| |}{\mid\{r d| |}$
recall $=\frac{|\{r m\} \cap\{r d\}|}{\mid\{r m\}}$
where:

$$
\begin{aligned}
& r m=\text { Relevant Matches } \\
& r d=\text { Retrieved Matches in Q3 }
\end{aligned}
$$

Using this prior information, we show how RAIMA performs with different profile sizes. We have built three test cases to demonstrate the capability of RAIMA in a growing social network. We have limited the size of the network to 50 concurrent users. This is based on the fact that a user does not normally get involved in a social interaction with more than 50 users simultaneously. Also, a practical limitation was that we had 50 mobile devices available to use at the same time. The testing application is being developed for
the Apple OS X platform, and it is based on the Quartz development framework. The tests were performed in a laboratory setting with virtualized machines, each of them having 4GB of memory. The application allows the user to select the attributes using a control panel. The control panel allows the Focus Node, Context Template, Precision/Recall, and Threshold to be selected for evaluation of RAIMA's performance.

The test application and its structure is shown in Figure 4.3.


Figure 4.3: Structure of the RAIMA test

### 4.1.2 Analysis of RAIMA with 50 users, Profile Size of 10 Attributes

We have used the algorithm based on 10,50 , and 100 profile attributes respectively. The horizontal axis in Figures 4.4-4.6 represents the number of users, and the vertical axis represents the probability of a good match. The good match in the case of RAIMA is considered a score within the third quartile of the conceptual distance score. The first test case is being performed in a growing social network of size 50 , and a good match is considered a recall in the third or fourth quartile. Figure 4.4 represents the performance of RAIMA when the user profile includes 10 attributes.

Table 4.1: Test case 1

| Number of users in the <br> social network | Number of profile <br> attributes | Matching Quartile |
| :--- | :--- | :--- |
| $n=50$ | $a=10$ | $\mathrm{Q}_{3,4}$ |



Figure 4.4: Probability of Good Matches with $\mathrm{a}=10$

### 4.1.3 Analysis of RAIMA with 50 users and 50 Profile Attributes

The second test case is being performed in a growing social network of size 50, and a good match is considered a recall in the third or fourth quartile. Figure 4.5 represents the performance of RAIMA when the user profile includes 50 attributes.

Table 4.2: Test case 2

| Number of users in the <br> social network | Number of profile <br> attributes | Matching Quartile |
| :--- | :--- | :--- |
| $n=50$ | $a=50$ | $\mathrm{Q}_{3,4}$ |



Figure 4.5: Probability of Good Matches with $\mathbf{a}=50$

### 4.1.4 Third Case Study: A Small Social Network with big profile size

The third test case is being performed in a growing social network of size 50, and a good match is considered a recall in the third or fourth quartile. Figure 4.6 represents the performance of RAIMA when the user profile includes 100 attributes.

Table 4.3: Test case 3

| Number of users in the <br> social network | Number of profile <br> attributes | Matching Quartile |
| :--- | :--- | :--- |
| $n=50$ | $a=100$ | $\mathrm{Q}_{3,4}$ |



Figure 4.6: Probability of Good Matches with $\mathbf{a}=\mathbf{1 0 0}$

### 4.2 Comparative Results

We have tested the framework in different social contexts. We compared the performance of RAIMA against the Stable Marriage Algorithm (SMA), which is used in many mobile social networking applications, such as BEDD.com and Mobiluck.com. We have compared the algorithms based on 10,50 , and 100 profile attributes respectively. The
horizontal axis in Figures 4.8-4.10 represent the number of users, and the vertical axis represents the probability of a good match. A good match in the case of RAIMA is considered a score within the third quartile of the conceptual distance score and in case of SMA is the ideal match for the user. Observations from the tests indicate that RAIMA and conventional matching algorithms perform similarly as the number of users increase using 10 profile attributes or less; however, the salient difference is that RAIMA is an algorithm capable of optimal and alternative matching, while SMA-based algorithms are not. The structure of SMA-based algorithms is shown in Figure 4.7. The primary reason RAIMA outperforms SMA is that SMA is not capable of relative matching and can retrieve only exact matches. Figures $4.8-4.10$ show the competence of RAIMA against SMA algorithms as the number of profile attributes increase. This observation is evident, and it is aligned with guidelines from the social sciences that indicate that the more attributes you share with other people, the more the probability of finding similar interests increases. Also, our framework increases the probability of a match by providing consistent context templates for each social setting. RAIMA outperforms SMA algorithms in Figures 4.9 and 4.10 because the probability of finding relative matches with 50 and 100 profile attributes is higher than finding the ideal matches within the same social network.

```
function stableMatching {
    Initialize all m}\inM\mathrm{ and }w\inW\mathrm{ to free
    while }\exists\mathrm{ free man m who still has a woman w to propose to {
        w = m's highest ranked such woman
        if w is free
            (m, w) become engaged
        else some pair (m', w) already exists
            if w prefers m to m'
                (m,w) become engaged
                m' becomes free
            else
                (m', w) remain engaged
    }
}
```

Figure 4.7 Logic in the Stable Marriage Algorithm

### 4.2.1 Comparative Results Based on 10 Profile Attributes and 50 Users

Table 4.4: Comparative test case 1

| Number of users in the <br> social network | Number of profile <br> attributes | Matching Quartile |
| :--- | :--- | :--- |
| $n=50$ | $a=10$ | $\mathrm{P}_{\text {match }}=\mathrm{Q}_{3,4}$ |
| $\mathrm{~S}=50$ | $a=10$ | $\mathrm{P}_{\text {match }}=\mathrm{Q}_{3,4}$ |



Figure 4.7: Comparisons between RAIMA and SMA based on 10 profile attributes

### 4.2.2 Comparative Results Based on 50 Profile Attributes and 50 Users

Table 4.5: Comparative test case 2

| Number of users in the <br> social network | Number of profile <br> attributes | Matching Quartile |
| :--- | :--- | :--- |
| $n=50$ | $a=50$ | $\mathrm{P}_{\text {match }}=\mathrm{Q}_{3,4}$ |
| $\mathrm{~S}=50$ | $a=50$ | $\mathrm{P}_{\text {match }}=\mathrm{Q}_{3,4}$ |



Figure 4.8: Comparisons between RAIMA and SMA based on 50 profile attributes

### 4.2.3 Comparative Results Based on 100 Profile Attributes and 50 Users

Table 4.6: Comparative test case 3

| Number of users in the <br> social network | Number of profile <br> attributes | Matching Quartile |
| :--- | :--- | :--- |
| $n=50$ | $a=100$ | $\mathrm{P}_{\text {match }}=\mathrm{Q}_{3,4}$ |
| $\mathrm{~S}=50$ | $a=100$ | $\mathrm{P}_{\text {match }}=\mathrm{Q}_{3,4}$ |



Figure 4.9: Comparisons between RAIMA and SMA based on 100 profile attributes

### 4.2.4 Comparative analysis in clustering a socio-centric social network using RAIMA

Although the focus of our analysis has been egocentric networks, we have performed one set of analysis in which the framework is used to analyze matches in a socio-centric network. This shows that the number of retrieved matches in the third and fourth quartile is higher in RAIMA in comparison with marriage matching in a population of 400 users and 50 random attributes.

Table 4.7: Performance of RAIMA in a socio centric network

| Number of users in the <br> social network | Number of profile <br> attributes | Matching Quartile |
| :--- | :--- | :--- |
| $n=50$, Socio-Centric | $a=50$ | $\mathrm{P}_{\text {match }}=\mathrm{Q}_{3,4}$ |
| $\mathrm{~S}=50$, Socio-Centric | $a=50$ | $\mathrm{P}_{\text {match }}=\mathrm{Q}_{3,4}$ |



Figure 4.10: Comparison between RAIMA and SMA in a socio-centric network
After running the algorithm, we can generate graphs where the population is structured into clusters of random users. Each cluster represents a group of users that have common relationships within a particular context. Each link represents a conceptual distance between two users satisfying a certain similarity threshold. In other words, if the reciprocal conceptual distance between user A and user B is less than the threshold, then we consider users A and B to be a good match; therefore, we draw a link A-B on the graph between the two users.

By calculating the in-degree for each node (total number of links associated with a user), we represent the number of good matches associated with that node. Next, we calculate the average in-degree in a randomly chosen cluster which as high as $n-1$, where $n$ is the
total number of users in the cluster, i.e. everybody is connected to everybody. Therefore, the best cluster would the one with the average in-degree of $n-1$ or in-degree ratio $(R)=$ average in-degree/maximum average in-degree $=1$;

$$
\begin{equation*}
R=\frac{\Sigma_{i=1}^{n} D_{i}}{n D_{\max }} \tag{4.4}
\end{equation*}
$$

Where $n$ is the number of users in a given cluster,
$D_{i} \quad$ is in-degree of user $i$
and $\quad D_{\max }$ is maximum possible in-degree in a given cluster $D_{\max }=n-1$.

Figures 4.11-4.13 show how the model is applied to three distinct scenarios. Figure 4.11 shows the model is applied in a socio-centric network of 52 users. By applying the model described in equation (4.1) to 11 randomly chosen clusters, the best candidate clusters within the context are selected. For example, if we want to find the best group/cluster of 3 nodes, we would calculate R for all subset of size 3 of given population. The highest R would indicate the best clusters.


Figure 4.11: 11 clusters based on using Context Template A with R from 0 to 1
Figure 4.12 shows how RAIMA has clustered and prioritized 8 clusters based on a different social context. Cluster 1 and 2 with the highest average in-degree represent the two top candidates based on the given context. Figure 4.12 also shows that using RAIMA for clustering can indicate users that are not suitable for the social context. Example of this is the 8 users in the middle of Figure 4.12 that do not belong to any cluster.


Figure 4.12: 8 clusters based on Context Template B with $\mathbf{R}$ from 0 to 1

Figure 4.13 shows how the clusters are dynamically changed when the provisioned context template is changed from context template $B$ to $C$.


Figure 4.13: 8 clusters based on Context Template $C$ with $\mathbf{R}$ from 0 to 1

Scenarios shown in Tables $4.8-4.10$ and Figures $4.11-4.13$ show that three weighted search criteria (Age, Hobbies, and Gender) are used to calculate the final conceptual distance in a social setting of 10 people. The lowest score of 0 indicates the least desirable node. The sum of all weights identifies the top score of 150 , which indicates a
perfect match. The graphs show high scores between opposite genders and low scores between the same genders.

Although "Gender," with a high weight of 75, was the major criterion in obtaining the conceptual distances in the scenarios defined in Tables 4.8-4.10, it is not the only matching factor. "Age," with the weight of 50, and "Hobbies," with the weight of 25, are subsequent factors in providing a more accurate match in accordance with the searcher's criteria. For instance, in Table 4.1., the best match for Person 1 is Person 3, with a high conceptual distance of 144.4, and the least desirable match is Person 9, with a low conceptual distance of 38.9 . These conceptual distances play a key role in visualizing the social structures as described in chapter 3. It is important to note that the provisioned context template described in chapter 3 defines the default weight for each attribute. To be able to customize the search further, the user also has the ability to change the default weight values on a device to customize the search criteria. For example, in Table 4.2, the user can decrease the weight assigned to "Gender" and increase the weight assigned to "Hobbies" to prioritize the search to find people with hobby "D".

Table 4.8: Search space and scores for person 1

|  | Age | Hobbies | Gender | Score |
| :--- | :--- | :--- | :--- | :--- |
| Weights | 50 | 25 | 75 |  |
| Person 1 | 25 (desired) | A, B (desired) | M |  |
| 2 | 30 | C, D | M | 44.4 |
| 3 | 33 | A, B | F | 144.4 |
| 4 | 23 | D | M | 50 |
| 5 | 29 | B, C | F | 131.9 |
| 6 | 15 | A, D | F | 126.4 |
| 7 | 40 | A, B, C, D | M | 58.3 |
| 8 | 45 | A | F | 120.8 |
| 9 | 39 | C | M | 38.9 |
| 10 | 21 | B | F | 131.9 |

Table 4.9: Search space and scores for person 4

|  | Age | Hobbies | Gender | Score |
| :--- | :--- | :--- | :--- | :--- |
| Weights | 50 | 25 | 75 |  |
| Person 4 | 30 (desired) | D (desired) | M |  |
| 1 | 25 | A, B | M | 44.4 |
| 2 | 30 | C, D | M | 75 |
| 3 | 33 | A, B | F | 125 |
| 5 | 29 | B, C | F | 125 |
| 6 | 15 | A, D | F | 133.3 |
| 7 | 40 | A, B, C, D | M | 63.9 |
| 8 | 45 | A | F | 113.9 |
| 9 | 39 | C | M | 44.4 |
| 10 | 21 | B | F | 113.9 |

Table 4.10: Search space and scores for Person 8

|  | Age | Hobbies | Gender | Score |
| :--- | :--- | :--- | :--- | :--- |
| Weights | 50 | 25 | 75 |  |
| Person 8 | 45 (desired) | A (desired) | F |  |
| 1 | 25 | A, B | M | 133.3 |
| 2 | 30 | C, D | M | 113.9 |
| 3 | 33 | A, B | F | 63.9 |
| 4 | 23 | D | M | 108.3 |
| 5 | 29 | B, C | F | 38.9 |
| 6 | 15 | A, D | F | 47.9 |
| 7 | 40 | A, B, C, D | M | 150 |
| 9 | 39 | C | M | 119.4 |
| 10 | 21 | B | F | 27.8 |



Figure 4.14: Search space for Person 1


Figure 4.15: Search space for Person 4


Figure 4.16: Search space for Person 8
We have developed a prototype using the RAIMA framework and tested it in two distinct social settings. The first setting is a sailing club in which club members can find others based on three attributes. The weights for each attribute are provided by the context generator and can be changed by the user if needed. Table 4.11 represents a scenario where Person 1 is calculating his distance with respect to 9 other users in a sailing club. As can be seen in the table, Person 1 is looking for someone in the age range of 25 who is interested in Sailing and Fishing and who is also a male. This is shown in Table 4.11.

Table 4.11: Usage of RAIMA in a sailing club simulation

|  | Age | Hobbies | Gender | Score |
| :--- | :--- | :--- | :--- | :--- |
| Weights | 50 | 25 | 75 |  |
| Person 1 <br> Criteria | 25 (desired) | Sailing, Fishing <br> (desired) | M |  |
| 2 | 30 | Skiing, Cruising | M | 44.4 |
| 3 | 33 | Sailing, Fishing | F | 144.4 |
| 4 | 23 | Salsa | M | 50 |
| 5 | 29 | Fishing, Skiing | F | 131.9 |
| 6 | 15 | Sailing, Cruising | F | 126.4 |
| 7 | 40 | Sailing, Fishing, <br> Skiing, Cruising | M | 58.3 |
| 8 | 45 | Sailing | F | 120.8 |
| 9 | 39 | C | M | 38.9 |
| 10 | 21 | Fishing | F | 131.9 |

Table 4.11 shows the conceptual scores calculated with respect to person 1 against the rest of the club members. It can be inferred from the table that person 1 has a lot in common with person 3 , person 10 , and person 8 , and fewer similarities with persons 9 and 4.

Figure 4.17 shows the output from the visualization engine in this scenario. The user of the system is always the focus node and is represented in the middle of the graph. The distances from the user represent the degree of similarity. The nodes are clickable, and relevant information about each user is shown when the user clicks on a node.


Figure 4.17: RAIMA Visualization of the sailing club simulation

The other scenario represents the same social structure in an academic conference context, in which person 4 is looking for a researcher with a research interest in Artificial Intelligence with the age range of 30 and preferably a Male.

The research interests Artificial Intelligence, Communications, and Control are being abbreviated to AI, Comm, and Ctrl according to the table's space restrictions. It is shown that Person 4 has a lot in common with researchers 5 and 6 and less in common with
researchers 1 and 9. Scores for the conference scenario are shown in Table 4.12. Figure 4.18 represents how this information is presented to the user during the conference.

|  | Age | Research | Gender | Score |
| :--- | :--- | :--- | :--- | :--- |
| Weights | 50 | 25 | 75 |  |
| Person 4 <br> Criteria | 30 (desired) | AI (desired) | M |  |
| 1 | 25 | Comm, Ctrl | M | 44.4 |
| 2 | 30 | Comm, AI | M | 75 |
| 3 | 33 | Comm, Ctrl | F | 125 |
| 5 | 29 | Ctrl, Comm | F | 125 |
| 6 | 15 | Ctrl, AI | F | 133.3 |
| 7 | 40 | Comm, Ctrl, <br> Theory, AI | M | 63.9 |
| 8 | 45 | Theory | F | 113.9 |
| 9 | 39 | Comm | M | 44.4 |
| 10 | 21 | Ctrl | F | 113.9 |

Table 4.12: Usage of RAIMA in an academic conference

Figure 4.16 shows the egocentric graph associated with the academic conference scenario. The conceptual scores are shown on the vertices and can be translated into more meaningful strings such as "High Score" for a short conceptual distance and "Low Score" for a large conceptual distance.


Figure 4.18: RAIMA Visualization of the academic conference simulation

## Chapter Five: CONCLUSIONS

### 5.1 Summary

In our research, we have proposed a software development framework for the design and development of adaptive mobile social networks. The framework includes a social decision support model allowing users to make optimal decisions in a short time without relying on controlled communications infrastructure. This part of our work is also demonstrated in [70] and [71]. A design principle in our work is to introduce a development framework that does not replace interpersonal interaction but enhances it by providing users with real-time information about proximate social nodes. Following this guideline, we have proposed a framework that includes a matching algorithm called RAIMA, a pairing scheme, a customizable context generator, and a visualization engine. The complete framework is also described in [72].

We have provided grounds for the use of a linear algorithm in a ubiquitous computing environment in comparison to techniques such as Fuzzy Logic. We have demonstrated the superior performance of RAIMA against the SLA matching algorithm, which is used extensively in Social Networks. We have performed our tests using a medium-sized social network with three different sets of attributes. In some our research works such as [73], [74] and [75] we have demonstrated how RAIMA is used to build sophisticated social software in public transit environments.

We have introduced a pairing and discovery mechanism that is suitable for use in volatile ad hoc social networks. We have explained and demonstrated that discovery mechanisms proposed by standardized protocols such as Bluetooth, IEEE 802.16, and UPnP are narrowing usability and practicality in ad hoc social networks. Our proposal suggests that using a System-defined Hash Distribution scheme can significantly improve the usability of real-time social networks. We explained why run-time adaptation and reconfiguration of the social software is important in dynamic and mobile social networks. Current practice in the development of social systems incorporates social interaction logic in the design principles of these systems. We have provided an apparatus and a method to separate the matchmaking paradigms from the social context domain. This enables the social software to dynamically adapt itself to the context domain. For example, one software application can be used in a multitude of social semantics without the need for the software to be recompiled, restarted, or modified. We believe this is an important contribution of our work, as it addresses an important challenge in social software as it has been raised by the literature. The literature signifies the need of social profiles to be portable within different social settings, and the interaction between these profiles adjusts accordingly when users roam between social settings. We have also provided a new representation model for visualizing real-time social networks. We have argued and explained that despite current practice, social graphs should be represented in egocentric graph forms rather than socio-centric. We have adopted this representation model based on the recommendations of social sciences results and discussed how this model is more suitable in real-time social networks.

Figure 5.1 summarizes the overall interaction and processes between components and modules in the RAIMA framework.


Figure 5.1: Interaction and Processes within the RAIMA framework

### 5.2 Contributions of the Research

1. This thesis introduces a complete development framework for the design and analysis of opportunistic social networks. Unlike well-known frameworks such as Social Serendipity [76] and Mobiclique [77] that are designed on top of existing protocols and algorithms, RAIMA introduces it's own algorithm and discovery model to address the system design guidelines derived from social sciences. A major advantage of RAIMA in comparison with existing frameworks is it's recursive multi-criteria matching algorithm that is proven to be more efficient than Stable Matching Algorithms in opportunistic networks. Both Social Serendipity and Mobclique use SMA as their matching algorithms.
2. In accordance with our literature review, RAIMA is the only available framework that addresses the following social networking concerns in one modular framework. These concerns as described in chapter 2 are as follows: mechanisms to enhance profile portability, service discovery, privacy and information abstraction, reconfigurability and multi-criteria selection.
3. Providing run-time adaptation in social networks without relying on code modification is an important contribution of RAIMA. This is achieved by proposing an ontology model for the modeling, synthesis, and distribution of complex social context information. The provisioned ontology is also "pluggable" in other social networking systems, this increases the scalability of the software applications that are designed based on RAIMA.
4. Introduced framework has the ability to process missing information when a user's complete social profile is not available. Since RAIMA is a relative matchmaking algorithm, missing social data can affect only the conceptual distances according to the associated weights. This means that the context generators can increase or decrease the weight of social attributes according to the context domains. For example, in each domain the highly-weighted attributes can be set as mandatory where in another context domain the same attribute can be identified as optional.
5. The hierarchical design structure of the framework increases the scalability and integration of social software based on the RAIMA framework. Figure 5.2 shows how the modularity and hierarchical design of RAIMA addresses the requirements of different computational platforms.

## RAIMA

LAYERS


Figure 5.2: Structure of RAIMA's modular design

### 5.3 Suggestions for Future Research

One of the limitations of the proposed system is its inability to effectively encounter network latency, if the number of nodes is more than 50 . We are working to enhance the scheduling algorithm to overcome this issue for large-scale deployments. To enhance the matching algorithm, the work of Sutterer et al. [78] and Ye et al. [79] is followed, and
plans are in place to replace the linear scoring model with a Bayesian model that enables processing and inference from incomplete social profiles. However, due to the processing power requirements for this scoring model, the architecture of the framework will change, and comparisons of the two frameworks can be the subject of a future research. The framework can become more intelligent if it can become learnable and adapt attributes based on previous social encounters. To implement this feature, the framework should become capable of handling feedback from users. This could be autonomous feedback or a manual tagging process performed by the user. The challenge will be how to use this tagged information as input for a learnable expert system. Works by Amini et al. [80] and [81] illustrate how information tagging is used in social networking systems. Filtering of profiles according to the social context would be another valuable feature to add to this framework. As described in our social sciences results, people may avoid certain social profiles depending on the social context. Collaborative filtering techniques such as the one used in [82] and [83] are interesting models for this feature.

## REFERENCES:

[1] S. Greenspan, D. Goldberg, D. Weimer, and A. Basso, "Interpersonal trust and common ground in electronically mediated communication," in Proc. of the 2000 ACM Conference on Computer Supported Cooperative Work, New York, NY, pp. 251-260, Apr. 2000.
[2] B. Kolko, E. Rose and E. Johnson, "Communication as information seeking: The case for mobile social software for developing regions," in Proc. of the 16th International Conference on World Wide Web, Banff, AB, pp. 863-872, May 2007.
[3] L. Tokarchuk, K. Shoop, and A. Ma, "Using co-presence communities to enhance social recommendation," in Proc. of the Sixth International Conference on Wireless on-Demand Network Systems and Services, Snowbird, UT, pp. 157-160, Feb. 2009.
[4] D. Lowet, and D. Goergen, "Co-browsing dynamic web pages," in Proc. of the 18th International Conference on World Wide Web, Madrid, Spain, pp. 941-950, Apr. 2009.
[5] D. Cosley, P. Ludford, and L. Terveen, "Studying the effect of similarity in online task-focused interactions," in Proc. of the 2003 International ACM SIGGROUP Conference on Supporting Group Work, Sanibel Island, FL, pp. 321-329, Nov. 2003.
[6] D. Yamamoto, I. Takumi, and H. Matsuo, "Location-based social network services employing student cards for university," in Proc. of the 2009 International Workshop on Location Based Social Networks, Seattle, WA, pp. 2124, Nov. 2009.
[7] J. Thom-Santelli and D. Millen, "Learning by seeing: photo viewing in the workplace," In Proc. of the 27th International Conference on Human Factors in Computing Systems, Boston, MA, pp. 2081-2090, Apr. 2009.
[8] K. Lang, "Newsweeder: Learning to filternetnews," In Proc. of the 12th International Conference on Machine Learning, Tahoe City, CA, pp. 331-339, July. 1995.
[9] W. Sack, "Conversation map: a content-based Usenet newsgroup browser," in Proc. of the 5th International Conference on Intelligent User Interfaces, New Orleans, LA, pp. 233-240, Jan. 2000.
[10] P. Resnick, N. Iacovou, M. Suchak, P. Bergstrom and J. Riedl, "GroupLens: an open architecture for collaborative filtering of netnews," in Proc. of the 1994 ACM Conference on Computer Supported Cooperative Work, Chapel Hill, NC, pp. 175-186, Oct. 1994.
[11] J. Bitsch Link, N. Viol, A. Goliath and K. Wehrle, "SimBetAge: Utilizing temporal changes in social networks for pocket switched networks," in Proc. of the 1st ACM Workshop on User-Provided Networking: Challenges and Opportunities, Rome, Italy, pp. 13-18, Dec. 2009.
[12] T. Nakajima, K. Fujinami, E. Tokunaga and H. Ishikawa, "Middleware design issues for ubiquitous computing," in Proc. of the 3rd International Conference on Mobile and Ubiquitous Multimedia, College Park, MD, pp. 55-62, Oct. 2004.
R. Ling and B. Yttri, "Nobody sits at home and waits for the telephone to ring": Micro and hyper-coordination through the use of the mobile telephone, Telenor Report, 1999.
[17] P. Resnick, N. Iacovou, M. Suchak, P. Bergstrom and J. Riedl, "GroupLens: An open architecture for collaborative filtering of netnews," in Proc. of the 1994 ACM Conference on Computer Supported Cooperative Work, Chapel Hill, NC, pp. 175-186, Oct. 1994.
J. Konstan, B. Miller, D. Maltz, J. Herlocker, L. Gordon, and J. Riedl, "GroupLens: Applying collaborative filtering to Usenet news," Communications of the ACM, vol. 40, no. 3, pp. 77-87, 1997.
[19] L. Iaquinta, M. de Gemmis, P. Lops, G. Semeraro, M. Filannino and P. Molino, "Introducing Serendipity in a Content-Based Recommender Systems", in Proc. of the 2008 Eighth International Conference on Hybrid Intelligent Systems, pp.168173, Sept. 2008
M. Ackerman, "Augmenting the organizational memory: a field study of answer garden," in Proc. of the 1994 ACM Conference on Computer Supported Cooperative Work, Chapel Hill, NC, pp. 243-252, Oct. 1994.
[25] L. Foner, "Yenta: a multi-agent, referral-based matchmaking system," in Proc. of the First International Conference on Autonomous Agents, Marina del Rey, CA, pp. 301-307, Feb. 1997.
[26] J. DiMicco, K. Hollenbach, A. Pandolfo and W. Bender, "The impact of increased awareness while face-to-face," Human-Computer Interaction, vol. 22, no. 1, pp. 47-96, 2007.
R. Fish, R. Kraut, R. Root, and R. Rice, "Evaluating video as a technology for informal communication," in Proc. of the SIGCHI Conference on Human Factors in Computing Systems, Monterey, CA, pp. 37-48, May 1992. of online design discussions in an Open Source Software community," Interactive Computing, vol. 20, no. 1, pp. 141-165, 2008.
[31] M. Smith and A. Fiore, "Visualization components for persistent conversations," in Proc. of the SIGCHI Conference on Human Factors in Computing Systems, Seattle, WA, pp. 136-143, Mar. 2001.
[32] W. Waller, "The rating and dating complex," American Sociology Review II, pp. 284-292, Oct. 1973.
C. Fischer, "To Dwell Among Friends: Personal Networks in Town and City", Chicago: The University of Chicago Press, pp. 41-58, 1982.
[34] D. Cosley, J. Baxter, S. Lee, B. Alson, S. Nomura, P. Adams, C. Sarabu, and G. Gay, "A tag in the hand: supporting semantic, social, and spatial navigation in museums," in Proc. of the 27th International Conference on Human Factors in Computing Systems, Boston, MA, pp. 1953-1962, Apr. 2009.
W. Whyte, The Organization Man, New York: Simon and Schuster, 1956. R. Kraut, "Applying social psychological theory to the problems of group work," in HCI Models, Theories and Frameworks: Toward A Multidisciplinary Science, John M. Carrol, San Francisco: Morgan Kaufmann Publishers, 2003, pp. 325356.
[37] J. Walther, "Interpersonal effects in computer-mediated interaction: A relational perspective," Communication Research, vol. 19, no. 1, pp. 52-90, 1992.
E. Goffman, The Presentation of Self in Everyday Life, New York: AnchorDoubleday Books, 1961. Press, 1977, pp. 59-78.
[40] J. Gabarro, "The development of working relationships," in Intellectual Teamwork: Social and Technological Foundations of Cooperative Work, eds. Galegher, J., Kraut, R. and Egido, C., Hillsdale, NJ: Lawrence Erlbaum, 1990, pp. 79-110.
[41] L. Freeman, "Computer programs in social network analysis," Connections, vol. 11, no. 2, pp. 26-31, 1988.
[42] M. Granovetter, "The strength of weak ties," American Journal of Sociology, vol. 78, no. 6, pp. 1360-1380, 1973.
T. Allen, Managing the Flow of Technology, Cambridge, MA: The MIT Press, 1977.
[46] S. Karau and K. Williams, "Social loafing: A meta-analytic review and theoretical integration," Journal of Personality and Social Psychology, vol. 65, no. 4, pp. 681-706, 1993.
[49] P. Kollock, "Design Principles for Online Communities: Lessons from Early Settlements," in Proc. of the 1997 Virtual Reality Annual International Symposium, IEEE Computer Society, p. 152, Washington, DC, Mar. 1997.
[50] P. Resnick, K. Kuwabara, R. Zeckhauser, and E. Friedman, "Reputation systems," Communications of the ACM, vol. 43, no. 12, pp. 45-48, 2000.
[51] M. Ackerman, L. Cranor and J. Reagle, "Privacy in e-commerce: examining user scenarios and privacy preferences," in Proc. of the lst ACM Conference on Electronic Commerce, Denver, CO, pp. 1-8, Nov. 1999.
R. Verbrugge and B. Dunin-Kęplicz, "Formal approaches to multi-agent systems," Autonomous Agents and Multi-Agent Systems, vol. 19, no. 1, pp. 1-3, 2009.
B. Nardi, S. Whittaker, E. Isaacs, M. Creech, J. Johnson and J. Hainsworth, "Integrating communication and information through ContactMap," Communications of the ACM, vol. 45, no. 4, pp. 89-95, 2002.
S. Whittaker, Q. Jones, B. Nardi, M. Creech, L. Terveen, E. Isaacs and J. Hainsworth, "ContactMap: Organizing communication in a social desktop," Computer-Human Interaction, vol. 11, no. 4, pp. 445-471, 2004.

55] M. Granovetter, "The strength of weak ties," American Journal of Sociology, pp. 1360-1380, 1973.
R. Yager, "Intelligent social network modeling and analysis, " in proc of the $3^{\text {rd }}$ International Conference on Intelligent System and Knowledge Engineering,, , vol.1, no., pp.5-6, 17-19 Nov. 2008.
K. Stanoevska-Slabeva, T. Wozniak, I. Hoffend and J. Ebermann, "Towards a concept for inclusion of social network information as context information," in proc of the International Conference on Ultra Modern Telecommunications \& Workshops, pp.1-5, Oct. 2009.
[58] W. Reinhard, J. Schweitzer, G. Volksen and M. Weber, "CSCW tools: concepts and architectures," IEEE Computer Journal, vol.27, no.5, pp.28-36, May 1994. large-scale study in the orkut social network," in Proc. of the Eleventh ACM SIGKDD International Conference on Knowledge Discovery in Data Mining, Chicago, IL, pp. 678-684, Aug. 2005.
[62] J. Heer and D. Boyd, "Vizster: Visualizing Online Social Networks," in Proc. of the 2005 IEEE Symposium on Information Visualization, pp.32-39, Oct. 2005.
[64] N. Henry, J.D. Fekete and M.J. McGuffin, "NodeTrix: a Hybrid Visualization of Social Networks," IEEE Transactions on Visualization and Computer Graphics, vol.13, no.6, pp.1302-1309, Dec. 2007
[65] W. Sack, F. Détienne, N. Ducheneaut, J. Burkhardt, J. Mahendran and F. Barcellini, "A Methodological Framework for Socio-Cognitive Analyses of Collaborative Design of Open Source Software," Computer-Supported Cooperative Work, vol. 15, pp. 229-250, Jun. 2006.
[66] M. Smith, and A. Fiore, "Visualization components for persistent conversations," in Proc. of the SIGCHI Conference on Human Factors in Computing Systems, Seattle, WA, pp. 136-143, Mar. 2001.
D. Charlet, V. Issarny, and R. Chibout, "Service discovery in multi-radio networks: an assessment of existing protocols," in Proc. of the 9th ACM International Symposium on Modeling Analysis and Simulation of Wireless and Mobile Systems, Torremolinos, Spain, pp. 229-238, Oct. 2006.
[68] A. Peters and A. Heuer, "Time Efficient Service Discovery in Mobile P2P Architectures Using Bluetooth," in Proc. of the 2008 19th International Conference on Database and Expert Systems Application, pp. 347-351, Sep. 2008.
[69] J. Lyons, J. Ritter, K. Thomas, L. Militello and P. Vincent, "Collaborative Logistics: Developing a Framework to Evaluate Socio-Technical Issues in Logistic-Based Networks," in Proc. of the International Symposium on Collaborative Technologies and Systems, Washington, DC, pp. 208-214, May 2006.
[70] H. Rahnama, A. Sadeghian and A. Madni, "Social Context Awareness in Ad Hoc System of Systems", in Proc. Of the IEEE 2007 International Conference of System of Systems, San Antonio, TX, April 23-25, 2007.
[71] H. Rahnama, A. Sadeghian and A. Madni, "Adaptive Social Contexts in Ad Hoc Mobile Networks", in Proc. Of the IEEE 2008 International Symposium on Communications, Control and Signal Processing, St. Julian, Malta, March 12-14, 2008.
[72] H. Rahnama, A. Sadeghian and A. Madni, "Relational Attribute Integrated Matching Algorithm (RAIMA) for the Design of Spontaneous Computer Mediated Social Networks," IEEE Systems Journal, Accepted, 2010.
[73] H. Rahnama, X. Aubry and G. Isabelli, "Adaptive Context-Aware Mobile Services - Solutions to help disabled people in public transport systems" in Proc. of the 2008 Smart Mobility Conference, Sophia Antipolis, France, Sept. 17-19, 2008.
[75] Invited Paper: H. Rahnama, A. Sadeghian, A. Madni, "Self-Adaptive Middleware for Building Context-Aware Software for Public Transit Systems," in Proc. 2010 International Symposium on Communications, Control and Signal Processing, Limassol, Cyprus, March 3-5, 2010.
[76] N. Eagle, A. Pentland, "Social serendipity: mobilizing social software," IEEE Pervasive Computing, vol.4, no.2, pp. 28- 34, March 2005 IEEE INFOCOM Workshops 2009, pp.1-2, April 2009
[78] M. Sutterer, O. Droegehorn and K. David, "User Profile Selection by Means of Ontology Reasoning," in Proc. of the 2008 Fourth Advanced International Conference on Telecommunications, Washington, DC, pp. 299-304, Jun. 2008.
M. Amini, A. Tombros, N. Usunier, M. Lalmas and P. Gallinari, "Learning to summarise XML documents using content and structure," in Proc. of the 14th ACM International Conference on information and Knowledge Management, Bremen, Germany, pp. 297-298, Oct. 2005.
R. Kildare, R. Williams and J. Hartnett, "An online tool for learning collaboration and learning while collaborating," in Proc. of the 8th Austalian Conference on Computing Education, Hobart, Australia, pp. 101-108, Jan. 2006.
H. Ma, I. King and M. Lyu, "Effective missing data prediction for collaborative filtering," in Proc. of the 30th Annual International ACM SIGIR Conference on Research and Development in information Retrieval, Amsterdam, The Netherlands, pp. 39-46, Jul. 2007.
J. Wang, A. de Vries and M. Reinders, "Unified relevance models for rating prediction in collaborative filtering," ACM Transactions on Information Systems, vol. 26, no. 3, pp. 1-42, 2008.

## APPENDIX A. RAIMA PROTOTYPE

## Appendix A. How RAIMA is used?

This appendix shows how the RAIMA framework is used. We demonstrate how RAIMA differentiates between context providers (RAIMA Hosts) and context clients (RAIMA Users) and how scoring models are associated with context templates.

## I) Registration and Signup:

Figure A1 shows the registration interface. The user can login as a client user or the host.


Figure A1. Registration interface on the RAIMA Framework

## II) Host Services:

A host user can create a zone and manage users in available context templates. Figure A2 shows the functionalities that are available to a host user.


Figure A2. Functionalities available to a host user

A host user (Context Provider) can create a new zone over a geographical area and create a template for the zone. The mapping interface is shown in Figure A3.


Figure A3. Zone selection interface in RAIMA

Each zone can then be associated with an existing context template, or the user can create completely new semantics for it using an information elicitation model shown in Figure A4.


Figure A4. RAIMA Interface for Context Generation

The templates are all stored in a context repository and can be accessed, archived, and edited as needed. Figure A4 shows a screen shot from the RAIMA context repository.


Figure A5. RAIMA Context Repository

## III) User Services:

A RAIMA client can select a context template to create a user profile and preferences. This is shown in Figure A6. In chapter 3, we demonstrated how we used information elicitation to create a scoring model for matchmaking within a context template. The user interface of this model is shown in Figure A7.

Please select a zone.


Next

Figure A6. Zone selection by a RAIMA user


Figure A7. Profile/Criteria generation using information elicitation

## IV)RAIMA client: representation and visualization:

After the user provides his or her profile attributes and preferences, he or she can use the RAIMA client to access the template. The user can set the threshold for the precision and recall of the network as well. The control panel in Figure A8 shows how a user can select a zone and define its matching thresholds.


Figure A8. RAIMA Client Control Panel
Depending on the matching thresholds, the RAIMA client recalls the matches and represents them in socio-centric form. This is shown in Figure A9. Figure A10 shows the
interaction interface when a node is selected. The user can send a message to other users or bookmark their profiles for future interactions.


Figure A9. RAIMA visualization interface


Figure A10. RAIMA Interaction interface when selecting a user


[^0]:    A Thesis
    Presented to Ryerson University
    In partial fulfillment of the requirements for the degree of Doctor of Philosophy In the program of Electrical and Computer Engineering

    Toronto, Ontario, Canada, 2010
    © Hossein Rahnama 2010

