Master’s Thesis Nr. 101
Systems Group, Department of Computer Science, ETH Zurich

in collaboration with
Credit Suisse

Enterprise Social Networks Analysis

by
Denitsa Dobreva

Supervised by
Flurin Capaul (Credit Suisse),
Prof. Gustavo Alonso (ETH)

April 2013 – October 2013
Abstract

If you take meetings, instant messaging, emails and calls out of your working day there is a fair chance that the remaining work-related communication is relatively small. Most of people’s daily lives is interacting with the social network at the office, engaging with people and using relationships to drive innovation and results. Most individuals would attest to having a good understanding of their network, knowing their key counterparts and using their connections wisely. However, people can not verify if that is correct. In this work, we present a novel way of aggregating employees social interactions at work and building their social graph. Key contributions include the development of an importance algorithm used throughout the application to access peers business relation and filter the best ones, determine communities of practice and find critical hotspots which could improve performance and collaboration. The interactions across the hierarchy are modeled to reveal insights into divisional-based communication patterns. Different methods are discussed how to get the best referral to a new business contact inside the company as well as outside it, across the organizational boundaries.
# Contents

1 Introduction ........................................... 1
   1.1 Motivation ........................................ 1
   1.2 Problem Statement ................................... 2
   1.3 Objectives .......................................... 3
   1.4 Thesis Outline ...................................... 3

2 Background ........................................... 5
   2.1 Social Networks ..................................... 5
   2.2 Social Network Metrics .............................. 5
      2.2.1 Centrality ..................................... 5
      2.2.2 Community Detection ............................ 8
   2.3 Related Work ........................................ 9
      2.3.1 Locating People with Common Interests ........ 9
      2.3.2 Semantic-based Relationships .................. 10
      2.3.3 Email Mining .................................. 10
      2.3.4 Referrals ..................................... 10

3 Approach .............................................. 12
   3.1 Use-Case Domains ................................... 12
   3.2 Input Data ........................................ 14
   3.3 Relationship Evaluation ............................. 14
   3.4 High-level Categories ............................... 15
   3.5 Types of Networks .................................. 16
   3.6 Individual Roles ................................... 18
   3.7 Who knows who ..................................... 19

4 Design ................................................. 21
   4.1 Platform ........................................... 21
   4.2 Application Modules ................................ 21
   4.3 Architecture ....................................... 22

5 Implementation ........................................ 25
   5.1 Project Structure ................................... 25
   5.2 Data Collection and Transformation ................. 26
      5.2.1 Outlook Access ................................ 26
      5.2.2 Data Retrieval ................................ 26
      5.2.3 Challenges .................................... 27
Chapter 1

Introduction

In real life and online people spent a tremendous amount of time interacting with others and building intentionally or unintentionally their social networks. Whether this is the communication on an online social platform, engaging with people on professionally oriented portals, sending emails or having dinner with friends and family, all these activities have a common ground: they strengthen individuals relationship to other people. At the office people collaborate with business-related contacts by sharing professional knowledge and working together on projects. Individuals strive to manage their time in the best way to keep their relations active. However, at work it often happens that people change projects and team members, new employees are hired or leave the company. Hence, the business network is changing very dynamically. This makes it very difficult to keep track of one’s relations status and maintain overall strong relations. It is necessary to identify the key partners and keep these close.

1.1 Motivation

On a daily basis people interact with friends and colleagues and generate a high amount of social data. At work, they write emails, send instant messages, arrange meetings and make calls. The data from all these logs is vast and represents an endless number of connections between people. These archives keep all business partners, daily schedules, frequency of interaction as well as number of messages ever written or received and to whom. People interact and receive messages from a variety of different contacts many of which they do not even know on a first name basis. If employees had an overview of their social graph and knew how strong separate relations are, they could utilize these in case where some information is required, but they do not know whom to approach. Knowing the personal social partners better could improve the productivity and performance. In addition, in organizations the question of who has already a good relation to a person of interest comes up very often. For example, if someone wants to get introduced to somebody else he would ask a friend who has a strong connection to this person. In order to solve these problems, a method is required that evaluates people’s relationships.

Employees interact with many people and try to manage their connections wisely. The lack of a mechanism to help one track his interactions and rate the strength of the relationships makes it nearly impossible to achieve. A mechanism is needed to bring to light forgotten relations or
neglected parties. Therefore, we want to design and develop an application that helps the user to assess his time distribution towards all communication partners.

In addition, employees need a way to explore the people and connections in the organization. This can by far increase innovation as the more people one speaks to, the higher the possibility he could learn something new that could potentially lead to new ideas. It is said that collaboration is the way to innovation [1] but one should know his key relationships in the organization to be able to utilize them. The fundamental idea of teams in companies is to share ideas, support each other and thus, develop projects quicker and better. Ideas are not being born in isolation. They are the result of people sharing their know-how. To reveal new possible valuable partners employees need a way to explore these.

1.2 Problem Statement

Over the years employees loose the overview of their complete social graph. The instincts often mislead individuals into believing that they keep a good relation with all their strong former contacts. However, the verification of this belief could help identify neglected relationships. The communication tools that employees use have many advantages like being fast and asynchronous. This makes people use them very extensively. Employees receive many messages from various people and loose track over their important contacts and the invested time with these. Loss of awareness to the existing people in medium-to-big sized companies is a known issue. However, there is no solution so far.

In order to identify collaboration problems, one needs a method to look from a higher-level perspective onto the team’s informal communication structure and observe how peers collaborate. The identification of bottlenecks or missing connections between divisions is in some cases the only way to improve results. When projects fail dramatically project managers start to search for a possible reason. This is usually right there. It is obvious but not easy to realize. Communication activities are central to teams and thus, to the organization. However, with the growing number of members it becomes hard to coordinate collaboration by means of standard tools.

Online social platforms like Facebook, LinkedIn and others introduce the concept of a social network in which users are connected to a group of other users. It is not known what the relationship rate of these connections is. Further, it is hidden whether people really know each other and if so how does the relation strength change over time. In order to identify communication disconnections, we need to realistically assess groups’ interactions and peers’ communication intensity. The listed platforms can present a list of all social contacts of a given person. However, they can not examine the real email and other electronic communication because this is property of the organization and thus, it is private.

To enable the evaluation of people relationships this thesis aims to analyze the actual exchanged communication items between employees as well as employees and their business contacts outside the company. Four problem domains are identified. These are the Personal Productivity and Learning, Collaboration and Innovation, Organizational Structure and Shared Knowledge and Connections.
1.3 Objectives

This thesis is motivated by the current situation in organizations and the problems described in the previous section. The set of objectives is built around the goal to make people aware of their working environment. We want to make it easier to find new people in the social network, their social communities and the network properties.

The main goal of the work is to implement a tool which offers a graphical display to explore relationships between people and visualize social groups of interaction. The most significant peers of the user and key individual roles are highlighted in the social graph. The objective is to make the users aware of how they distribute their time among their communication partners and show relationships that have deteriorated over time. This is a must to better utilize your time and get more out of your working day. We aim to offer a method to explore the structure of informal communities formed by social interactions. Having this insight the user can reveal new groups of interest and engage in different activities with new people. Social interactions are essential to every individual and as important as hard work.

Further, the goal is to construct a subset of the global network of an organization based on multiple user networks. The target is to research and develop a method that analyzes the structure of the aggregated network and highlights people with important informal roles as e.g. people that are the glue of the graph by connecting otherwise distant peers or groups. A method is needed to highlight disconnected organizational units and weak ties in the formal structure. In addition, we aim to study different mechanisms of how to get the best referral to a new contact of interest.

Our objective is to discover the application domains of such tool and the possible use cases inside Credit Suisse where the research takes place. We want to utilize the existing social data as proof of concept of the useful and practical applications. The thesis focuses on people’s relationships that can be retrieved by wisely aggregating over one’s communication archive. An innovative way of mining the electronic data is introduced as part of a prototype developed during the study.

1.4 Thesis Outline

In this chapter we presented the current tendencies around extensive email communication in organizations and the need for a social network analysis tool. A discussion was given based on the thesis ideas and goals. The rest of this thesis is organized as follows:

Chapter 2 presents the origins and evolution in social networks. It gives a background on popular social analysis metrics and examples of their practical use. In the second part of Chapter 2 several examples of related studies on email mining and social analysis are discussed as well as their downsides when applied to our specific case.

In Chapter 3 it is presented the approach to the problem. The thesis guidelines are discussed together with a short description of the proposed solution. The idea on determining who is important in a network is described together with the use cases and domains of the implemented prototype with several real-world examples.

Chapter 4 is an overview of the architecture. A discussion is presented on the building blocks of the prototype as well as the modules of the implementation. We describe the integrated
frameworks and what is the reasoning for their selection.

**Chapter 5** is a detailed presentation of the implementation. The *Importance* function which evaluates people’s relationships is described together with additional algorithms like finding the best chain of referrals. Implementation details and used design patterns are discussed. The complete data mining is presented with details on the aggregation process.

**Chapter 6** is an evaluation and overview of the results. We show examples of different network views and give an analysis on insights from the prototype.

Finally, in **Chapter 7** we make conclusions on the work in this thesis and present some ideas and directions for future work.
Chapter 2

Background

2.1 Social Networks

Social networks is an area of study that has attracted the curiosity of researchers already in the 20th century. It is an interdisciplinary field studied and applied in various domains like e.g. sociology, anthropology, psychology, biology and many others [2]. One famous example in SNA is a study conducted by the sociologist Stanley Milgram in the 1960s. This is known as the 'Small-world' phenomenon [3]. With series of experiments, he proved that the average path length between two random people in the United States contains at most 6 hops. Milgram’s results were revolutionary as they shown that people live in a high connected world. Many years later, Malcolm Gladwell disagrees [4], that the 'six-degrees-of-separation' phenomenon is driven by a limited number of very high-interconnected peers with large networks of contacts, who serve as hubs.

Our approach is based on these ideas. The existence of very highly interconnected communities of people that form informal groups of practice in the organization is studied. Driven by the idea that there are probably only few people with a high number of contacts who keep the network connected we study different social network analysis metrics that help us identify these people and other important individual roles.

2.2 Social Network Metrics

Social network analysis is used to identify the important individuals in a given social graph. For that purpose SNA defines different metrics which measure the significance of a node or its centrality in the network.

2.2.1 Centrality

Freeman (1978) proved that important nodes in a graph have a central role using different aspects of importance [5]. To confirm his suggestion he used a very simple star-liked graph with 5 nodes and 4 edges like the one in Figure 2.1. He argued that node A is most important to the network.
Figure 2.1: A star network that presents Freeman's ideas of node centrality

because of three main arguments. It has more connections to any other node compared to
the others, it can reach faster all other nodes using these connections and thus, it controls the
network and the information flow. Freeman formalized these considerations as three metrics of
node centrality: degree, closeness and betweenness centrality.

Figure 2.2: Examples of A) Degree centrality, B) Closeness centrality, C) Betweenness centrality and
D) Eigenvector centrality of the same graph.

Hue (from red=max to blue=min) shows the node centrality
(source: wikipedia, author: Claudio Rocchini)

There is a long history of research in the field of social network analysis and especially in cen-
trality. However, there are no formalized definitions on the different types of centralities. Our
study and definitions below are based on the following sources: [2, 5, 6, 13] which present clari-
fication, method of computation and further analysis on social metrics. These are applied in the
prototype to determine the important actors in the organizational graph. As there are various
aspects of being significant in a network, there are also different individual roles identified by cen-
trality. In this work we focus on the four main centrality measures because they model relevant
user roles in the organizational network. A graphical overview of these is presented in Figure 2.2.
**Degree Centrality.** It measures the number of links incident to a node. The more direct neighbors a node has, the more active it is, and the greater the chance is to convince more people of its opinion.

Degree Centrality is the most simple and intuitive centrality measure. It is applied in the thesis to find the social peers with the highest number of acquaintances.

The degree centrality is computed like:

For a graph $G := (V, E)$ with $n$ vertices, the degree centrality $C_D(v)$ for vertex $v$ is [6]:

$$C_D(v) = \frac{\text{deg}(v)}{n - 1} \quad (2.1)$$

**Betweenness Centrality.** It denotes the number of shortest paths between any two nodes containing the node in question. The more shortest paths run through a node, the more important the node is to the network. High Betweenness Centrality (BC) indicates a role of a broker or a gatekeeper. This centrality measure models the consideration that a node which is on the paths of many interactions taking place in the network has control over the communication in the graph.

In the network of employees analyzed in this thesis, the BC serves to determine the connectors between separate social groups. These connectors or brokers can be seen as unique ties between these groups but also as a bottleneck if the connection is very intensive.

To compute the betweenness centrality the following formula is applied [6]:

$$C_B(v) = \sum_{v \neq s \neq t \in V} \frac{\sigma_{s,t}(v)}{\sigma_{s,t}} \quad (2.2)$$

where $\sigma_{s,t}$ denotes the total number of unique shortest paths between $s$ and $t$, and $\sigma_{s,t}(v)$ is the number of these which contain node $v$.

**Closeness Centrality.** It is defined as the reciprocal of the sum of the geodesic distances from a given node to all other vertices in the network. It measures how close a node is to all other nodes in the graph. The smaller the distance to all other actors is, the more significant the node is in the network.

This metric is an analog to the minimax problem (minimize maximal distance). It is computed as:

$$C_C(v) = \frac{1}{\sum_{t \in V \setminus v} d_G(v, t)} \quad (2.3)$$

where $d_G(v, t)$ is the distance from node $v$ to node $t$ [6].

**Eigenvector Centrality.** It defines the importance of a node considering the importance value of its direct neighbors. Node is more important the more important neighbors it has.

This metric is very suitable for modeling networks of people. It depicts that one is as important as the number of influential contacts it has.

To compute the Eigenvector Centrality we use the adjacency matrix $A_{i,j}$ of the graph. For any nodes $i, j$ we define $A_{i,j} = 1$, when there is a connecting edge between the nodes or $A_{i,j} = 0$ if the $i$th node is not adjacent to the $j$th. $s_i$ denotes the score of the $i$th node. The score of a node
is dependent on the scores of its neighbors. In every iteration of the algorithm we update \( s_i \) as [6]:

\[
s_i = \sum_j A_{i,j}s_j
\]  

(2.4)

The algorithm finishes after \( n \) number of iterations. The value of \( n \) is defined based on our observations when the centrality results converge.

### 2.2.2 Community Detection

Very often social groups form independent of the formal organizational structure. Intensively communicating employees can be observed as non-overlapping units of interaction. People unintentionally construct informal groups based on different criteria like shared interests, professional knowledge, organizational hierarchy, culture and others. The informal communities are the groups of people that individuals spent most of their time with. These are very important to our analysis as they show collaboration patterns and the flow of information.

![Figure 2.3: A simple network partitioned into three modules indicated by the color of the nodes](image)

To reveal the social topology in the organization we use modularity in weighted undirected graphs. Modularity is described in the literature as a quality function for graph clustering [7] that aims to optimize the density of links inside the same community and lower the connections between the different communities. Hence, it identifies communities of practice based on a high number of intra-cluster connections and relative independence from the rest of the graph. In Figure 2.3 we show an example of a graph split into modules. The color of a node is an indication for its module assignment.

In weighted graphs the modularity of a partition is a scalar in the range \([-1, 1]\) and is defined as follows [8]:

\[
Q = \frac{1}{2m} \sum_{i,j} \left[ A_{i,j} - \frac{k_i k_j}{2m} \right] \delta(c_i, c_j)
\]

(2.5)

where \( A_{i,j} \) is the weight of the connection between nodes \( i \) and \( j \), \( k_i = \sum_j A_{i,j} \) is the sum of the weights of all edges incident to vertex \( i \). The value of \( c_i \) is the community assigned to vertex \( i \). \( \delta(c_i, c_j) = 1 \) where \( c_i = c_j \) and 0 otherwise. Parameter \( m \) is the sum of the weights of all edges in the graph.

Different ways have been proposed to find high-modularity partitions relatively fast. In this thesis we apply the method originally proposed by Etienne Lefebvre and known as 'The Louvain
method’ for community detection in big networks [8] or further called the Fast Unfolding algorithms. The Louvain method, developed during a master thesis at UCL, is a heuristic method based on modularity optimization. It has been proven to be very efficient in computation time and has a good accuracy in communities clustering in large networks as tested with a network of over 2 million nodes.

The Fast Unfolding algorithm is applied in the prototype to discover the social communities of practice. It runs iteratively in two phases. The first phase at random assigns communities to the nodes and then by combining communities it runs until it reaches local maxima. In turn, the second phase aggregates the nodes from the same module into a single node and thus, produces a new network. In the new graph, the weights between the nodes (communities) are the sum of the weighted connections between the nodes from the previous two communities. These two phases represent a single pass in the algorithm. It runs until it reaches a global maximum of the modularity parameter $Q$.

2.3 Related Work

This section describes related work in the field of social analysis based on electronic communication in organizations. There are different studies conducted on social networks but we leave out online communication platforms from our scope as they are not a standard tool in enterprises. The work presented below on analyzing electronic communication and identifying relationships between users is focused on similar to this thesis research topics. The authors deduce connections between peers based on email communication or web-related relevance between individuals. However, these solutions are not applicable to our use case. They focus only on email communication or else, whereas we want to examine all sources of electronic interaction in enterprises. In addition, none of the studies defines the strength of the relationship. In our application, this is important since based on the strength we model further associations.

2.3.1 Locating People with Common Interests

Schwartz and Wood (1992) discuss in [9] a study based on email communication from multiple sources. They group people based on communication patterns and determine clusters of shared interests. To do this Schwartz and Wood use a Graph Theory approach. They develop a two-staged algorithm which first groups people in clusters based on neighbors similarity. To derive better results and make the neighborhood partitions more concentrated, they filtered out people with very low connectivity stating that these are outliers. Further, they remove the intra-domain edges so that nodes like administration staff and office managers do not appear as the best contacts to all other nodes of the same domain. Looking into the inter-domain edges, they derive a relationship between two people when they are connected to the same group of people or a common subset. From the size of the symmetric difference of the sets of relations of the one and the other person over the size of the union, they deduce the value of the relationship. The second stage of the algorithm is to find the clusters. Their approach is to use an initial set of people for every topic. The discussed method has the advantage that no processing of the email content is required. However, there are the downsides that it starts with an initial set of people interested in a given topic (initial clusters) and that as the authors state it cannot deliver very accurate relationship rates. Both make it inapplicable to our scenario. First, a reliable rate of the
relations is very important to make further conclusions. Second, instead of an initial set of social groups as an input parameter we want to derive these from the communication archives.

2.3.2 Semantic-based Relationships

Another work in the field of finding relationships among people based on email communication is described in [10]. In this study, the authors conduct a semantic email-content analysis to derive relationships between people. They build a high dimensional context space from the email text which serves to identify separate networks of people and hidden relations. The study is conducted in a real-life communication network of a small-to-medium sized organization and aims to improve people awareness to their environment. They bring to light serendipitous relationships between people by mining semantic associations. On the one hand, their approach provides a set of people connected to given information without a predefined information set. They deduce the key actor connected to e.g. particular projects in the organization only from looking into the email content. On the other hand, they face very severe privacy issues by parsing private email text. Another drawback of their approach is that they have no method to filter out peers that appear to be very important to certain topics only because they have a role in the organization which requires to coordinate with many different people from different divisions. In conclusion, these shortcomings make this approach inapplicable for us.

2.3.3 Email Mining

Paper [11] presents a different approach to identify clusters of people. They apply data mining based on neural networks to identify peers that participate in common email communication threads. The algorithm initially starts by counting the number of emails exchanged between any two peers without considering the direction of the communication. Only the connections containing a satisfying number of emails exchanged are preserved. On the resulting graph, a version of decision tree technique is performed which further forms memberships of people. Finally, the authors deduce percentages of overlapping between the memberships which should further imply relations between people. Their approach is interesting and different compared to others in the field. However, it can not determine the strength of the relationship between any two members of the network which makes it unsuitable in our case.

2.3.4 Referrals

A study conducted on referrals in the Web space [12] touches common ground with a use case from our system which is how to get an introduction to a person from acquaintances. The authors describe a system that mines information about people relations available on the Internet. They do a comparison between manually finding a chain of referrals to an expert and automatically deriving a path of individuals to the target. The authors claim that while the manual approach tends to be very accurate to point to the next hop, the dynamic method is more responsive in finding the best path. This statement confirms with our assumption. We expect that it is relatively easy to determine the best referral chain in a small social network because people know their business partners and probably their connections. However, in a broad network like of a medium-to-big sized company this is cumbersome and time consuming task. In addition, the authors identify correlations which are applicable to our study. They state that the bigger the
distance is from a person to an expert, the more difficult it is to determine the best hop. Further,
they argue that the further away an individual is from the initiator, the smaller the probability
of a response. In summary, this study contains statements that help us model the best chain of
referrals. However, the study is conducted on relations based on information from the Internet
space which makes it not fully comparable to our scenario.
Chapter 3

Approach

Our primary goal is to make people aware of their social environment. To reveal employees social interaction we decide to utilize communication tools like email, chat, calls and meeting-organizing clients because these compose a substantial part of a working day in almost any organization. We meet with people in strictly organized meetings, write and receive emails on a variety of topics, send instant messages to some of our colleagues or make phone calls with business partners. The volume of the social history persisted in these archives is tremendous. Therefore, we mine these to reveal insights about the organizational network.

This project is carried out in the IT department of Credit Suisse. Therefore, the approach we take and the delivered solution are to some extent driven by the environment and the requirements of the enterprise.

To understand how the social network in this organization works and how are the individual roles distributed we start by asking ourselves three main questions that became the guideline for the study:

- Which are the use cases of an enterprise network analysis tool
- How can this be achieved
- Who is important in a network

In the following sections of this chapter, it is described the approach and the considerations to answer each of these question.

3.1 Use-Case Domains

This section presents the description of use cases applicable to organizations which are influenced from the insights shared in [1]. This book illustrates various studies conducted in enterprises which are based on social analysis. These aim to reveal organizational disconnects and further structure issues. The fundamental problems that we address can be best summarized in four domains. Some of these are very tightly related and overlap to some extent.

The domains that we distinguish are as follows:
• Personal Productivity and Learning
• Collaboration and Innovation
• Organizational Structure
• Shared Knowledge and Connections

To give a better understanding of each one together with the more abstract description, we give an example of some of the use cases implemented in this work.

**Personal Productivity and Learning**
Employees personal networks provide an input on how to increase individual’s efficiency [1]. Coworkers are the most common way of obtaining information and solving problems. Hence, who you network with has a significant impact on your effectiveness. Maintaining these relationships is of value to the individual. For example, a tool which visualizes one’s network, the strongest relations and how these change over time, can show where the user need to spend more time to strengthen relations which have begun to deteriorate.

**Collaboration and Innovation**
Collaboration is very important to organizations. It is the driving force to get work done faster and create innovation. SNA can shed light on problems of collaboration and connectivity in organizations [1]. Lack of communication across groups can make it difficult to come up with new perspectives. Knowing your teammates and cross-unit colleagues is crucial to leverage another’s expertise and achieve better results. It is very important to promote vibrant networks because success depends on effective collaboration [1]. Therefore, the application that lets one explore the friends of friends or the business partners of the direct peers and the social groups they most interact in can further boost one’s performance. This can further improve the collaboration in the organization. We claim that awareness to the environment is of high importance to the organization therefore we aim to develop an application can improve this.

**Organizational Structure**
Social network analysis can be applied to discover interaction patterns in the informal organizational structure. The analysis of this can give information on eventual information flow problems or disconnects. If an overview is available, that can visualize these one could find missing connections between divisions. This can by far improve the health of the business by recognizing these hot spots. Our model aims to present critical disconnects or very weak ties across divisions. If one could explore the rate and quality of interactions across the formal divisions in the organization, this could be the first step to improve the collaboration across units or show insights on how to reorganize them.

**Shared Knowledge and Connections**
Similar as the Word Wide Web (WWW) the organizational social network persists big amounts of information. However, the question is how to access this information. Many are good if not specialist in some field. Therefore, to solve faster business problems people often try to find a professional that could support them. As the various web sites on the WWW that are connected by links pointing to one another, employees in an organization know other people.
People are related to internal but also to external to the company people. The connectivity of people and the shared knowledge is of immense value to the organization. Therefore, we research and design a model to get the best introduction to a person through a referral chain. We call this the 'Who knows who' problem. This topic is of big interest to researchers but also to many organizations. For example, let us consider the scenario where an employee at Credit Suisse want to get introduced to Larry Page. Hence, he aims to find who are the people that could provide the highest chance of meeting him.

### 3.2 Input Data

The main ingredient of this research is the data. Therefore, it is carefully considered which information sources could at the best support the analysis on people's social patterns. Individuals use various electronic communication methods intensively while working.

To cover the complete scope of social items at Credit Suisse we consider incoming and outgoing email messages from MS Outlook, meetings stored in the Outlook Calendar application, instant messages (IM) and calls from MS Lync. The various communication types are generalized as communication items (e.g. a sent email message is single communication item). People use different types of communication items in different cases. For example, employees tend to write emails in more formal cases to contact people they have more formal relation to. On the contrary, instant messages are used within informal groups. Meetings last longer, usually do not occur that often and tend to group people working on the same project.

Outlook stores the history of all emails. Instant messages and calls from Lync are also finally persisted under Conversation History in Outlook, and meetings under Calendar. Only metadata information is utilized to show how users interact with other people.

All globally exchanged communication items are persisted on the MS Exchange Server. However, the access to these communication archives is restricted by privacy concerns. To overcome this issue we consider another approach. Instead of accessing directly the Exchange Server we design the prototype called Bunea (Business Analysis Tool), as a desktop application. It must bind to the local Outlook application and extract the metadata information of all available communication items.

### 3.3 Relationship Evaluation

People’s social interactions usually converge to a limited set of people and are sparse with the rest of the network. Individuals think they have a good understanding of their social interactions and know whom they spend most of their time with. However, often this is not the case. People believe that the ones they feel most comfortable with or share more than strictly business talks are their strongest connections in terms of social interaction. By analyzing Outlook communication we show that this expectation often fails.

Various factors which constitute and influence the importance are studied. Based on our observations on the digital communication inside Credit Suisse the following considerations are selected which define a good relation.
• Number of exchanged communication items: A high sum of sent and received items indicates an intensive communication. However, it is not always correct that a high intensity leads to a good relationship. There are scenarios where e.g. mail boxes send many emails to various individuals. This type of relation is one-directional. In our study, the definition of a good relationship is one that is strong and bi-directional. Hence, both communication parties are taking part in the interaction.

• The frequency of the communication: It is stated that the frequency is an important factor as it quantitatively shows how often the interaction occurs. The more days peers communicate, the stronger the relationship should be.

• Communication type: There are different types of messages one uses to contact a person. The type of the message is an indicator of how formal the relationship is. For example, if one has a good relation with his supervisor he would usually not only sent emails but use often instant messages. However, if one needs to contact someone that he does not know directly or another person that has a higher position in the organizational hierarchy, he would probably send an email.

• Communication balance: This factor indicates whether the communication is mutually important. Even if it is a good relationship it does not mean that the balance should be equal, hence as many items sent as received. However, a strong relation usually has similar rates of incoming and outgoing messages. This factor helps to filter out the mail addresses from which people receive many items but never respond to.

• Broadband: The more directed a social interaction is, the more important it tends to be. It is observed that messages sent to many people do not indicate a relation between sender and receiver. On the contrary, a message sent to one single peer or a very limited number of individuals is an indication that these are closely related together. A single direct message cannot define a relation as strong but many direct items contribute to a better relationship.

• Message timing: The more recent an interaction is, the more relevant it is to the present. We state that someone the user has interacted with intensively in the past is not as important anymore as someone he is interacting intensively in the present. Therefore, the timing factor is introduced to determine the time proximity of the relation.

### 3.4 High-level Categories

We differentiate between two high-level categories depending on the scope of the network and the method of extraction: *My Network* and *Our Network*.

The term *My Network* defines the local view of the organization from the perspective of the Outlook user whom we further refer to as the *Ego*. He has access to a limited partition of messages and other communication items. These are the ones sent to him, from him or where he was in the recipients list. Hence, the Ego is part of all these communication items which makes him the center of *My Network*. This network is used to define the most important peers and other individual roles with regard to the Ego. It can track his relationships over time and help improve his efficiency and performance.

*Our Network* refers to a subset of the global network. It is the result of the aggregation over multiple Ego networks. *Our Network* has the combined knowledge about more peers and the history of interaction among them. It does not have a defined center. Hence, it can be used
to research the key actors on a more global scale compared to My Network. In addition, we study the relation among the separate formal divisions in the organization and the strongest path of referrals to a distant person in the network. However, to conduct these studies we need a method to merge the available Ego networks. To achieve that, we introduce a collaborative network sharing where people volunteer their local networks. To be able to combine these into a global network without having all possible My Networks, we ensure that the value of the relation between any two nodes has the same value in both Ego networks.

In Figure 3.3 it is demonstrated an example of the two discussed categories. In both we have applied the simple importance rate algorithm described in chapter 5 and performed the Fast Unfolding algorithm to determine the social groups. There is a big difference in the number of nodes and communities between both networks because Our Network is the result of multiple Ego networks.

3.5 Types of Networks

The social network scope is split in two high-level network types depending on the method of extraction as described in the previous section. However, we model additional subtypes of networks which depend on the use cases of the graph. In the following, we introduce the different network types or also referred to as views in the context of the application.

- My Circle

My Circle is based on My Network. As the name suggests this view presents the social circle or closest community of the user. Its purpose is to show the strongest relationships in regard to the Ego. The use cases are self-evaluation of productivity and time management based on the overview of the best communication partners. My Circle shows the 30 strongest relationships of the user over a long or chosen by the user period of time. The Ego knows his job profile and based on the evaluated communication overview, he could estimate whether these are the correct people to be most dedicated to.
• Circle Over Time

This graph is complementary to My Circle. It presents the strongest communication partners but helps the user track the relationship to these over time. It can be observed if some of the previously good relations have weakened or improved, identify new people that become significant in the social network or such that leave the closest circle. To track the relation development in the circle a relation status is introduced which is compared to the previous two month of communication history.

• Direct Peers

This network is an overview of the direct contact partners. To explain the term direct we consider the following scenario. Peer A receives an email from B sent to A and C. B is a direct contact of A because he addressed A directly. On the other hand, A has never written an email to C, received from him and they never had a call or meeting. This defines C as a second level or indirect contact. In this graph we illustrate how wide the Ego’s direct network is, the rate of the separate relations and how are these people connected. A use case scenario of the Direct Peers graph is the ‘Know-how transfer problem’. By applying different masks to the network like show only internal or external to the organization contacts, only email communication partners and further, the user can get the complete picture of who are the strong communication partners of the individual in observation and other details required in a know-how transfer process.

• My Network

It presents the overall view of the Ego’s network. It includes all people visible from the user communication archive and evaluates the relationships between all peers in the network. We introduce this view as a method to observe the complete Ego network, the communities of practice from his perspective and individual roles described in the next section. One can explore the connections and dependencies extracted out of the communication items and transformed into a social network.

• Global Groups

The Global Groups network is of type Our Network. This depicts a subset of the complete global social network in a given organization. It has all the use cases of My Network but on a global scale which makes it more reliable. The insight about communities of practice and individual roles are not any more from the local user’s perspective. This is the combined knowledge and view of the network. In addition, the ‘Who knows who’ or ‘Referrals’ use case is modeled. This finds the best chain of acquaintances that can ensure a good introduction to a new person.

• Divisional Relations

The Divisional Relations graph models the collaboration across the hierarchy. To achieve this, we aggregate the inter-cluster communication between sub-divisions or organizational units. The goal is to visualize which teams have a high collaboration level and where are the hotspots of weak ties or disconnects. This network is helpful to model the need of reorganization or show how relations change before and after an event.
3.6 Individual Roles

Node importance is a main topic in Bunea. Therefore, we study relative actors’ relevance by introducing centrality metrics. In section 2.2.1 the main SNA metrics were discussed, their definition and general use cases presented. Every applied centrality metric is wrapped into an easy to understand role in the network. The roles are as follows:

- Social Hubs

  These are the actors that have the broadest network of direct peers. The Social Hubs are based on the degree centrality metric. This can be interpreted as a type popularity measure with no difference between quantity and quality [13].

  Social Hubs are important in the network as they help to keep the graph connected and decrease the number of hops to reach from one node another one. However, being connected to many is not always a good property. It depends on the context. This is in our scenario the job position of an employee. High centrality degree could be a sign that a node has become a bottleneck in the network. Such employees are overloaded by queries from their co-workers. To overcome this issue, a better task distribution could be beneficial or an additional employee with similar knowhow that could take over.

- Information Brokers

  The Information Brokers have a very important role in the social network. Similar to the Social Hubs these connect people but in a different way. Information Brokers or also called Boundary Spanners [1] communicate across subgroups of the informal network. These are, in this sense, the connection between the social communities in the network.

  According to the theory of ”The-strength-of-weak-ties”, which is first described by Granovetter in 1973 [14], innovation normally succeeds through weak ties connecting different parts of the network. As an example, if one is searching for a new job, he could hear about different job opportunities by an acquaintance, who is out of his closest communication circle. The reason is that the people that surround individuals usually have the same information. On the other hand, when one speaks to a peer from a different division, the chances are higher to hear something new. Further, combining the shared knowledge of differently-skilled people increases the likelihood to create new ideas. In [14] it is argued that to some extend the links in the network must present a heterogeneity in order for the diffusion of innovation to occur.

- Coordinators

  Closeness centrality is yet another approach to determine node importance in a graph. It is considered the measure that at the most applies to the term centrality because it captures the average distance between a vertex and every other vertex in the network [13].

  We define individuals that have a high closeness centrality as coordinators. They need the shortest time to coordinate a task or event with all others. In addition, to identify people with low closeness centrality is as important as finding the most central people. Recognizing that a person is very isolated from a group is the first step to improve the structure of the given organizational unit.
• Influential People

Influential People are determined by the eigenvector centrality. It follows the principal that connections to people with high centrality contribute more to the centrality of the node in question than connections to lower-scored nodes. For example, a person with few connections could have higher eigenvector centrality if these few are very central to the network compared to another individual that has many but not that strong connections.

The nodes with high eigenvector centrality are defined as influential people because they have influential connections. This way they have control and influence over the complete network.

3.7 Who knows who

In section 3.1 the use case 'Who knows who' was described or also referred to as 'Referrals'. In organizations the knowledge of who are the people that already have a relationship to a person in question is very significant. This can improve ones performance and efficiency by utilizing existing relations to get the best introduction to a new contact. A good chain of referrals could help get the attention and thus the willingness of the person to respond. This individual is often an expert with knowledge in a certain field whose expertise is required or a person with job position, and influence of interest. However, the problem of how to get to this person is yet not solved.

Finding the best referrals path is of big interest to enterprises. Therefore, the Irish start-up company DataHug focuses exactly on finding 'Who Knows Who' based on electronic communication similar to our data sources. For a topic of interest to an individual DataHug can determine who are the people in the organization that are experts in the field and which are the direct connectors to these. However, often the people that have a strong relation to the person of interest are not direct contacts of the requester. In this case additional insight is needed on how to approach these intermediate people.

In section 2.3.4 a study was introduced conducted on referrals in the Web. To recall, the authors derive two important conclusions. The first one, that the further the expert is, the more difficult to manually determine to best referral chain. The second one, that the bigger the distance from the requester to a person on the path, the higher the chance that the referral does not succeed.

Based on these insights we define the following criteria which are the guidelines of our solution on finding a best referral path:

1. Min hop count
2. Max edge weights
3. Balanced edge weights on the path

We claim that the likelihood of an introduction to succeed along the referral chain is higher if the path contains a lower number of hops. The reason is that the more people are involved, the higher the possibility that someone decides not to forward the request to the next hop. For example, consider the scenario where an employee from the Software Engineering division needs an introduction to a person from the Sales. If manually choosing this referral path, one would
not choose to involve more than a few people since the further the requester from these is, the lower the expectation to respond to do a flavor for the initiator.

The referral engine is designed to rate higher paths with higher edge weights. The reason is that in our system, the weight of an edge represents the strength of the relationship between the peers. The better the connection is, the higher the chance that the asked person will accept to participate in the introduction process.

Finally, the referrals engine rates better paths with a balanced edge weights. A balanced weighted path is such where the fluctuation of the weights of the edges is minimal. A path with a medium-strong ties is better than such with a strong and weak ones. The reason is that a single weak connection in the chain decreases the likelihood to proceed to the next hop. In section 5.8 we introduce the complete algorithm of dynamically finding the best referral chain.
Chapter 4

Design

This chapter describes the details around the platform of choice, building modules based on the composed requirements and application architecture.

4.1 Platform

Bunea is a Windows-based desktop application. It is designed around the idea to extract Outlook social data. Therefore, the choice of operating system (OS) is predefined. In addition, the prototype is designed specially for the requirements of Credit Suisse. Windows is the OS used on all employees’ organizational laptops. Therefore, the application is designed to be compatible with this precondition. The implementation and testing is done on machines running Windows 7 Enterprise, 64-bit.

.Net and Java are the main development platforms used in the company. To simplify the extension and integration of the application after the thesis is finished the choice of a development platform is narrowed down to these two. Various open source frameworks and libraries are compared to make the decision about the implementation platform. Java has a bigger open-source community than the .Net platform. Therefore more visualization frameworks are available in Java. To verify that Java is compatible with the project requirements various experiments are carried out to prove that using a 3rd party open-source library the access to Outlook is possible from a Java application. Finally, the prototype is compatible with Java SE 6 and higher. This is the current default Java version available on most laptops in the company.

4.2 Application Modules

Before describing the complete architecture of Bunea, an overview of the building blocks is presented. These can be summarized in four main modules as shown in Figure 4.1.

The Data module is required for the data-related activities as e.g. data access and data parsing. In order to perform any data processing an access to the data is needed. This is the responsibility of the Access module. Whether it will connect to the MS Exchange Server, read from a plain file with pre-extracted social data or connect to the local Outlook application this module must
encapsulate this information and hide it from the other modules. Further, another method is required that must parse only the fields of interest. The Parsing module wraps this logic. It is aware of which message fields to extract, depending on the type of the message. After the data is accessed and read, it must be stored so that it is available for future preprocessing. This must be executed by the Persist block. This module encapsulates the details regarding database connection and operations.

The next building block is the Data Mining. It combines two important sub-processes which are namely data preprocessing and data aggregation. In order to evaluate people relationships, a method is required to processes the social data. This is modeled by the Preprocess sub-block. In addition, to combine the peer-to-peer relationship rates based on various communication items over a period of time one additional processing module is introduced. This is the Aggregation block. It merges peer-to-peer relation values in a single tie-strength rate.

Various algorithms are represented by the Algorithms module. The fundamental sub-modules it combines are shown in Figure 4.1. These are closely related to the set of goals described in chapter 1. Few examples of these are the Roles block, which wraps the logic of finding important individual roles in the social network, the DB Merge module that introduces the algorithms for aggregating over multiple Ego networks and the Relationship Evaluation module which formalizes the algorithms to evaluate the interactions between two people in the network.

Finally, the Visualization module represents the GUI and Graph Layout modules. The GUI wraps the logic and functionality of the graphical display of the application. The Graph Layout has a very significant role to the application usability. To be able to explore the social graphs and make conclusions on the interaction patterns, it is curtail that the structure of the network is laid out suitably.

4.3 Architecture

The prototype is designed with a modular loosely coupled architecture. It is a three-tier system. The idea behind is to support easy component extension and addition of new modules. Every high-level building component is represented by a separate module which makes it easy to understand and extend in the future.

Figure 4.2 illustrates the architecture of Bunea. This is constructed of four high-level building blocks which constitute the three tiers and namely: the Data, the Business Logic and the Front
The Data Access building block represents data access-related activities like the choice of data source and connection to this. The Sources sub-module is a facade for the various sources that could be integrated. However, in our implementation this is the MS Outlook application. The calls are defined by the Outlook Interop component which is the API of Outlook. To transform the extraction commands from Bunea to native VBA calls to the Outlook COM object we introduce the JACOB component. It translates the Java calls into COM readable instructions. In summary, the combination of JACOB and Outlook Interop allows us to dynamically access any information stored by the Outlook application and exposed by the Outlook API.

Further, the process of message extraction and persisting is encapsulated in the Parsing module. This is responsible for finding if any new communication items, parse and transfer the metadata information to the JPA component which handles the persistence. The Extraction block is a facade for the method of extraction and parsing of all communication items of interest. These are emails, meetings, IMs and calls. This mechanism is hidden from the higher-level components of the architecture. The persistence process is summarized in the JPA module. This contains further the implementation details of connecting to the database. We integrate the EclipseLink framework [15] which is an implementation of the JPA specification. Bunea is designed as a Java client application. Therefore, an embedded database is required to store the extracted social data. The H2 Database Engine is selected.

The Network Creation module is a facade for the complete process of a single graph construction. It combines data mining based on application-defined algorithms and graph composition based
on predefined properties. This represents the business logic of the application.

The *Visualization* is the most top component in the architecture. The application GUI is implemented with Swing and the social graphs are visualized by the Gephi open-source Java based framework.

More information about the implementation, applied software patterns and challenges are described in chapter 5.
Chapter 5

Implementation

5.1 Project Structure

![Figure 5.1: Bunea Project Structure: Modules Hierarchy](image)

This section describes the package structure and implementation of the prototype. Figure 5.1 shows a package diagram of Bunea. For simplicity we have excluded all association links. The main building blocks are the module, app, combridge, graph, gephi and db. In the following we describe the purpose of each one.

Object creation and class dependencies are carried out by CDI (Context and Dependency Injection). The advantages are easier code change at a single place, explicit class dependencies, easier module-based testing and separation of concerns. The chosen CDI implementation is Guice [16]. The Guice modules, which are used to configure the bindings between interfaces and implementations as well as scopes are located in the modules package.
The app package is a wrapper for the GUI logic. It is the starting point of the application.

The combridge package allows the connection to Outlook. It integrates the Jacob framework [17] which translates the application requests into Outlook readable commands and contains the logic for the Outlook folder traversal and social data extraction.

The graph package wraps the logic of the data mining, preprocessing, graph construction and algorithms. In short, the graph package contains executors that determine the sequence of steps for the construction of the different types of graphs. Under the algorithms from the graph package are the referrals methods, centrality metrics and others.

The gephi package integrates the Gephi visualization framework into the application. This package contains the writers that generate the input files for Gephi, customizations of Gephi and more.

The db package implements the logic of the database interactions. It contains sub-packages as the dto, dao, entities and other that follow the software design patterns for reading and writing from and to the database. Further, the db package has another sub-package called merge which handles the merging of multiple Ego networks.

5.2 Data Collection and Transformation

5.2.1 Outlook Access

Outlook exports an API to reuse its functionality and access it from external applications. The API acts as a bridge between COM applications and managed .Net applications to support the development against the COM environment. To access Outlook one needs to call the Microsoft.Office.Interop.Outlook API or short Outlook Interop [18].

A method is required to make calls to the Outlook COM object from a Java application. For this purpose, the Java Com Bridge (Jacob) [17] framework is integrated into Bunea. Jacob is an open-source project that allows developers to make calls to COM components from Java application. It transforms the Java requests through JNI into native calls to the COM and Win libraries. The combination of Outlook Interop and Jacob allows us to make any calls to MS Outlook. However, there is no object model that wraps these requests. Instead, plain String requests are passed to the Jacob Dispatch objects.

5.2.2 Data Retrieval

In the organization under analysis, every Outlook folder such as Inbox and Outbox has a certain capacity. When this capacity is reached the older items are moved from the current (the Active) to the Archive folder. The structure of the Active folder is preserved identical to the one in the Archive folder. To capture all conversation items we parse both, the Active and the Archive folders. To handle the different types of subfolders (e.g. Inbox and Conversation History) different Parser types are implemented. The Parser wraps the logic of how a high-level folder like the Inbox is traversed. Further, the Handler from the handlers package is a class that keeps the logic of how a single item (e.g. an email) is extracted. It implements the AbstractHandler.java abstract class that holds the common logic for the different handlers.
Following is a description of the process of extracting Outlook data from the local Outlook application. The mechanism follows the Producer/Consumer design pattern. This is used to decouple processes that produce and consume with different speed independent from one another. Finding Outlook folders matching certain criteria and parsing these is a typical use case. `FolderProducer` is the implementation that finds folders and `FolderConsumer` is one which parses the selected folders. Both classes share an instance of `FolderQueue` which extends the Java ArrayBlockingQueue. The `FolderQueue` is a Singleton. The `FolderProducer` accepts an instance of the root folder of Outlook and traverses the complete tree structure. It searches for subfolders that match a given pattern. In total, 4 high-level subfolders of the Active parent folder are parsed. These are the `Inbox`, `Outbox`, Conversation history and `Calendar` and respectively the same folder types from the Archive parent folder. When a folder matching a given search criteria is found, it is added to the shared blocking queue. During the study it is observed that many employees create custom folders in the default ones e.g. in the `Inbox`. People move messages from one folder to another so that they can find the messages easier in the future. The moved messages are extracted as well in order to cover the complete conversation range.

The `Consumer` parses a single folder at a time. This process is more time consuming than finding a folder. Every `Producer` and `Consumer` is a separate thread. To optimize the performance as many `Consumers` as the number of cores on the machine are created. The `Consumers` process folders until the blocking queue is empty and the `Producer` is finished.

To further improve the performance, the prototype stores a log with the timestamp of the last point in time when new messages were extracted from any high-level Outlook folder. Thus, at any start up only new messages are read.

### 5.2.3 Challenges

By parsing the folders in multiple threads, we gain a performance advantage. However, there is another gain from this mechanism. The Jacob framework has a very unreliable garbage collection. The framework works internally with the so called `Rot` table which keeps an entry of all allocated objects. To release the memory for an object it waits for a `release` call. In this case it goes through the entire `Rot` table and release all objects from there.

The challenge is that in case when all allocated By Jacob objects are released at the end, an out-of-memory exception is thrown. On the other hand, if releasing after the parsing of a single folder, all not yet parsed folder objects are released.

On `release` Jacob only frees up the allocated memory from the current context which is the current thread. Therefore, the folders are matched in one (the `Producer`) thread and extracted in one of the (`Consumer`) threads. Further, there are employees who receive and send thousands of items per month. Thus, Outlook folders are not always of small capacity to be completely parsed before releasing the allocated memory by Jacob. Therefore, we additionally read every folder in a batch of items and release the memory after every batch. To determine the size of the batch series of tests with social peers were performed.

### 5.2.4 Data Normalization

From every item we extract the type, sender, recipients list and timestamp. The name entry does not always have the same format. However, usually it looks like **Surname, Name (Division**
The name entry of an individual can change over time as e.g. the division or organization entry (OE) can change. For example, it could happen that John Doe, an employee at Credit Suisse, was in the Operational Risk Management with OE - ORM, but moved to some other division with shortcut BMD. In this scenario the name entry for John will change from Doe, John (ORM) to Doe, John (BMD). Further, it is not only the division that changes but also the user can change his name e.g. when married or others. Therefore, a method is needed that maps people occurring with different names to the same person entry. This is crucial because otherwise some people appear multiple times in the social graph and their relation with the others appears to be weaker than it actually is. This is because the communication amount is distributed among multiple nodes representing the same person.

Before describing a mechanism that finds multiple name entries of the same person let us consider the email address entry. For Credit Suisse employees at random sometimes the output is the personal identification number (PID) of the employee (John: F123456) or the actual email address (john.doe). Mapping from the one to the other is impossible without additional information. The email is generated by the name and surname of the person where the PID number is a random not already allocated identification number starting with a letter and followed by 6 numerical signs.

To identify people uniquely, the pair \((\text{name}; \text{identification})\) is used as this is the only information available about people. The parameter \text{identification} denotes email or PID. However, there is no one-to-one relation between identification pair and person because for the same person multiple such identification pairs could occur. For example, there are cases when John Doe would appear like (Doe, John (ORM); F123456) in some messages and (Doe, John (BMD); john.doe) in others. Both, name and identification differ. Therefore, we can not say with certainty that these are the same person. However, there are cases when multiple pairs for the same person are found like (Doe, John(ORM); F123456), (Doe, John (BMD); john.doe) and a combination of both: (John, Doe (ORM); john.doe). In this case, we identify that john.doe is the email address of both Doe, John (BMD) and Doe, John (ORM). Both names are added to name aliases of the person. In addition, it is concluded that the email address and PID are the identifications of the same person and added to email aliases. In the presented case, the name combination excluding the OE shortcut was the same. However, the name cannot be used to map person to multiple identification pairs because as stated names also do change often.

In Algorithm 1 we present the complete algorithm to map people to multiple entries. The algorithm receives as input parameter a set of Person objects that have name and email. For simplicity, we have generalized the identification entry to an email entry. If a person is found in the database by a given name/email and we have found a new email/name, this is added as alias. The Person object is added for merging to the mergingSet. If a new person is created, the object is added to the persistSet that is persisted to the database at the end. For performance reasons we apply persist and merge at the end of the algorithm. Alternating operations like select, merge and persist decreases the performance. By executing only select operations during the Find Aliases algorithm and persist and merge operations at the end, the performance is improved by a factor of 10 to 15 depending on the size of the database and the incoming set of persons.
Algorithm 1 Find Aliases

Data: people: set of persons
nameMap ← empty hashtable // index for Person objects using key name
eMailMap ← empty hashtable // index for Person objects using key email
persistSet ← empty set // set of Person objects to be persisted
mergeSet ← empty set // set of Person objects to be merged

foreach p in people do
  name ← get p name;
  email ← get p email;
  person ← findBy(name); // search in the DB for a person with this name
  if person is null then
    person ← nameMap get by name;
    if person not null then
      foundNameCheckEmail();
    else
      person ← findByEmail(email); // search in the DB by email
      if person is null then
        person ← emailMap get by email;
        if person not null then
          foundEmailAddName();
        else
          createNew(name, email);
        end
      else
        foundEmailAddName();
      end
    end
  else
    foundNameCheckEmail();
  end
end

persist persistSet;
merge mergingSet;

function createNew(name, email)
  person ← new person(name, email);
  add person → persistSet;
  add pair (name, person) → nameMap;
  add pair (email, person) → emailMap;
end

function foundNameCheckEmail()
  if person has email then return;
  add email → person;
  add person → mergeSet;
  add pair (email, person) → emailMap;
end

function foundEmailAddName()
  add name → person;
  add person → mergeSet;
  add pair (name, person) → nameMap;
end
5.3 Relations Assessment Model

5.3.1 Importance Factors

A multi-level criteria that constitute the importance function is defined. The high-level factors define different properties of the interaction: Magnitude, Quality and Timing. The combination of these properties is used to identify the strong relationships.

Magnitude

The Magnitude is the parameter that defines the intensity of the interaction. A higher intensity gives a higher final score. It consists of two sub-factors which are the Frequency and the Intensity.

- Frequency

The amount of days in a month two parties communicate with each other. Higher frequency indicates a higher level of importance. To compute the Frequency we consider the number of days in a month in which the two parties communicated relative to a constant number of expected days of communication:

\[
F_{i,j}(m) = \frac{f_{\text{count}}}{f_{\text{const}}}
\]  

where \(F_{i,j}(m)\) is the frequency rate of the interaction between peers \(i\) and \(j\) in month \(m\), \(f_{\text{count}}\) is the count of days in which they have communicated and \(f_{\text{const}}\) is a constant number which normalizes the result. This is based on series of experiments to determine the average relative frequency of a user.

- Intensity

The number of communication items sent/received in a month. The formula to compute the relation intensity of any peers \(i\) and \(j\) in a month \(m\) is given by:

\[
I_{i,j}(m) = \frac{i_{\text{count}}}{i_{\text{const}}}
\]  

where \(i_{\text{count}}\) is the sum of sent and received items. Similar to the frequency computation, we introduce the \(i_{\text{const}}\) parameter which is a constant applied to normalize the output of the Intensity factor.

Quality

- Weight

Every communication event is of a certain type (Email.To, Email.BCc, Email.Cc, IM, Meeting.Accept). A weight value is defined for every type. This value depicts the concept of how direct or lasting the communication throughout this item is. Earlier it was described
that emails are used in more formal communication, whereas instant messages are more
direct and informal. The more direct the communication is, the more it contributes to the
relation. Therefore, the weights are defined as follows:

<table>
<thead>
<tr>
<th>Type</th>
<th>Weight</th>
<th>Consideration</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mail.Cc</td>
<td>1.0</td>
<td>The recipients are not the direct recipients but additional ones. Emails are used in formal communications and do not depict a close relation to the recipient</td>
</tr>
<tr>
<td>Mail.BCc</td>
<td>1.0</td>
<td>Similar consideration as in Mail.Cc.</td>
</tr>
<tr>
<td>Mail.To</td>
<td>2.0</td>
<td>Mail.To is more direct than Mail.BCc. Mails do not depict a close relationship</td>
</tr>
<tr>
<td>Lync IM</td>
<td>3.0</td>
<td>IMs are more direct and informal compared to email communication. These are often used when peers know each other on a first name basis</td>
</tr>
<tr>
<td>Accepted Meeting</td>
<td>4.0</td>
<td>Meetings last long on average. In addition, often people have meetings in person. This is another reason to assume people know each other.</td>
</tr>
</tbody>
</table>

Table 5.1: Communication Types and Default Weights

The weights are relative. To determine the final weight score \( W_{i,j}(m) \) for peers \( i \) and \( j \) in month \( m \) we compute the overall communication weight:

\[
W_{i,j}(m) = \frac{\sum_k \text{count} \cdot \text{weight}_k}{\text{count}} \cdot \frac{1}{w_{\text{const}}} \tag{5.3}
\]

where \( \text{weight}_k \) is the weight if the \( k \)th sent/received item and \( \text{count} \) is the number of exchanged messages. The constant \( w_{\text{const}} \) is introduced to normalize the output.

- **Balance**

The ratio of incoming vs. outgoing communication items between two nodes. The more skewed the Balance is (e.g. a mailbox sending regularly emails to many employees but emails sent back are close to zero), the weaker the relationship is.

The first approach to compute the Balance is to compute the ratio of incoming vs. outgoing. However, this method is not suitable because even strong relations often do not have a perfect ratio (sent equal to received). Consider two peers who send each other emails. They interact with nearly the same intensity. Hence, they should have a good Balance. Let assume that in couple of cases the first one has forgotten to attach a document to the message. Thus, he needed to send one additional email every time. In summary, it would appear that the first peer is the more active communication party and the balance would be low. Therefore, a method is researched which computes the Balance as a relation of incoming and outgoing but considers the value of the relative difference.
To model the Balance we define the following function:

\[ B_{i,j}(m) = 1 - \frac{|c_i - c_j|}{\max(c_i, c_j)} \left[ 1 - \frac{1}{e^{p \frac{c_i + c_j}{2}}} \right] \]  

where \(c_i\) and \(c_j\) are sent by peer \(i\) and peer \(j\) for any \(i, j\) and \(const\) is a constant, in our computation \(p = 0.05\). \(Balance \in [0, 1]\), where 0 is min and 1 max.

The balance function is constructed of two main parts:

\[ \text{Relative Difference} = \frac{|c_i - c_j|}{\max(c_i, c_j)} \]  

\[ \text{Adaptor} = 1 - \frac{1}{e^{p \frac{c_i + c_j}{2}}} \]  

The Relative Difference is a linear function that measures the difference of the number of sent versus received items as opposed to the maximum of both. This formula is applied to express the difference of incoming and outgoing. It does not differentiate between small and large amounts of items. To take the item amount into account, we introduce the Adaptor factor. It is an exponential function that takes as parameter the average value of incoming and outgoing. Hence, it depends on the magnitude of the interaction. The higher the magnitude is, the more the Relative Difference influences the output of the Balance. The idea is that the more messages have been exchanged, the more it is known about the relation. Therefore, the Adaptor controls the Balance ratio. In addition, the \(p\) parameter
is a constant which influences the Adaptor as shown in Figure 5.3. The lower $p$ is, the higher number of items are needed for the Adaptor output to converge to 1. A constant is selected which leads to a convergence near to the average number of items in expectation.

Finally, Figure 6.4 illustrates the output of the Balance function with different sent and received number of messages. The conclusion is that with small numbers in the range $[1-10]$ the Balance value does not fluctuate a lot. However, as we move to the higher range of $[20-300]$, higher differences are observed. Close amounts of incoming and outgoing items in this range preserves a high Balance ratio which is the goal.

- **Broadband**

  The more concentrated the group of recipients is, the higher the relevance of the communication item because the targeted audience is more concentrated. Hence, it can be concluded that people know each other in reality. Broadband factor reduces the weight of a communication item sent to a large group of recipients as e.g. 100 people. The threshold levels in Table 5.2 are based on our study and feedback from employees at Credit Suisse. We research how big the size is of the direct targeted groups. Four levels of recipients list sizes are defined which respectively control the percent of the weight of the item that is taken into account. These are as follows:

<table>
<thead>
<tr>
<th>Type</th>
<th>Max Size</th>
<th>Percent of Weight</th>
</tr>
</thead>
<tbody>
<tr>
<td>Direct</td>
<td>3</td>
<td>100</td>
</tr>
<tr>
<td>Small-Sized Group</td>
<td>10</td>
<td>75</td>
</tr>
<tr>
<td>Medium-Sized Group</td>
<td>20</td>
<td>50</td>
</tr>
<tr>
<td>Massive</td>
<td>Infinity</td>
<td>25</td>
</tr>
</tbody>
</table>

*Table 5.2: Threshold Levels for the Broadband Factor*
Our considerations are that a recipient list of up to 3 people is a direct group because very often under these recipients are the private or additional email addresses of the sender itself. A group of up to 10 members models a small group. Such messages are not direct in the sense of 1:1 but they still depict a strong relation. Therefore, these are assigned 75 percent of the weight. List size in the range of [11, 20] models a medium-sized group and preserves 50 percent of the item weight.

Timing

• Present Proximity

The more recent a communication event is, the more it contributes to the relationship. E.g. someone the user has collaborated with an year ago, is not that important as some he is interacting with currently. Therefore, a higher relevance is assigned to recent months.

5.3.2 The Importance Function

The importance function is based on the presented three main properties of an interaction which are the Magnitude, Quality and Timing. It is defined as follows:

$$IF_{i,j}(m) = \left( \frac{2I_{i,j}(m) + F_{i,j}(m)}{3} \right) \cdot \left( \frac{BrW_{i,j}(m) + B_{i,j}(m)}{2} \right) \cdot TF_m$$

(5.7)

where $I_{i,j}(m)$ denotes the intensity of the communication between nodes $i$ and $j$ in month indexed by $m$, $F_{i,j}(m)$ defines the frequency, $BrW_{i,j}(m)$ is the weighted communication where the weight of every message is altered by the broadband factor. In addition, $B_{i,j}(m)$ is the balance of this relation and $TF_m$ is the timing factor of the month in question (or the weight of the month).

To compute the relation rate for every two peers we iterate over all months of history considering the complete social data for the user. The first month respectively has $TF_1 = 1$. Sequentially, every next month has an increased timing weight by a constant factor. This constant could be selected depending on the context. The prototype is implemented to analyze the social data over a period of one year. Therefore we set to increase the $TF$ by an additive factor of 5% from the initial value. Thus, the first month of observation is not irrelevant to the final result compared to the last month. However, having a high rate in the last month gives a better overall result than having a high rate in the first month of observation.

In addition, it is observed that by considering with equal weight the two sub-factors intensity and frequency leads to the following case scenario with unjustified rate difference. Consider the scenario with the three CS employees Bob, Tim and Alice. Bob and Tim are located at the same office in Zurich. Alice on the other hand works in New York. It happens very often that an item sent from Bob is replied by Alice on the day after when she gets to work whereas Tim replies on the same day. In summary, the frequency of the communication between Bob and Alice is higher than between Bob and Tim whereas the intensity is equal. To overcome this issue we consider intensity more relevant than frequency.
5.4 Data Aggregation

This section describes the complete data mining process. The UML activity diagram in Figure 5.4 depicts the general steps for creating any view of the ones described in section 3.5 from extracting the data through data aggregation and finally visualization.

![Activity Diagram]

The activity Data Extraction was partially described in section 5.2. This represents the access to Outlook and retrieval of all new communication items. Every item is split into multiple peer-to-peer communication items with a single sender and a single recipient. These are Communication entities as illustrated in the database schema in Figure 5.5. The initial items is split to model the relation between a node A and a node B based only on their interaction with or without additional people. For example, an email message from John to 10 recipients is split into 10 Communication entities with a sender John. Meetings, IMs and calls are split into the number of all possible pairs of nodes based on the number of participants. It is considered that there is no sender but only participants. In addition, for every pair of participants John and Alice two Communication entities are persisted. The first one has a sender John and recipient Alice and the second one with reversed sender and recipient.

In case, there are new Communication items, the Persist New Communication Items activity is executed. The items from all considered Outlook folders are persisted to the database in table Communication. Thus, this table is the storage of raw social data split into pairs if sender and recipient as described above. The timestamp of the message is persisted to aggregate based on the time proximity, number of recipients is stored to apply later the Broadband factor and type of the item to assign the appropriate item weight.

On every start of the prototype only the new items are extracted and persisted. If there are no new items the data from the last run is reused. The same applies to the aggregated data.

From all Communication items in the period of observation and between any two peers a rate of the relationship is constructed. If there is no new data the relation value can not have
changed from the perspective of the application. Therefore, to improve the performance by not recomputing the value of the relationship over the complete time period, this is split in monthly evaluations. These are persisted in the table MonthlyCommunication. If new communication items of a readily persisted month are extracted, the rate is re-evaluated and updated into the database. This process is illustrated in Figure 5.6.

The Determine New Months to Aggregate step selects the months for which there is no aggregation history in the database but available Communication entities from the period of question which must be aggregated. These months are selected and sequentially passed to the aggregation step. In case there is no data for a new month, the next steps checks whether there is a month to be re-evaluated. Re-evaluation is required when the last evaluation was on a subset of the data currently available for the month.

So far, the high-level communication aggregation over a period of one month was presented. In the following we present how the monthly interaction mining is implemented. To recall, to determine the relationship rate between any two peers the following parameters are required:

- Frequency of the interaction: number of days communicated
- Intensity of the communication: number of exchanged items
- Number of recipients of any communication item
- Type of the item

The MonthlyCommunication entities store the peers of observation, month timestamp, exchanged
Chapter 5. Implementation

Figure 5.6: The method of selection of communication months to be aggregated or re-evaluated

items count, type of the aggregated items and weighted sum. The final monthly rate is not persisted. The reason is that only a single new e.g. email message will require that the complete month is re-evaluated. In addition, if the user requires to explore the social graph based only on e.g. IM communication the relationship rates must be recomputed. Therefore, the mining is oriented towards the following parameters creating separate aggregated units:

- Peer-to-Peer Basis
- Type of Communication
- Month

The final monthly communication rate is computed when required. In addition to MonthlyCommunication which depicts the intensity and weight of the communication, the Frequency entity is introduced. The logic of the frequency of the interaction is separated. This way if a new item is extracted only the reflected MonthlyCommunication entity is updated together with the Frequency entity for the month of observation.

Thus, the problem of mining the social history has been narrowed down to the subtask of extracting the information presented by the two entities Frequency and MonthlyCommunication (MC). The process of extracting these is illustrated in Figure 5.7. It takes as an input parameter the month that is being mined. To determine the frequency between any pair of peers in a given month the SQL query presented in Listing A.1 is applied. The list of item types is presented in table 5.1.

To aggregate the MC related information, the following parameters are retrieved for every pair of sender and receiver given the communication type and month:

(P1) Intensity by number of sent items: [sender, receiver, nrItems]
(P2) Number of items grouped by the count of recipients: [sender, receiver, nrRecipients, nrItems]

To compose a new MonthlyCommunication entity the weighted sum for the given communication type needs to be computed. This is determined based on the number of recipients (Broadband),
Chapter 5. Implementation

Figure 5.7: The process of extracting Monthly Communication and Frequency information

the predefined weight of the type and the count of items. The more difficult part is to map the weighted sum to the overall number of items of this type within a reasonable time. We could iterate over the (P1) entities and inside this cycle over the (P2) entities and compare the sender and receiver. However, this mechanism is time consuming because sender and receiver are String identifiers.

Algorithm 2 Aggregate Monthly Communication

Data: month, type, weight

j ← 0
AList ← empty List<A>
/*A fields: sender, receiver, nrItems*/
BList ← empty List<B>
/*B fields: sender, receiver, nrRecipients, nrItems*/

foreach a in AList do
    weightedSum ← 0
    sender ← a.getSender
    receiver ← a.getReceiver
    count ← a.getCount
    i ← 1
    while i ≤ count do
        b ← BList.get(j)
        nrRecp ← b.getNrRecp
        nrItems ← b.getNrItems
        i ← i + nrItems
        j ← j + 1
        weightedSum ← weightedSum + computeWS(nrRecp, nrItems, weight)
    end
    mc ← createNewMonthlyCommunication(month, sender, receiver, type, weightedSum, count)
end
Algorithm 2 is an improvement of this process. It is based on the knowledge that the sum of all communication items of a given type is equal to the sum of all items grouped by the number of recipients. The (P1) and (P2) entities must be ordered by the pair sender, receiver.

Finally, to retrieve an evaluation value for every relationship, the SQL query listed under A.2 is applied.

5.5 Ego Networks Merge

Section 3.4 described the two high-level categories My Network and Our Network. Our approach on building the social graph is based on My Network because of privacy and security concerns. This network is suitable for analysis that improves the productivity of the user and presents an overview of the key actors. However, to model global questions the Ego network is not sufficient as it is from the viewpoint of a single user.

To overcome this constraint a collaborative global network creation based on the Ego network is introduced. The Network table stores the complete Ego network. This is represented by a set of Edge entities that contain both ends of the edge and a relation value. One way to merge multiple networks would be to iterate over all MonthlyCommunication entries of each database to find the union of communication items persisted in all databases. However, this is a time consuming task since a medium-sized MonthlyCommunication table over a period of 9 months contains about 50000 entries. In addition, for the merged network to be consistent we need to aggregate over multiple databases.

The merging algorithm is based on the following five statements:

1. For every pair of nodes there is a complete and a partial evaluation rate. If the rate is based on the complete communication volume between the peers, the rate is complete. Otherwise it is partial.
2. Every Ego network contains the complete rate from the Ego to all other peers.
3. The rate of the relationship between two nodes A and B has the same value in A’s network as well as in B’s
4. For the peers for which there is no contributed Ego network the rates to the other people are partial except to the peers who contributed their networks.
5. The highest computed rate between two peers is the most accurate one. This assumption is incorrect in cases where the relation has deteriorated. However, for the purpose of the prototype we assume that this statement is correct.

To build the global network the edges with a complete evaluation rate must be identified and added to Our Network. When no complete rates are available, the highest persisted value is selected based on statement five. The complete algorithm is given in Listing A.3 in the appendix. Throughout the execution the edges which must be persisted or updated are added to temporal sets. These are persisted or updated in a batch. The reason is not to alternate between Select operations and Persist/Merge operations which adds high latencies by using the EclipseLink JPA.
5.6 Graph Construction and Visualization

In Bunea different network views are modeled as described in section 3.5. Each of these focuses on a different network properties and actors. This section presents a brief description on the different construction methods and layouts applied.

As described, *My Circle* selects the top 30 people from *My Network* that have the strongest relation to the user. They are presented in a user-friendly circle layout and ordered by the strength of the relation. The *Ego* has a direct connection to all top 30 and is thus the center of the network.

*My Circle Over Time* is a complement of the previous one. It illustrates the relationship evolution on a single month basis. To track how the status a simple method is applied which compares the current status to these from two months before. In cases when a node has the same position in the *circle* as in the previous month, it is modeled that this keeps its position. To visualize this we color the node in blue. When a node lowers its ranking compared to the previous month but not two months in sequence it is marked in orange. The two months in sequence deterioration of the relationship is a sign that over a longer period of time the user and this person have had a weaker communication than before. Therefore, such nodes are colored in red. In comparison, nodes which improve their position are marked with green. This intuitive method makes it easy to follow how peers’ ranking change over time.

The *Direct Peers* network includes the communication partners with whom the user has a direct relation evaluated over a certain threshold. The threshold is selected based on series of experiments. Its goal is to filter out the nodes with whom the user exchanged only few items over a very long period of time like an year or relationships that are one-directional. Hence, the not important communication partners are excluded from the network. The *Fast Unfolding* algorithm [8] is applied to find the communities of practice. These are colored uniquely. In addition, the Fruchterman-Reingold layout algorithm [19] is executed on the network. This groups together and lays out the nodes which belong to the same or close clusters.

In *My Network* and *Global Groups* based in experiments a threshold $t$ is selected to visualize the $t$ number of nodes who have the highest sum of relationship values to other peers. To find the communities of practice the *Fast Unfolding* algorithm is applied and the clusters colored uniquely. To lay out the members of the same cluster together we develop a clustering layout algorithm. This is inspired by the *Group-In-A-Box* layout mechanism [20]. The authors describe the approach of their algorithm as well as the use cases of the layout. As in the *Group-In-A-Box* layout we apply the treemap space dividing technique [21] to assign each community a space rectangle in the final graph. We apply the unordered squarified treemap algorithm to achieve better visualization results than with the other tree map methods. In addition, in every cluster the nodes are placed at random with the exception of the most social node who is centered in the assigned rectangle. Thus, the most edges are directed to the center of every community box.

Finally, the *Division Relations* network focuses on the inter-cluster relationships between organizational divisions. To visualize this we aggregate all communication items exchanged between the members of the first unit and the members of the second into one inter-unit connection. The level of aggregation is given as input by the user.
5.7 Finding Important Actors

To find the significant roles in the network as presented in 3.6, the social network metrics as defined in 2.2 are applied.

To determine the Social Hubs in the graph, the Degree centrality metric is computed for all nodes. The ones with the highest degree are the most social peers.

To determine the Information Brokers the betweenness centrality metric is applied. As described in section 2.2.1, these are the nodes that lie on most of the shortest paths between any two nodes in the network. In the modeled graph the edges have a positive value which is the strength of the relation. The inverse of this rate is used to mark the connection distance. The lower the relation value between two nodes, the bigger the distance is. To find all the shortest paths in the network we use Dijkstra’s algorithm. To boost the performance we implement it with a Fibonacci heap. This requires $O(|E| + |V|\log|V|)$ time to find the shortest past from a given node to all others. $|V|$ is the number of vertices and $|E|$ is the number of edges.

On every iteration of the algorithm (for every starting node) the sum of shortest paths to all other nodes is computed. This is reused by the closeness centrality metric. To find the number of shortest paths going through a node a counter is introduced. Every time a new shortest path is found, the counter of all nodes on the path is increased by one.

The closeness centrality is computed to find the Coordinators in the network. This is implemented as shown in Formula 2.3. The output is the inverse of the sum of the shortest paths to all other nodes. This way the result follows the intuitive understanding that higher closeness centrality is better than a lower one. The computation is done iterating over all nodes as described above for the betweenness centrality. The algorithms runs once for both centralities because they require equal steps with some minimal overhead to compute both.

A main limitation of the closeness centrality is its applicability only to connected graphs. There is no finite path between nodes in different components. To overcome this issue a value is determined which rates the distance between not connected by a path nodes in the network. This value is based on measurements of minimum and maximum edge weight in the graph and the Milgram’s ‘Small-World’ phenomenon [3] which determines the average path between any two nodes in a network.

To identify the influential people Formula 2.4 is applied. To take into consideration the strength of the relationships we apply this on the weighted graph. Therefore, instead of the adjacency matrix $A_{i,j}$ the weighted adjacency matrix is used in the computation. After every iteration the temporary centralities are normalized by $\lambda = \max_{i \in |V|} s_i$.

5.8 Referrals

In section 3.7 the three criteria to find a best referral path are introduced. To recall these were:

1. Min hop count
2. Max edge weights
3. Balanced path
These parameters define the properties of the best chain of referrals. However, it is not deterministic what the trade-off must be for a solution to be in all cases the most optimal. Let us consider the following example which is very similar to the problem of finding the best referral path. The task is to define the most optimal way to get from location A to location B using public transport. In order to model this problem we need to define what properties are most favorable. However, similar to our use case, this is not deterministic. There are users that prefer to travel e.g. 5 minutes more but not need to make many changes along the way. On the other hand, others request to arrive as fast as possible.

In finding the most optimal referral chain there is no single solution. However, the criteria from above are all important properties of the path. Therefore, to solve different requirements we model three methods that find a chain of referrals. We claim that the Best path method is the most reasonable one because it is based on a trade-off between the three criteria. The methods are as follows:

- *Fastest path:* Contains a minimum number of intermediate hops.
- *Strongest path:* Max sum of edge weights along the path
- *Best path:* A trade-off between min hop count, max edge weights and balance of the weights

It was stated that the value of an edge represents the strength of the relationship. Hence, the higher the weight, the stronger the relation is. To explain the different methods we introduce Figure 5.8 which shows an example of referral chains from the Ego to another peer John. The precondition of the algorithm is that there is no direct connection between the user and John. In case there is, the user does not need an introduction. Hence, their relationship evaluated to 0. For simplicity, it is assumed that all edge weights are in the range [0, 10] where 10 is the maximum weight.

![Figure 5.8: An example of various referral paths](image)

The Fastest path is the one that selects the minimum number of hops to get to the target node. This method does not take into consideration the relationship between peers. Thus, to find this path we apply Dijkstra’s algorithm to the unweighted graph. It is the fastest in the sense that it selects the minimum number of intermediate individuals.
On the other hand, the *Strongest path* is based on criteria 2): Max edge weights. However, to avoid the selection of paths that go through unlimited number of individuals since the sum of weights can only grow with every additional intermediate the edge weights are inversed. Thus, the method is transformed into finding the shortest path. Dijkstra’s algorithm is applied to find the solution.

Both methods take only single criteria of finding the best path into consideration. However, usually finding the most optimal path depends on the trade-off between all criteria listed above. The solution is based on the following consideration:

*When is the selection of an additional edge justified?*

![Diagram](image)

*Figure 5.9: Examples illustrating the considerations when an additional hop is justified*

To model the trade-off in cases when there is no path which is both the *Fastest* and the *Strongest* one, a formalization is required that justifies the addition of one more edge on the path. Figure 5.9 shows 4 simple examples that illustrate the problem. Consider a path consisting of a single edge of weight 5. It is asked which is the equally rated referral path that contains one additional edge. Case (1) with edge weights lower or equal to 5 can not improve (0). However, case (2) illustrates an example where both edges on the path have a higher rate than the edge in (0). We claim that in the *best* chain of referrals problem (2) is not better than (1) because both edges are only by 10% higher than the initial case. However, this case involves one additional intermediate on the path to the target. Since the more individuals involved, the lower the likelihood to have a successful introduction it is stated that path (2) should be rated lower than (1). It is a matter of heuristic how higher must the edge rate be in order to be justified the addition of one more hop on. Transforming this example into relationships between people, a relatively stronger rate should confirm that an additional individual on the path improves the chance of success. Therefore, it is defined that case (3) is a better path than (0) where the weights are in the range [0, 10].

This consideration is modeled by the following equation:

\[
\frac{1}{x^r - w} = 2 \frac{1}{x^r - (w + p \cdot r)}
\]  

(5.8)

where \(w\) denotes the weight of the single edge path and \(r\) the maximum weight in the graph. Parameter \(p\) is a constant that defines what percentage of \(r\) models equal rates between single edge path or path of two edges weighted equally like:

\[
w = 2 \cdot (w + p \cdot r)
\]  

(5.9)

In addition, \(x\) in Formula 5.8 needs to be a constant near to the exponent in order to force high-rated edges make a difference in the choice of the solution towards the more hops path. Solving equation 5.8 results in the following formula for transforming the original value \(w\) of
each edge for the purpose of the Best path method:

\[ w_{new} = 2^{\frac{r - w_{old}}{p}} \]  \hspace{1cm} (5.10)

The transformed values are in the range \([1, \text{max}]\) where 1 is the best possible new value and \(\text{max}\) depends on \(r\) and the constant \(p\). Figure 5.10 illustrates that function 5.10 has near to exponential behavior where \(x\) is derived from the model and not explicitly set. To find the optimal path we apply Dijkstra’s shortest path algorithm in the transformed valued graph.

![Figure 5.10: Edge weight transformation for the requirements of the Best path method. Original weights range = 10 and percentage = 0.2](image)

In addition, both the Strongest path and Best path methods give a higher score to a balanced route as shown in Figure 5.11, path (6) compared to path (7).

![Figure 5.11: Two examples illustrating a balanced path (6) and a non-balanced one (7)](image)

To compare the different path methods we refer to Figure 5.8. Path (1) has the minimum number of hops. Therefore, it is the Shortest solution. It is used in cases where the strength of the relationships is not important. It is arguable whether solution (2) which is derived using the Strongest path method or solution (3), using the Best path method is better. In (2), the consideration is that the shortest path with lower-valued weights is better whereas according to the Best path algorithm one hop longer path with weights in the upper range is optimal.

In addition, two more possible path are illustrated. They are not under the solutions using any of the methods. The alternative path (4) would be preferred over path (5) using the Strongest path method whereas path (5) would be preferred over (4) with the Best one. It is again not deterministic which path is better. On the one hand, one would prefer to add one additional individual in the chain but ensure a better relationship. On the other hand, some users would try to get introduced using a limited number of intermediates even when these have a weaker relation to the next person on the path. We leave it up to the user to choose his best option from the three solutions. In many cases these overlap.
Chapter 6

Evaluation and Results

This chapter presents an evaluation of the *Importance Function* described in 5.3.2, results and insights from the modeled social graphs.

6.1 Evaluation

Section 5.6 introduced an overview of the different networks and views implemented in Bunea. Figure 6.1 illustrates the *My Circle* network which shows the strongest relationships of the user over a chosen period of time. For privacy reasons the presented *My Circle* is anonymized.

![Anonymized My Circle: Best 30 Relationships](image)

*Figure 6.1: Anonymized My Circle: Best 30 Relationships*

To evaluate the strongest relationships, series of measurements are performed based on various single criteria which are presented in Figure 6.7 together with an illustration of the scores from
the *Importance Function* in 6.6. The measurements are conducted on the electronic communication of a relatively intensively interactive employee. Hence, these are measurements on a real data of a period of 9 months in year 2013. For privacy reasons, details about the job profile of the user are not included.

The measurements present the 20 top scored peers according to the applied criteria. The chosen parameters measure how strong a relationship is and how intensive the communication was. The same anonymization pattern as in Figure 6.1 is applied to these charts. Hence, *P1* from Figure 6.1 is labeled *P1* in all of the evaluation charts.

The first proof of the reliability of the *Importance Function* is that all circle peers are under top 20 in chart 6.3 which depicts the total number of items exchanged between two persons, thus the intensity of the relationship. This criteria is the most intuitive parameter to determine the strength of a relationship. In the previous chapter, we argued that the intensity of the communication must not be applied in an isolation from other criteria to determine the *best* peers because there are scenarios where the communication flow is very intensive but only one-directional. Since a good relationship involves both parties to be interested in the relationship and thus take part, such relationship must not be evaluated as strong. However, in the network of an intensively-communicating peer, as the one under scope, the top 20 peers based on the *Intensity* criteria are in reality two-directional communication partners. Mailboxes and news feeds which are one-directional interaction partners show a lower intensity in this experiment and are therefore not presented in top 20.

![Intensity over Time](image)

*Figure 6.2: Comparison of the intensity of the communication with different people over time*

However, the most intensive communication partners appear in the same sequence as in *My Circle* only until position 6. This proves that according to our relationship evaluation method the intensity is only one side of the overall score of the relation. To prove that this statement is correct, we include the other measurements in the comparison.

Let us take as an example the communication to peer *P7* and to *P11*. According to Figure 6.3 *P11* has more intensive communication with the user than *P7*. On the other hand, Figure 6.5 shows that *P7* has a higher frequency of interaction with the *Ego* than *P11*. The balance of the communication or sent vs. received illustrated by Figure 6.4 shows near to perfectly balanced communication with peer *P7* whereas with *P11* there is a high difference in sent versus received items. Finally, to take into account the *Timing* factor we introduce Figure 6.2 that presents
the intensity of the communication over time. There are high fluctuations in the interaction with both peers over the various months in observation. However, in the last month of the study (September) the communication with $P7$ is by more than a factor of 2 higher than with $P11$. Thus, since the more recent the interaction is, the higher the overall score must be, we conclude that the intensity factor considered together with the timing does not lead to a better overall communication with peer $P11$ than with $P7$. In conclusion, considering all factors of the interaction we prove that although both peers have a strong relationship with the user, peer $P7$ must have a higher overall score. To compare with the output of the Importance function, $P11$ has although lower, a relatively near score to $P7$.

6.2 Insights

This section presents the results and insights from our study. We show examples of some of the graphs that we model and views from the implemented prototype.

Figure 6.8 illustrates My Circle Over Time which is based on the real communication data of the same employee analyzed in the previous section. As described earlier this network helps the user track the status of his relationships over time and detect peers that have been neglected if such.

For the proof of concept we track the relation evolution of four peers: $P1$, $P4$, $P6$ and $P21$. 
The anonymized names correspond to the names given in Figure 6.1. Hence, over the complete period of observation these peers take positions 1, 4, 6 and 21 in My Circle.

The study is conducted on the communication data of the time period January 2013 to October 2013. Hence, prior to January there is no communication history. Therefore, the top 30 relations illustrated in this month are in blue which means that these hold their position in the Circle. Let us now compare every 2 sequential months to see how the positions change. In February peers P1 and P6 keep their ratings. However, it is observed that P21 has a lower rating compared to the previous month. In March, P21 is colored in red which is a sign that the peer sequentially lowered his rating in the Circle and that the relationship to the user is weaker compared to 2 months before. To find the reason for this degradation we compare with Figure 6.2 which illustrates the intensity of the communication over time. It is shown that the interaction has been intensive in the first three months and strongly has degraded after that. The real case scenario illustrated by the network is the following. P21 has been the supervisor of the user which is the reason they have been interacting intensively up to the beginning of year 2013. In March, P21 is moved to another division which explains the weaker relationship. However, the user tries to keep this contact active which is illustrated in month May where P21 is again under the best 30 relations of the Ego.

Another interesting phenomena is the appearance of peer P4. As denoted by the label, this individual has place 4 in the overall ranking of the year. However, he or she is not under top 30 in the first two months. In March and April, this person keeps position 19 and from then on
keeps improving the relationship. This is a new contact of the user who has been introduced in March for the first time.

Finally, peers $P1$ and $P6$, both show some fluctuation in the relationship ranking. $P1$ keeps over the six-months observation period position one with the only exception of month April where he moved down in the ranking to position 4. On the other hand, peer $P6$ shows a higher variation in the ranking. In the first two months, he keeps the leading position 4. In March and April the relation degrades and after that it is strengthen again.

In conclusion, *My Circle Over Time* illustrates which are the strong relations of the user and how does their status change over time. It is easy to follow who are the peers that receive less attention of the user or vice versa. This allows ones to keep track of his most significant contacts and preserve their status.

![Figure 6.9: Direct Peers in My Network](image)

The complement of *My Circle*, which is *Direct Peers*, is illustrated in Figure 6.9. For privacy reasons in this and all further snapshots the labels are hidden. This network depicts all direct relations of the *Ego* and their strength noted by the size of the node. The graph shows further how are the direct contacts related to each other and which are the communities of practice that they form. The user distinguishes groups based on current projects, others based on the office location as well as such that group family or friends members together. This is the overview of the real social network, thus the graph that contains the two-directional communication partners. This network fulfills its goal to make the user aware of his social environment and illustrate different properties like relation strength, social communities and complete set of business contacts in- and outside the company. In addition, this network can make a necessary ‘Know-how-transfer’ process easier and more efficient than going manually through all possible contacts and their role in one’s work.

To get further insights about the social patterns and individual roles on a global scale we performed a collaborative study which aggregated the *Ego* networks from eight users from a single team. As expected, the results are biased from the participants. However, interaction patterns can be identified as well as significant peers in the network that are not under the network con-
Chapter 6. Evaluation and Results

Figure 6.10: Global Communities in Our Network

tributors. To examine this we refer to Figure 6.10. The database volunteers are labeled with $X[1-8]$. As stated these individuals are from the same team. However, in reality their projects have a limited overlap as they are responsible for different topics. In addition, the subset of people they work with is to high extend different. This can be seen in the created social graph. The identified social communities are colored uniquely and the members are located in the boundaries of the community. It is visible in the network that all team members except $X5$ and $X6$ are part of different informal communities. In reality, these are the two persons from the team that share a higher number of projects together. In addition, although working on different topics these employees have weekly meetings together and further more interactions. This is the reason why although being part of different communities they have a strong relation to one another. It is interesting to observe that peer $X9$ who did not take part in the study can be identified in the social network. This individual is part of the same team but responsible for yet another topic. Therefore, he is located in a separate community.

In this study we identify peers significance to the social network and different individual roles like Information Brokers, Influential People, Social Hubs and other. However, for the brevity of the thesis we analyze only the Influential People in Our Network who are illustrated in Figure 6.11. For further insights on the global network please refer to Appendix B.

The rate of the hue identifies how influential a person is, where red marks very maximal influence and dark blue marks low influence. It is interesting to identify many additional peers with high influence who are different from the network contributors. The team members or nodes $X1$ to $X8$ have a high significance in the network. However, their rates differ. To recall, the influential people in the network are the ones that have a strong relation to other influential people. We observe that the two most important people according to this criteria are person $X3$ who is leading the team and person $X1$ who is not officially part of the team but at the same hierarchy.
level as \( X_3 \). It is shown that both peers are very important to the network as they are connected to other significant peers. Hence, they are suitable intermediates to get introduced to a new contact.

The network is biased from the database contributors directly since the global view is based on their electronic communication. However, the identification of other influential people is a proof that these have in reality strong relationship to significant to the network peers. Hence, they are of high significance as well. To point only a few of them, it is observed that peers \( X_{12}, X_{19}, X_{20} \) appear to have a high impact on the network.

As a last example, the results from the divisional-based relationships network are analyzed. The collaboration across the hierarchy is modeled to identify missing ties and weak connections. The prototype allows the relation exploration at a different level of granularity. To explore the results we request the connection strength at a medium level in the hierarchy. Hence, one unit which is presented by a single node potentially combines multiple sub-teams. Figure 6.12 illustrates the results from the aggregation. The contributed \( Ego \) networks are from unit \( A \). Hence, the knowledge about the communication from \( A \) to all other units is modeled. It is visible that there is no connection to unit \( I \) which is a medium-sized unit. This means that there is a missing tie in the network and no collaboration with this organizational unit. In addition, \( A \) has a strong relation to the units \( K, N, B \) and \( E \) but the ties to \( J, P \) and \( C \) are weak. Thus, it must be closely observed that the relevance between these organizational unit is. It could possibly be identified that the collaboration between these must be improved in order to use more efficiently the knowledge inside the organization and drive this way better results.

Figure 6.11: The influential people in Our Network
Figure 6.12: Collaboration across the hierarchy
Chapter 7

Conclusions

7.1 Contribution

In this Master’s thesis, we studied different use cases applicable to organizations to improve employees awareness to other people in the company and improve the collaboration. Individuals’ electronic communication data is mined to extract insights about people interaction patterns and key business partners.

A novel social data mining method is developed to evaluate one’s relationships with his co-workers and external to the organization people. Individual roles in the graph and key actors were detected to define who are the significant people and what is their position with regard to the other peers. The evolution of the strongest relationships of the user is tracked over time to help keeping important actors active in the network and introduce an overview of the actual communication intensity. Communities of practice are visualized on different levels of granularity to illustrate the social patterns and oppose these to the formal structure of the organization.

In addition, a mechanism is designed to aggregate multiple single-user networks into one global graph. This is used to further illustrate global communities, practices and insights about the merged organizational network. An algorithm is implemented to find the most optimal chain of referrals to utilize peers contacts to receive an introduction to a target of interest from an acquaintance of his. The collaboration across the organizational units is modeled to discover disconnects and weak ties between teams, and divisions.

In conclusion, the implemented relation assessment mechanism is evaluated against different criteria to verify that the concept of rating people’s interaction holds. It is shown that it is valid against different quantitative measures and thus can reliably determine the strength of any business relationship.

7.2 Future Work

There are various directions for future work. The current implementation of Bunea binds to the local Outlook application. An interesting aspect to us is to develop a mechanism that solves
the privacy concerns about connecting to the MS Exchange Server. The idea is to extend the prototype to work with the real global communication data. Hence, to utilize its full capabilities to identify collaboration patterns on a global scale. This would lead to more reliable results as it would show insights based on higher communication data volumes.

In addition, a mechanism that utilizes the collaboration across the hierarchy and shows a possible reorganization models is of high interest as this could solve problems occurring in many enterprises. This is a useful complement to discovering divisional disconnects as it might suggest how these can be reorganized to improve the collaboration and thus lead to more innovations in the enterprise compared to the present. However, for this model to be reliable a complete global network is required.
Bibliography


Appendix A

Code Snippets

Listing A.1: Find Frequency on Peer-to-peer Basis

```sql
SELECT t.usender, t.ureceiver, Count(*)
FROM (  
    SELECT DISTINCT sender as usender, receiver as ureceiver,  
    day as uday, month as umonth  
    FROM (  
        SELECT sender, receiver,  
        Day(date) as day, Month(date) as month  
        FROM Communication  
        WHERE date >= STARTDAY  
        AND date <= ENDDAY  
        GROUP BY sender, receiver,  
        Day(date), Month(date)  
    UNION (  
        SELECT receiver as sender, sender as receiver,  
        Day(date) as day, Month(date) as month  
        FROM Communication  
        WHERE date >= STARTDAY  
        AND date <= ENDDAY  
        GROUP BY sender, receiver,  
        Day(date), Month(date)  
    )  
) t
GROUP BY t.usender, t.ureceiver
```

57
SELECT q.peerOne, q.peerTwo, q.ws, q.c1, q.c2, w.freq
FROM
  (SELECT peerOne, peerTwo, SUM(weightedSum) as ws,
       SUM(countOne) as c1, SUM(countTwo) as c2
       FROM MonthlyCommunication
       WHERE month >= StartMonth
       AND month <= EndMonth
       GROUP BY peerOne, peerTwo) as q
JOIN
  (SELECT peerOne, peerTwo, SUM(frequency) as freq
       FROM Frequency
       WHERE month >= StartMonth
       AND month <= EndMonth
       GROUP BY peerOne, peerTwo) as w
ON q.peerOne = w.peerOne AND q.peerTwo = w.peerTwo
Listing A.3: Merge Ego Networks into Our Network

**Data:** egoNets: list of EgoNetworks
persistSet ← empty List<Edge>
mergeSet ← empty List<Edge>
idx ← 0
ourNet ← empty Network

**foreach eNet in egoNets do**

  **if idx == 0 then**
  allEdges ← eNet.getEdges
  ourNet ← add allEdges
  **else**
  **foreach edge in eNet do**
  peerOne ← edge.getPeerOne
  peerTwo ← edge.getPeerTwo

  /* search in ourNet, persistSet and mergeSet*/
  edgeInMerged ← find(peerOne, peerTwo)

  **if edgeInMerged not null then**
  oldVal ← edgeInMerged.getVal
  tempVal ← edge.getVal

  **if tempVal > oldVal then**
  edgeInMerged ← updateVal(tempVal)
  setMerged ← add edgeInMerged
  **end**
  **else**
  val ← edge.getVal
  setPersist ← add new Edge(peerOne, peerTwo, val)
  **end**
  **end**

**idx ← idx + 1**
**end**
persist persistSet;
merge mergeSet;
Appendix B

Bunea Snapshots

Figure B.1: The coordinator individuals in Our Network
Figure B.2: The Information Brokers in Our Network