UNIVERSITÀ DI PISA
Scuola di Dottorato in Ingegneria "Leonardo da Vinci"

Corso di Dottorato di Ricerca in INGEGNERIA DELL'INFORMAZIONE

Tesi di Dottorato di Ricerca

Analysis of Online Social Networks for the Design of Cyber-Physical Mobile Social Networking Services

Valerio Arnaboldi

Anno 2014

UNIVERSITÀ DI PISA

Scuola di Dottorato in Ingegneria "Leonardo da Vinci"

Corso di Dottorato di Ricerca in
INGEGNERIA DELL'INFORMAZIONE

Tesi di Dottorato di Ricerca

Analysis of Online Social Networks for the Design of Cyber-Physical Mobile Social Networking Services

Autore:

Valerio Arnaboldi _____

Relatori:

Prof. Enzo Mingozzi _____

Dott. Marco Conti _____

Ing. Andrea Passarella _____

Anno 2014
SSD ING-INF/05

Sommario

L'utilizzo di Online Social Network (OSN) e la diffusione di dispositivi mobili stanno cambiando le modalità con cui le persone interagiscono fra loro. I dispositivi, dotati di svariati sensori, permettono la creazione di contenuti legati al mondo fisico che circonda l'utente. I contenuti sono poi condivisi nel mondo virtuale tramite servizi come le OSN. Il legame fra questi due mondi genera nuove opportunità per i sistemi di comunicazione. Ad esempio, le interfacce wireless dei dispositivi possono essere utilizzate per abilitare la comunicazione diretta fra persone in prossimità. Ciò ha portato alla nascita delle Mobile Social Network (MSN), estensione delle OSN con il supporto a reti opportunistiche.

Lo studio delle proprietà delle OSN può rivelare schemi di comunicazione e abitudini degli utenti utili all'ottimizzazione della comunicazione nelle MSN. Ad esempio, un'elevata frequenza di interazione fra gli utenti nelle OSN può rivelare la presenza di contatti nel mondo fisico, canale principale per la comunicazione su MSN.

Molte delle proprietà strutturali delle OSN non sono però note in letteratura, il che risulta un limite per la progettazione di MSN. Ulteriore limitazione è la mancanza di una piattaforma comune che doni alle MSN un insieme di funzionalità di base, come servizi di accesso a comunicazione su rete opportunistica, collezionamento ed elaborazione di dati di contesto e sociali e gestione di dati di sensori.

Questa tesi presenta un'analisi delle proprietà strutturali delle OSN che pone le basi per la creazione di MSN future, che possano considerare aspetti sociali degli utenti per l'ottimizzazione della comunicazione e la personalizzazione dei servizi. Inoltre, questa tesi presenta una nuova piattaforma middleware per dispositivi mobili, chiamata CAMEO, che dona una serie di funzionalità di comunicazione su rete opportunistica e facilita la gestione di dati di contesto e sociali.

Abstract

The use of Online Social Networks (OSNs) and the proliferation of mobile devices witnessed in the last few years is changing the way people interact with each other. Mobile devices, equipped with several sensors, allow users to create content concerning the physical world around them. This content is then shared in the cyber world of digital communications through services such as OSNs. In this scenario, new opportunities are arising for the communication systems. For example, wireless interfaces can be used to enable direct communications between devices. This paved the way for the creation of Mobile Social Networks (MSNs), that extend OSNs with support for opportunistic networking.

The properties of OSNs could reveal aspects such as communication patterns and habits useful for optimising communication within MSNs. For example, the frequency of contact between people in OSNs could indicate the presence of contacts in the physical world.

Unfortunately, many structural properties of OSNs are still unknown, resulting in a limitation to the design of new MSNs. Another strong limitation to MSNs is the lack of a common platform able to provide basic functionality, such as the access to opportunistic networking services, the collection and elaboration of social and context data, and the management of sensing data.

This thesis provides a detailed analysis of the structural properties of OSNs. The results represent one of the bases for the design of future MSNs, considering social aspect of the users to optimise communications and to personalise services. Moreover, this thesis presents a novel middleware platform for mobile devices, called CAMEO, that is able to provide opportunistic networking functionality, and ease the management of social and context data.

This thesis is dedicated to my family
and to Viviana,
for the love they gave me during these three years

Contents

List of Figures

List of Tables

1

Introduction

In the last few years the advent of Online Social Networks (hereinafter OSNs), such as Facebook and Twitter, and advances in the field of mobile computing have changed to a large extent the way people communicate with each other. Modern mobile devices are equipped with several sensors (e.g. camera, GPS, accelerometer) and communication interfaces (e.g. Wi-Fi and Bluetooth), which enable the users to generate content enriched with information concerning the physical world and to transfer it to the cyber world of digital communications. Thence, services such as OSNs allow the users to share the generated content in a real-time fashion with their social contacts. This actively contributes to the so called cyber-physical world (CPW) convergence, which envisions a world where virtual interactions between people and the physical communication systems are knitted into a single whole [41]. In the CPW, actions taken in the cyber world depend upon conditions of the physical world and vice versa. For example, in OSNs, users often communicate with people they already know in the physical world [38], and events occurring in the physical world affect communication in the cyber world.

In this context, several new opportunities for future communication systems are arising. For example, wireless interfaces can be used to enable direct communications between devices in proximity. Based on this concept, a new communication paradigm for the Future Internet, called Mobile Social Networking, has been proposed. Mobile Social Networks (hereinafter MSNs) exploit the mobility of the users and the ability of their devices to communicate directly with each other, through their wireless interfaces, to share user-generated content by means of opportunistic networking [80], without the need to access centralised servers. MSNs directly inherit the communication mechanisms of OSNs, but they exploit the physical contacts between nodes (i.e. the ensemble of the user and her device) in proximity to

allow information to spread in the network. Content is stored and carried by the nodes. When nodes encounter other nodes, they pass them the content through direct communication.

Physical contacts between the devices could reveal a wealth of useful context data regarding the nodes in the network and MSNs can greatly benefit from the presence of this additional information. For example, they could reveal mobility patterns of the users, permitting MSNs to selectively forward content to the nodes which maximise the probability to spread it in the network. This kind of information is clearly not available from OSNs, and deriving it from other sources is often too complex. Nevertheless, MSNs and OSNs are complementary since social contacts in the physical and in the cyber worlds are tightly connected. For this reason, social data derived from OSNs can reveal information regarding contacts in the physical world (as witnessed in [38]), and these data could be used to improve communication in MSNs. For example, people with a high frequency of contact with the user could have higher probability to be encountered in the future than acquaintances. Thus, these nodes could represent a stable communication channel with the user. On the other hand, meeting acquaintances could represent a unique opportunity for data exchange. Having this kind of information is clearly important for efficient data dissemination in the network. In addition, the use of sensor data can further enrich MSNs, providing a better characterisation of the surrounding environment and the situations in which the node is involved. For example, if the user is visiting a museum during an exhibition of Italian sculpture, she could be more interested in content regarding the specific works of art in the museum than in other kinds of information available form neighbour nodes. Sensor data could be obtained directly from the device, but also from external sensing services (e.g. proximity sensors in the museum indicating the distance from works of art) and from other nodes, in a cooperative fashion.

The main driver of communications in social networks is represented by human sociality. A complete understanding of the social properties of OSNs could thus improve the design of MSNs. In particular, since MSNs are built around the user, the structural properties of personal social networks, also called ego networks, could reveal important information regarding communication patterns, needs, and interests of the users that could substantially improve the effectiveness of MSNs. For example, users with a large number of acquaintances (i.e. weak ties), that have been identified in the literature as the main source of new information and ideas acquired from the network [57], could be more interested in a wider range of topics than people with strong social ties only.

Even though the properties of ego networks have been largely studied by sociologists and psychologists in offline social networks (i.e. not mediated by digital communications), there is no detailed information about their structure in OSNs. Specifically, it is not clear whether the use of OSNs is changing the properties of ego networks or, on the contrary, ego networks are independent from the use of a particular communication medium. As the structure of ego networks is important to determine key properties such as trust between people and willingness of collaboration, having a clear understanding about the structure of ego networks in OSNs is important for the design of communication solutions for MSNs.

Another strong limitation to MSNs is the lack of a common development platform for mobile devices. Without a common platform, each application must implement its communication protocols to permit data dissemination in opportunistic networking environments. Moreover, context and social data collected and elaborated by an application are not shared with other applications running on the same node, that could be interested in the same information. A common platform could enrich MSN services by providing a unified access to context and social data and to create a complete ecosystem that could foster the diffusion of MSNs.

1.1 Contributions

This thesis contributes to provide a support to the design and development of MSNs. On the one hand, OSNs are analysed to characterise the the structural properties of ego networks in online environments, with the aim to better understand the needs of MSNs. On the other hand, a novel social- and context-aware middleware platform for opportunistic MSNs is introduced to support the development of MSNs on mobile devices, giving a set of common APIs that provides access to opportunistic networking functionality and to context and social data management.

1.1.1 Analysis of the Structural Properties of Ego Networks in OSNs

This thesis provides a detailed characterisation of the structures of ego networks in OSNs, presenting a series of analyses on OSN data sets built from communication data collected from Facebook and Twitter.

The first analysis presented in this thesis delves into the properties of tie strength between individuals in OSNs. Tie strength represents the fundamental property of social relationships, which indicates their importance and their different roles in the network. The results of the analysis indicate that all the factors

composing tie strength in its classic definition (from sociology) are present also in OSN communication data. This means that tie strength can be derived from a combination of measurable variables obtainable from OSN data and this supports the possibility to study the structural properties of ego networks in OSNs. Specifically, it has been possible to create linear regression models able to estimate tie strength with accuracy higher than 80%, using combinations of only 4 variables describing different aspects of users interactions in Facebook. The findings also show that using the frequency of contact between users in Facebook alone provides a good tie strength estimation. Nevertheless the accuracy is higher when the frequency of contact is combined with other variables. This confirms previous studies in offline environments and represents a building block for all the subsequent analyses presented in this thesis, that use the frequency of contact (easily obtainable from OSNs) as a measure of tie strength to study online ego network properties.

The results of the other analyses presented in this thesis indicate that the structural properties of ego networks in OSNs are compatible with the results in the literature about offline social networks. Specifically, offline ego networks are formed of a series of inclusive concentric circles containing social contacts with different properties and characterised by typical size and frequency of contact with the ego. The main characteristic of this structure is that the scaling factor between the size of adjacent circles appears to be a constant close to 3. By applying cluster analysis on the frequencies of contact of each ego network extracted from Twitter and Facebook it has been possible to find that online ego networks have the same number of circles (i.e. 4) as offline ego networks. Moreover, the scaling factor between the circles is close to 3, as found offline. In addition, in Twitter ego networks there is the presence of an additional circle containing one or two persons tightly connected to the ego, with frequency of contact considerably higher than the others. This group, termed *super support clique*, could contain a partner and/or a best friend. This result confirms what was previously hypothesised in offline ego networks, but, for the lack of data, remained unsubstantiated hitherto.

These results show that, although the use of OSNs is changing the way people create and maintain social relationships with each other, the size and composition of ego networks, that are indeed shaped by the constrained nature of human brain, remain unaltered. The results are consistent amongst different OSNs.

After the analyses related to the static structure of ego networks in OSNs, this thesis presents a study about the dynamic evolution of ego networks (in terms of size and composition) in Twitter. The nature of the collected Twitter data set, containing the whole history of communication of a high number of users, helped

to discover that ego networks in Twitter, despite maintaining structural properties similar to offline ego networks, are affected by more turnover. In other words, Twitter users constantly add new contacts in their ego networks, abandoning some of their old friends, and this process appears to be more marked than in offline environments. Specifically, the average percentage of turnover in Twitter ego networks is about 75%. Only a small fraction of users in Twitter (5.4% of the users considered in the analysis) show values of turnover in their ego networks compatible with findings in offline social networks. This fact suggests that the general behaviour of Twitter users is to maintain a light-weight ego network formed of weak social relationships suitable for maximising the amount of resources accessible through the network. From the analysis it has also been possible to identify three different categories of users in Twitter: (i) occasional users, with an initial phase of very high activity followed by sudden decay or abandonment; (ii) regular users, with a more regular pattern of activity over time; (iii) aficionados, characterised by slow start in terms of activity followed by a gradual increase and longevous ego networks.

1.1.2 CAMEO - A Novel Middleware Platform for Opportunistic Mobile Social Networks

As a step towards a better support for MSNs, this thesis presents a novel software platform called Context Aware MiddlEware for Opportunistic Mobile Social Networks (CAMEO). CAMEO enables direct communications between devices through Bluetooth and Wi-Fi interfaces and gives a set of APIs to access opportunistic networking functionality. Moreover, CAMEO collects and elaborates context information of the local node (local context), the environments surrounding it (external context), and the context of other nodes encountered in the network (social context). Context data are used to discover common interests, habits or needs amongst the users. Based on these commonalities, CAMEO suggests possible interesting contents available from neighbour nodes to the users and to the applications. The users are stimulated to communicate with each other to obtain useful information, to discuss about common interests, and so forth. CAMEO has been provided with a context model able to encompass all the possible situations in which the node could be.

Context data have been enriched with sensing information coming from external sources, by the integration in CAMEO of Sensor Web Enablement (SWE) standards, which permit the access of external sensors and sensor repositories in in a standard way through the Web. To further improve the elaboration of context data on mobile devices, a new version of SWE data format for mobile devices (called Sensor Mobile Enablement - SME) has been defined and implemented in

CAMEO. SME overcomes the limitations of SWE, that is designed for the Web and is not suited for mobile devices. Experimental results indicate that SME data format improves content density (the ratio between the value size of XML attributes and elements) of the XML sensor data codifications by more than 80% compared to SWE format. In addition, the time needed to serialise and deserialise SME files is in the order of a few seconds for up to 150 sensor descriptions and observations. These results show that CAMEO with SME is able to manage the concurrent reception and elaboration of a sufficiently high number of sensor data from external sources.

CAMEO augments the social experience of the users by providing personalised and geo-located content, without the need to aggregate information in a centralised server. The efficiency of CAMEO is showed by several experimental results that indicate that it clearly outperforms other middleware platforms for the creation of context-aware applications, also providing additional functionality specifically defined for MSNs. In this thesis, two prototypes of MSNs are presented, one aimed at enriching the social experience of tourists during their visits and the other aimed at enlarging the social-awareness of sportsmen during their physical activities. These prototypes demonstrate the ease in the use of CAMEO APIs and the potential of the middleware.

1.2 Thesis Organisation

In Chapter 2, the analyses aimed at characterising the structural properties of ego networks in online environments are presented. Then, Chapter 3 presents the paradigm of context- and social-aware mobile social networking and illustrates CAMEO architecture, along with the description of the new data encoding format (SME) used by CAMEO to efficiently interact with external context data sources. Moreover, real examples of MSN applications are presented to show the potential of CAMEO. Chapter 4 presents the main conclusion of this thesis.

2

Structural Properties of Ego Networks in OSNs

This chapter presents a detailed analysis of the structural properties of ego networks in OSNs aimed at characterising online social environments and comparing the properties of personal social networks in such environments with those formed offline. Before presenting the analysis of OSN data, the main results found in offline ego networks and the established properties of OSNs are reported in an introductory section.

2.1 Background Work in the Literature and Definitions

To be able to characterise OSNs and to compare their properties with well-known findings in the literature, it is essential to make a clear distinction between *online* and *offline* social networks. Whilst the former are mediated by the use of digital communications, the latter are based on more traditional communication means, such as face-to-face communications. For the sake of clarity, in this thesis the term OSNs is used only for describing social networks formed by social media, without considering other possible social channels (e.g. e-mails), whose properties have already been studied in the literature. The difference between online and offline social networks is thus determined by the nature of the communication medium. This intuitive distinction between the two types of networks permitted to assess how human social behaviour adapts to different social environments, as well as to describe the structural differences of ego networks offline and online.

2.1.1 Offline Social Networks

Most of the work done in the literature to describe the properties of social networks has been carried out in offline environments, since OSNs appeared recently and research in the field of OSN analysis is relatively new.

Collecting data concerning offline social networks is a rather difficult task, since reconstructing the history of face-to-face communications between people requires the involved participants to recollect facts about their past social contacts. This clearly implies that information regarding best friends or people contacted more frequently is more accurate than that concerning people met a long time before the data collection. For this reason, manually collecting data about entire social networks is often infeasible, and research on offline social networks is thus focused on personal social networks of single individuals, called ego networks.

Ego Networks

An ego network is a simple social network model formed of an individual (called ego) and all the persons with whom the ego has a social link (alters). Ego networks are useful to study the properties of human social behaviour at a personal level, and to assess the extent to which individual characteristics of the ego affect the size and the composition of their network. The most important result found on ego networks is that the cognitive constraints of human brain and the limited time that a person can use for socialising bound the number of social relationships that she can actively maintain in her network. This limit lies, on average, around 150 and is known as the *Dunbar's number*, from its first discoverer, Robin Dunbar, a British evolutionary psychologist. Dunbar found a positive correlation between primates' brain size and the average size of their social groups [49]. Hence, he hypothesised that, since humans have larger brains compared to primates, they should have an average group size of 150. This result has been further confirmed by several experiments, and the Dunbar's number has been empirically estimated to a value equal to 132.5 [105].

Inside their social groups, humans form small coalitions with other individuals to reduce the frequency of aggression or harassment [88] and to reduce thus the cost of group living. This technique is used at different levels, from small groups of one or two strong allies, to larger groups of people sharing the same interests or ideas, leading to the formation of a typical hierarchical structure of a series of sub-groupings arranged in an inclusive sequence, that in human ego networks is typically formed of four or five layers. An individual ego can be envisaged as sitting at the centre of a series of concentric circles of alters ordered by the strength of

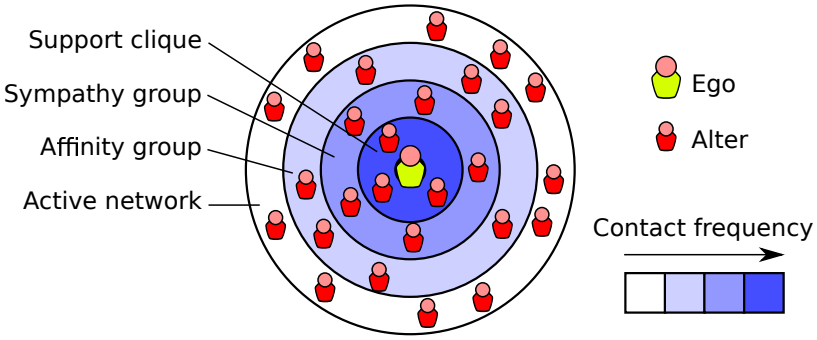


Figure 2.1: The Ego Network Model.

their social ties [82], as depicted in Figure 2.1. Each of these circles has typical size and frequency of contact between the ego and the alters contained in it. The first circle, called *support clique*, contains alters with very strong social relationships with the ego, informally identified in literature as *best friends*. These alters are people contacted by the ego in case of a strong emotional distress or financial disasters. The size of this circle is limited, on average, to 5 members, usually contacted by the ego at least once a week. The second circle, called *sympathy group*, contains alters who can be identified as *close friends*. This circle contains on average 15 members contacted by the ego at least once a month. The next circle is the *affinity group* (or *band* in the ethnographic literature), which contains 50 alters usually representing causal friends or extended family members [83]. Although some studies tried to identify the typical frequency of contact of this circle, there are no accurate results in the literature about its properties, due to the difficulties related to the manual collection of data about the alters contained in it through interviews or surveys. Indeed, people hardly remember their social contacts besides their best and close friends. The last circle in the ego network model is the *active network*, which includes all the other circles, for a total of 150 members. This circle is bounded by the limit of the Dunbar's number and contains people for whom the ego actively invests a non-negligible amount of resources to maintain the related social relationships over time. People in the active network are contacted, by definition, at least once a year. Alters beyond the active network are considered inactive, since they are not contacted regularly by the ego. These alters are grouped in additional external circles called *mega-bands* and *large tribes*. One of the most important properties of ego network's circular structure is that the ratio between the size of adjacent circles appears to be a constant with a value around 3. For

a complete discussion about the properties of these circles we refer the reader to [88].

The Importance of Social Tie Strength

A milestone in social network analysis has been reached thanks to Mark Granovetter's work [57]. The American sociologist discovered that tie strength (i.e. the importance of a social tie), informally defined as a (probably linear) combination of time, emotional intensity, intimacy, and reciprocal services which characterise the tie, determines the functional properties of a social relationship. Social ties can be broadly divided in two categories: *strong* and *weak* ties. The former are related to a small set of intimate friends and are useful to consolidate a core group of trusted people on whom an individual can count in case of troubles. On the other hand, weak ties are acquaintances, socially far from the ego and usually included within different social milieus. This distinction can be seen as a broader outline of the ego network circles, with only two categories. Granovetter found that, despite their low strength, weak ties are important for the ego to access resources from other social groups, and their total strength exceeds that of strong ties. Granovetter's findings indicate that tie strength must be taken into account to fully understand social aspects of a social network.

Despite this, tie strength is not directly measurable, since the factors composing it are influenced by emotions. Nevertheless, Peter Marsden demonstrated the feasibility of constructing measures of tie strength through multiple indicator techniques [72]. Marsden built an analytic model to explain the relation between a set of tie strength predictors (i.e. aspects of relationships that are related to, but not components of, tie strength) and tie strength indicators (emotional closeness, duration, frequency of contact, breadth of discussion topics, and confiding). The results of his analysis demonstrated that emotional closeness (or emotional intensity) is the best indicator of the strength of a social relationship. Moreover, measures of the time spent in a relationship (e.g. frequency of contact and duration) are related to the concept, even though they tend to systematically overestimate tie strength in case the involved persons are co-workers or neighbours. These results indicate that tie strength can be effectively estimated using some measurable indicators. In the ego network model both measures of emotional closeness and frequency of contact are used to describe the different circles.

This thesis presents a study of tie strength in OSNs. Specifically, measures of tie strength are constructed using variables obtained from OSN data. The results indicate that, as found by Peter Marsden, the frequency of contact is the best

indicator of tie strength also in OSNs, but using additional factors could improve the accuracy of the measure. Nevertheless, since the frequency of contact is often the only measure of tie strength available from social media, in the rest of the analyses presented in this thesis it has been used to estimate tie strength and to study ego networks structural properties in OSNs.

Macro-Level Structural Properties

Seen from a macro-level perspective, social networks show some typical properties that have been observed in many different environments. Stanley Milgram, through his famous experiment, demonstrated the presence of the so called *small-world effect* in social networks [95]. According to this property, any two persons in the network, indirectly connected by chains of social links, have a short average distance. This is often identified as the *six degrees of separation* theory, for which everyone in a social network is six steps away. This fact directly influences the ability of the network to quickly spread information, ideas, innovations and so forth. It has been demonstrated that the diffusion of information in social networks takes place through single social links, creating the *word-of-mouth* effect. This property has been largely used by a collection of marketing techniques whereby the presence of social links between consumers is exploited to increase sales [62].

Other distinctive properties of social networks, that differentiate them from other types of networks, including technological and biological networks, are represented by the presence of a non-trivial clustering or network transitivity, that is, in other words, a high probability that two neighbours connected to a node will also be connected to each other. Moreover, social networks show positive correlations between the degrees of adjacent vertices, also called *assortativity* [76].

2.1.2 Online Social Networks

As far as OSNs are concerned, social media have generated a wealth of data containing the whole communication history between their users. This fostered analyses about the macro-level properties of the whole networks. The availability of OSNs communication data revealed the presence of some distinctive social traits also in online environments. Specifically, the small world effect has been found also in online environments [71, 48]. In [74] the authors presented a detailed analysis of the macro-level structural properties of a set of different OSNs, finding results in accordance with the properties of whole social networks observed in offline environments.

The different roles of weak and strong ties has been confirmed in [77], revealing a relation between the frequency of contact and the presence of local structures in the network. Moreover, the authors found that social networks are robust to the removal of strong ties, but fall apart after the removal of a sufficient number of weak ties.

Although a large body of work has been done to characterise OSNs, most of the analyses have been performed on unweighted social graphs (see for example [96]), without considering the strength of social ties. This is due to the hardness of collecting information about social interactions between people in very large social networks. Nevertheless, in [101] the authors demonstrated that there is a significant difference between the properties of weighted and unweighted graphs representing the same social networks. In addition, in [56] the unweighted social graph extracted from publicly available data on Google+ has been augmented with four nodes' attributes (i.e. school, major, employer and city). The results confirm that in some cases the network of attributes shows properties significantly different from the unweighted network.

Tie Strength and Online Social Networks

The possibility to deduce social tie strength from OSN data has been proved in [54]. Specifically, the authors used a Facebook data set and explicit evaluation of tie strength done by the users. Interaction variables have been used to fit the explicit evaluation. A linear regression model has been used to fit values of tie strength collected from questionnaires. The authors of [64] presented a study aimed at predicting tie strength from online interactions. They asked a set of participants to indicate the name of their close friends. Hence, they used the collected evaluations to train a classifier to distinguish between strong and weak ties. The classifier gives a membership probability calculated from a set of online interaction variables. This probability represents a prediction of tie strength. Since the proposed model is based on evaluations of close friendships only, it is less accurate in the prediction of weak ties. This represents a strong limitation, since weak ties form about $60 - 80\%$ of an ego network [85].

Preliminary results on Ego Networks in OSNs

In [55] the authors analysed a large-scale data set of Twitter communication data, finding that the average intensity of communication of each user towards all her friends, as a function of the number of social contacts of the user, shows an asymptotic behaviour, ascribable to the limits imposed by the Dunbar's number. In [67],

the authors demonstrated that inter-individual variability in the number of social relationships in online social networks is correlated with brain size. The authors used magnetic resonance imaging techniques to measure grey matter density of a small sample of participants, comparing the brain volume of the participants the number of their Facebook contacts.

In [85], the authors analysed online social network data of a sample of thirty participants discovering that each ego shows a typical tie strength distribution within their ego network. This distribution is in accordance with the ego network model. In [73], mobile-phone data extracted from the logs of a single mobile phone operator has been analysed. The results indicate that the limited capacity people have for communicaton limits the amount of social ties they can actively maintain.

Although these results give a first insight on the constrained nature of online social networks, revealing a similarity between online and offline human social behaviour, there is still a lack of knowledge about all the other ego network structural properties of OSNs. Specifically, it is not clear if structures similar to those described by the ego network model could be found also in OSNs.

The aim of this thesis is to bridge this gap providing a solid analysis of the ego network structures in OSNs.

2.2 Egocentric Online Social Networks: Analysis of Key Features and Prediction of Tie Strength in Facebook

This section presents a detailed analysis of a real Facebook data set collected from a sample of 30 participants aimed at characterising the properties of human social relationships in online environments [27]. The results represent a first indication that the properties of OSNs appear to be similar to those found offline. Specifically, on Facebook there is a limited number of social relationships an individual can actively maintain and this number is close to the Dunbar's number (150) found in offline social networks.

This section also presents a number of linear models used to predict tie strength from a reduced set of observable Facebook variables. Specifically, it has been possible to predict with good accuracy (i.e. higher than 80%) the strength of social ties by exploiting only 4 variables describing different aspects of users interaction on Facebook. The recency of contact between individuals - used in other studies as the unique estimator of tie strength and that is directly related to the frequency of contact - showed the highest relevance in the prediction of tie strength. Nevertheless, using the frequency of contact in combination with other observable quantities (in case they could be used), such as indices about the social similarity

between people, can lead to more accurate predictions. These results are fundamental for the analyses presented in the other sections of this thesis, which use the frequency of contact between users in OSNs to estimate their tie strength.

Facebook Data Collection Process and Data Set Description

Although Facebook generates a huge amount of data regarding social communications between people, obtaining these data is not easy. In fact, publicly available data have been strongly limited by the introduction of strict privacy policies and default settings for the users in 2009. To collect data From Facebook, a Facebook application able to download all the data related to the logged-in users, called Facebook Analyser (FBA), has been developed. FBA collects all the data obtainable from Facebook, including socio-demographic variables (i.e. related to the user) and relational variables (i.e. related to the social relationships the user has with other people). FBA also collects values of tie strength manually evaluated by the users towards their social contacts (Facebook friends). To this aim FBA asks the users to manually rate their friendships by answering the following question: *"How do you rate, with a value between 0 and 100, the social relationship between you and this person in Facebook?"*. Using a generic question it has been possible to capture the most generic definition of tie strength. The question is also supported by additional context information to make the users effectively express their perception of tie strength. Given the limited number of users involved in the study, it was possible to explain to them in detail the background of the study, and the purpose of the application. They were made aware of the concept of tie strength, of different types of social relationship and the typical way used in the anthropology literature to quantify it. Typical questions used in Dunbar's studies have been used as examples of how they should have evaluated social relationships. The qualifier "in Facebook" was also clearly explained to them. Specifically, they were asked to evaluate social relationships considering only their activity and interactions in Facebook, thus disregarding any other interactions occurring with their friends offline. Finally, a numerical range between 0 and 100 has been selected since it proved to be a natural evaluation scale, once the context of the evaluation had been made clear to the users.

Currently, a new version of FBA with a simplified tie strength evaluation method is under development. The improved graphical user interface is based on a ruler with tics between 0 and 100 on which pictures of the social contacts can be placed to perform tie strength evaluations. The new application is currently being used to collect a larger sample of participants and to continue the analysis reported in this

section. A preliminary version of the new application, called "Ego-Net Digger" is reported in [53].

Although FBA and Ego-Net Digger permit the download of detailed data regarding the users and their ego networks, the number of users' profiles that can be downloaded depends on the number of participants involved in the experiments. To test the results found in this section, additional analyses have been performed on an additional data set collected from Twitter, described in Section 2.3. This data set, compared to the one presented here, contains a higher number of user profiles, but with less detailed information. Combining the studies presented here and in the following sections, this thesis presents a complete view of the properties of OSN ego networks, both from a fine-grained analysis of a small, but detailed, sample of egos, and from large-scale studies on massive data sets. This combination led to the validation of the results at different scales and on different social media.

The data acquisition campaign performed to download Facebook data using FBA involved 30 people, who were asked to use the application and to rate all their Facebook friendships. A total number of $7,665$ relationships has been collected, from which $3,245$ active friendships (the definition of active social relationship used in this analysis will be given later in this section) have been extracted. Whilst the number of sampled social relationships is significant, the number of users involved in the experiment is not sufficient to draw definite conclusions. However, this sample is already sufficient to provide interesting indications on the properties of OSN ego networks, and their similarity with ego networks observed in offline social networks.

To study the properties of the ego networks of the downloaded users, two sets of variables have been identified. The first set contains all the variables related to the users' profiles, describing properties of the ego. These variables are called *socio-demographic* variables. On the other hand, variables describing the relationships between users in Facebook are called *relational* variables.

The socio-demographic and relational variables selected for the analysis are listed in Table 2.1. The rationale was to select a rather broad set of variables describing overall properties of ego networks and the interactions between egos and their alters, intuitively related with the properties of tie strength, and then use statistical analysis to identify the variables that better describe and predict it. For the sake of clarity, the collected variables have been divided into two distinct groups, treating them separately in the next phases of the analysis. In addition to the quantities concerning ego's characteristics, the first group contains also the total amount and the mean values of each *relational* variable (e.g. total and mean number of messages, posts and other quantities received or sent by ego). These

Table 2.1: Facebook variables chosen as possible descriptors of ego networks characteristics.

Socio-demographic Variables
gender
number of friends
total number of status updates
sum of each relational variable - all alters
mean value of each relational variable - all alters
mean value of each relational variable - active alters
Relational Variables
number of likes[2]
number of posts[2]
number of comments[2]
number of private messages[2]
number of tags on the same pictures
number of days since first communication[3]
number of days since last communication[3]
number of events attended together
number of groups in common
number of likes on the same fan pages
frequency of contact[2]

variables should contribute to describe the behaviour of a user in Facebook. For example, the mean number of comments sent per alter can be seen as a descriptor of how much a person uses Facebook (i.e. the Facebook use rate).

All the user-filled fields available on Facebook profiles (e.g. political view, religion, hometown, education) have been excluded from the analysis. This choice was mainly driven by the fact that many of these fields are intentionally left blank by the users. Moreover, the information contained within these variables could be difficult to be correctly interpreted through automatic tools, for people are prone to use sarcastic phrases or provoking words that cannot be easily categorised. The idea is that using a simple good/bad words count (as done in [54]) is not enough to fully interpret people's social attitudes in OSNs. Extracting information that can help to assess the type of social relationships from this fields also require to take care of cultural aspects and differences between the users, discern humorous and paradoxical expressions and much more. Whilst this is an exciting subject

[2] From ego to each alter and vice-versa.

[3] From ego to each alter and vice-versa divided for each type of communication (likes, posts, comments, messages).

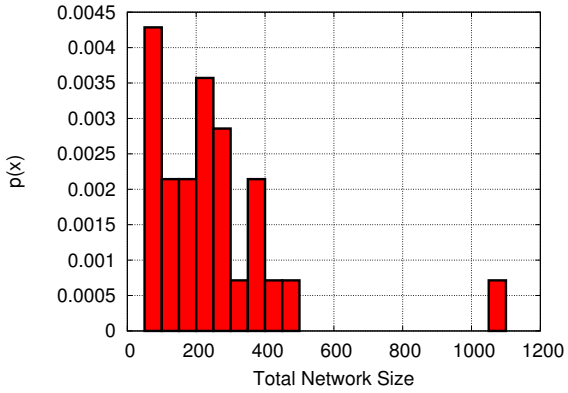


Figure 2.2: Distribution of the number of Facebook friends

to explore, it has not been included in the current analysis, and is left for future improvements. Focusing exclusively on quantitative measurable variables, which do not require particularly refined interpretation, led to a reduction in terms of complexity of the analysis, understanding the extent to which those variables alone can be used to predict tie strength, without having to deal with possible inaccuracies and errors in interpreting the real semantic of user-filled fields. For sure, it can be anticipated that including also this information - after a correct processing - will further increase the prediction accuracy of the presented models.

The collected evaluations of tie strength have been used as “ground truth” to compare and calibrate the tie strength prediction models. Note that the ultimate goal of the prediction models is to avoid to ask explicit tie strength values to the users. Yet, in order to calibrate the models, tie strength evaluations are essential in this phase.

From the data set of 30 participants, which is the same used in [28], two participants have been discarded, because they provided only a partial evaluation of the tie strength related to their friendships (i.e. they have not completed the survey). Hence, the data set analysed in this work is composed of 28 participants and $7,103$ social relationships. For the figures previously analysed in [28], the refinement introduced discarding the data from the other two participants led to more significant and reliable results.

Although this data set needs to be enlarged in terms of number of users, it already contains a significant number of samples of social relationships to start deriving interesting results, both for ego network analysis and for tie strength pre-

diction. In particular, this data set permitted a sensible analysis about the factors that characterise the virtual relationships, and a well grounded regression analysis to estimate the social tie strength.

For the purpose of the analysis of tie strength prediction presented in this section, it is worth noting that the participants with more friends could possibly have a higher impact on the determination of the coefficients of the models. To avoid this limitation, the data set has been sub-sampled 100 times, randomly extracting the same number of friends (100) for each participant. As far as the participants with less than 100 friends are concerned, all their social relationships are considered. Hence, 100 data sets have been used in the analysis concerning the prediction models. Note that for the analysis of the ego network properties no sub-sampling has been used. This analysis looks, amongst other, to overall properties of ego networks, such as their size, and thus sub-sampling would have significantly biased the results.

Ego Networks Properties

To assess the properties of Facebook ego networks in the sample and to compare them with the characteristics found in offline ego networks, the descriptive statistics of the data have been analysed. This analysis is divided in two different parts, the first related to *socio-demographic* variables and the second concerning *relational* variables. Note that each user, along with her Facebook friends and their respective social relationships, forms a separate ego network. For this reason, each statistic presented in the following has been calculated for each user and then averaged for all the users, to assess the average properties of the ego networks in the sample.

Socio-Demographic Variables

As far as *socio-demographic* variables, the 28 participants within the sample are researchers, Ph.D students or master students from 24 to 48 years old (M $= 32.86$, SD $= 6.77$), 15 males and 13 females. The number of friends of each participants ranges between 86 and 1099 (M $= 253.68$, SD $= 204.14$). The distribution of this variable is shown in figure 2.2. In the figure, an outlier can be clearly identified, but it has not been discarded from the analysis, since from now on only the *active* part of the networks are considered, and in such part the mentioned ego is not an outlier. To distinguish between active and inactive social relationships in the analysis, the “active network” is defined as the set of friends for whom the value of tie strength indicated by the user is greater than 0. This definition differs from

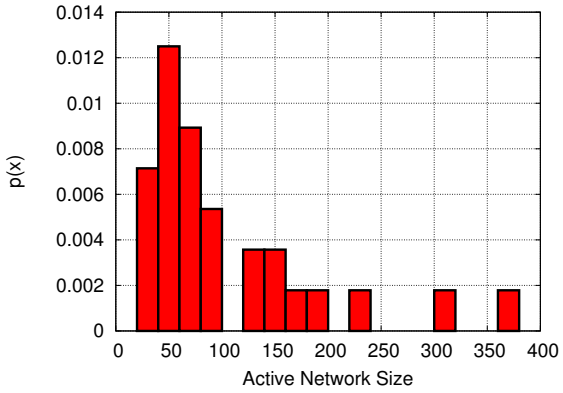


Figure 2.3: Active network size distribution.

that used in the literature about offline social networks and reported in Section 2.1, which is based on the frequency of contact (formally the set of people contacted at least once a year). This definition has been preferred in the present analysis as in the data set many more variables other than frequency of contacts are considered. Using the explicit evaluation of the users to discriminate active from inactive relationships is a way to compactly consider all variables altogether, without giving higher importance to any of them a priori. Note that this methodology was explained to the users, so that they knew that giving a score of 0 to a relationship would mean marking it as inactive.

Active network sizes in the sample range between 29 and 368 (M $= 105.14$, SD $= 85.42$). Users have, on average, 45.88% of their Facebook friends that can be considered active, with a 95% confidence interval equal to $(38.99\%, 54.77\%)$. The distribution of active network size in the sample, depicted in figure 2.3, is qualitatively similar to those found in other work about offline social networks [83, 61]. Moreover, the mean active network size is also comparable to the same measure in offline social networks (e.g. 124 in [61] and 132.5 in [105]). This suggests that a maximum number of active relationships in the order of the Dunbar's number can be also found in OSNs.

Figure 2.4 depicts the distribution of tie strength in the sample, considering active networks only. The figure also shows the tie strength density for each ego network divided in ten different bins of ten units of tie strength each, then averaged for all the ego networks. The shape of the tie strength distribution indicates the presence of a small set of alters tightly connected to ego and a larger number

of people loosely coupled with her. This is in accordance with the findings about offline ego networks [49, 82, 57].

Relational Variables

This section presents the descriptive statistics of the *relational* variables listed in Table 2.1. For each user, the total amount and the mean values of these variables have been considered, to describe the behaviour of egos in terms of the amount of information they exchange or they have in common with others. These quantities have been calculated both for all Facebook friends of a user and also for active friends only. Whilst all the other variables are self-explanatory, the concept of "like" needs to be discussed before continuing with variables description. Like-based communication relies on the "like" mechanism. Likes are a special kind of marks left on Facebook objects (e.g. pictures, comments, status updates), used to give a favourable feedback towards these objects.

For the sake of readability, the complete statistics of the relational variables have been omitted from this thesis. Nevertheless, the reader can find them in [27]. The sampled users make, on average, broadly the same number of comments and likes in Facebook. This is in accordance with the results in [58] and it highlights the growing importance of new kind of communications in OSNs. The average number of posts sent by egos is higher than the number of likes and comments. The *number of days since first outgoing/incoming communication* gives an estimation of the mean duration of the considered social relationships. This duration is, on average, between 3 years and 284 days and 3 years and 175 days. This result indicates

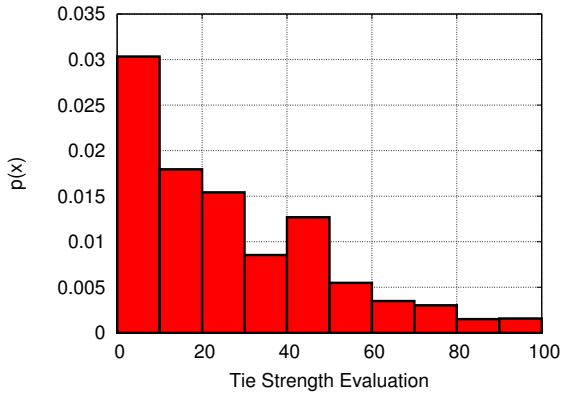


Figure 2.4: Tie strength distribution.

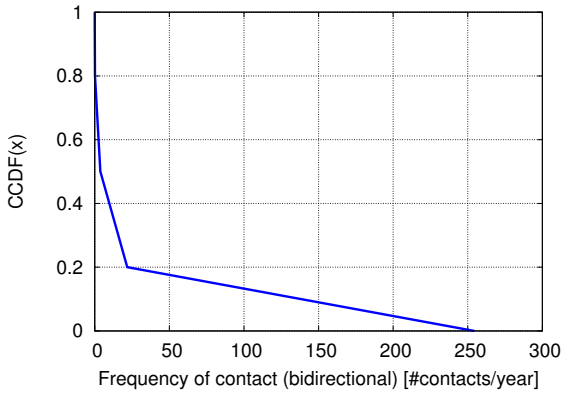


Figure 2.5: CCDF of the Frequency of Contact (bidirectional) Between Ego and Alters.

that the temporal window considered in the analysis is sufficiently large. The *time since last outgoing/incoming* communication indicates how recently, on average, people have been contacted on Facebook by their friends and this measure will be referred to as the *recency of communication*. This measure has been used in the literature as an estimation of tie strength and, as we will see in the following, this variable proves to be a good prediction of the frequency of contact between people, playing a central role in the prediction of tie strength.

The analysed statistics indicate that all the variables representing the incoming communication received by ego from alters (e.g. the number of likes, comments, posts) take values considerably higher than the variables concerning the outgoing communication made by ego to alters. This is in accordance with the findings in [16], where the authors indicate that in Facebook a person has a limited number of friends with whom she directly communicates and a much larger portion of people from whom she only passively receive and consume information, without reciprocating their interactions. The incoming and the outgoing communication could thus have different roles in the prediction of tie strength. This result will be verified when the prediction models will be presented.

As expected, the statistics concerning the active network have always greater values than those calculated on the entire ego network (apart from the *time since last contact*). This confirms that between the ego and her active friends - defined by the tie strength - there is much more activity on Facebook than between the ego and the entire set of her friendships.

To extract additional information from the relational variables, their complementary cumulative distribution functions (hereinafter CCDF) have been analysed. The typical pattern found is that of a long tail shape. In Figure 2.5 one of these CCDFs is reported, related to the frequency of contact. The plots for the other variables are similar, and can be found in [27]. The percentiles indicated in the distribution have been obtained for each user, and averaged over all users. The frequency of contact shows a distribution similar to that found in offline ego networks [61] and the shapes of these variables are similar to the one of the tie strength in Figure 2.4. This is an initial indication that this set of variables could be effectively used to predict tie strength.

Facebook variables' long tail shape indicates the presence of a large set of friends with whom egos have little communication or with whom they have little things in common (e.g. groups, pictures), and a small set of alters with whom egos have a strong interaction. The "elbow" that can be noticed in the curve depicted in Figure 2.5 indicates a clear distinction between these groups of "weak" and "strong" ties. Analysing the sizes of these groups, calculated for each distribution of the variables in the data set, the percentage of strong ties in the ego networks in the sample is, on average, 23.53% of the total number of social relationships (i.e. 59.69 over 253.68) considering all the interactions, and 40.09% of the total number of active relationships (i.e. 42.15 over 105.14). Under the hypothesis (verified in the next sections of this thesis) that structures similar to those found in offline social networks are also present in online social networks, this result indicates that relational variables would discriminate relationships in the external part of the active network from the stronger ones, in the more internal layers of ego networks (the first three layers).

Models for the Prediction of Tie Strength

This section presents a set of tie strength prediction models built using the Facebook data set collected using FBA and the respective tie strength evaluations. Before introducing the models, a preliminary correlation analysis between the relational variables and the tie strength explicit values is presented. The rationale of each modelling approach is presented at the beginning of the corresponding sections. A linear approach has been adopted, since the aim of this analysis is to maintain the models as simple as possible to validate the use of simple estimations of tie strength with variables obtainable from social media (e.g. Facebook, Twitter), that are adopted in the rest of the analyses presented in this thesis. In addition, the choice has been motivated by the background work in sociology, where tie strength is considered to be a mostly linear combination of social factors.

All the steps taken during this part of the analysis required all the variables to be normally distributed and standardised. Thus, they have been log-transformed in case they showed absolute values of skewness and kurtosis greater than 1 [40] and they have been standardised.

As already pointed out during the data set description, the data set has been divided into 100 different sets, by sub-sampling 100 relationships from each of the 28 egos in the data set (and including all relationships for egos with less than 100 friendships). This is to avoid that people with more Facebook friends affected the results of the analysis. Therefore, to obtain average results from the different data sets, the techniques described in the following have been applied to each sub-sampled data set, then averaging the obtained results. This permitted the study of the average correlation of each Facebook variable with respect to tie strength and also to obtain statistically solid regression models, that can then be used to predict tie strength.

To train and validate the accuracy of the models a training and test set have been defined out of the data set as follows. Each of the 100 sub-sampled data sets have been split into a training set containing 23 randomly selected ego networks and a test set with the remaining 5 ego networks. To prevent the results to be influenced by a particular combination of these sets, five different pairs of training and test sets have been created for each of the 100 sub-sampled data sets. Then, a regression model has been fitted for each of the resulting 500 training sets (formed of 100 sub-sampled data set for each of the 5 different combinations of training sets) and an overall model has been derived by averaging the coefficient of all the obtained 500 linear regression models. Hence, the accuracy of the obtained model has been evaluated by applying it on the test sets, making a comparison of the output of the model and the explicit evaluations of tie strength contained in the test sets. The accuracy results presented in the following have been averaged over the 500 test sets (100 sub-samples for each of the 5 combinations of test sets).

Since the data set contains some variables that could be highly correlated between each other (e.g. the number of posts sent by ego and the number of comments received from alters and many others), linear regression could be affected by multicollinearity. Multicollinearity represents a near exact relationship between two or more variables [84], which can impact on the accuracy and correctness of the regression model. Specifically, linear regression could force the sign of the regression coefficients to be different from the sign of the correlation between the respective variables and tie strength, invalidating the correctness of the results. To avoid this problem two different approach have been taken. On the one hand, the correlation between all the combinations of pairs of variables has been calculated

Table 2.2: Correlation between Facebook variables and tie strength.

#	Variable	r
1	Number of days since last comm.	$-.56$
2	Bidirectional frequency of contact	$.55$
3	Number of days since first comm.	$.51$
4	Frequency of incoming comm.	$.50$
5	Number of received comments	$.47$
6	Frequency of outgoing comm.	$.44$
7	Number of comments sent	$.43$
8	Number of received posts	$.41$
9	Number of received private msg	$.34$
10	Number of posts sent	$.33$
11	Number of likes sent	$.32$
12	Number of received likes	$.29$
13	Number of alters' pictures in which ego appears	$.24$
14	Number of fan pages in common	$.20$
15	Number of tags on the same objects	$.20$
16	Number of groups in common	$.20$
17	Number of ego's pictures in which alters appear	$.17$
18	Number of events in common	$.14$
19	Number of private msg sent	$.11$

and a regression model has been created using uncorrelated variables only (thus excluding the sources of multicollinearity)[1]. On the other hand, Principal Component Analysis (PCA) has been used to extract a set of uncorrelated factors from the data set and the latter are used to create a tie strength predictive model. The results obtained in the two cases are presented in the following Sections, after presenting the initial correlation analysis.

Correlation Between Facebook Variables and Tie Strength

The correlation between each variable in the data set and the evaluations of tie strength provided by the users has been studied using the Pearson product-moment correlation coefficient, described in greater detail in [27].

For *socio-demographic* variables, the correlation analysis indicates that the average tie strength of each ego network is significantly correlated with the mean bidirectional frequency of contact $(r = .474, p < .01)$, the mean number of comments made by ego to her alters $(r = .418, p < .05)$, the mean number of days since last communication from ego to alters $(r = -.485, p < .01)$, the mean number of days since first communication from ego to her alters $(r = .376, p < .05)$, the mean number of days since last communication received by ego $(r = -.473, p < .05)$, the mean number of likes made by ego to her alters $(r = .476, p < .05)$ and the mean number of groups in common between ego and alters $(r = .379, p < .05)$. In the sample, age does not influence tie strength. This result, in contrast with [50], could be explained by the fact that the sample is rather homogeneous, with a narrow age difference, which could not be enough to catch the influence of age on social relationships.

As far as the *relational* variables are concerned, their correlation with tie strength has been calculated for the 100 different sub-sampled data sets, averaging the obtained values for all the different data sets. The correlation values, ordered from the highest to the lowest, are reported in Table 2.2. In the table the p-values related to the correlation are omitted, since they all satisfy $p. < .01$. The variables showing the strongest correlation with tie strength are the *number of days since last communication*, the *frequency of contact* (both bidirectional and related to incoming interactions only) and the *number of days since first communication*. The first of these variables, representing the recency of communication, has been used in previous work as an estimator of the frequency of contact between individuals and as a tie strength estimate [61]. The correlation between Facebook variables and tie strength provides a first indication of the feasibility of the creation of a tie strength prediction model.

Table 2.3: Coefficients of the regression models based on uncorrelated variables.

Variable	estim.	std err.	p value
Model with one regressor			
Intercept	13.168	.376	$< .01$
1	-10.957	.375	$< .01$
Model without pairwise products			
Intercept	13.120	.357	$< .01$
1	-9.004	.379	$< .01$
11	3.798	.373	$< .01$
13	3.394	.384	$< .01$
15	.784	.388	$< .01$
Model with pairwise products			
Intercept	13.192	.376	$< .01$
1	-8.900	.380	$< .01$
11	4.317	.506	$< .01$
13	3.904	.567	$< .01$
15	.254	.419	$< .05$
$1 * 13$	$-.621$	.381	$< .05$
$1 * 15$	$-.326$	.226	$< .05$
$11 * 13$	.197	.229	$< .05$
$11 * 15$	.161	.136	$< .01$

Model With Uncorrelated Variables

The first family of models that has been created to predict tie strength and to describe its composition is based on a set of uncorrelated regressors. To build these model the correlation between all the possible combinations of pairs of variables has been firstly calculated and a set of regressors has been built following an iterative procedure. This procedure starts with an empty set of regressors, called R_t, where t indicates the maximum value of correlation any two variables within R_t can have. Hence, one variable at a time is taken from those listed in Table 2.2, according to a descending order - from the most correlated to tie strength to the less correlated - and it is added to R_t if all the correlation values it has with the other regressors already present in R_t are lower than t. Note that when t is equal to 1 all variables are in R_t irrespective of their mutual correlation, while t equal to

[1] Using correlation to select the regressors in the models led to results that can be easily interpreted and reduced as much as possible the number of regressors of the models. Nevertheless, stepwise regression was also used to select the best combination of regressors in the models and the accuracy of the obtained models is of the same order of that found using correlation.

0 would result in having in R_t only the variable with the highest correlation with tie strength (i.e. only the variable that is first introduced in R_t). The procedure is iterated until all the variables are processed. Thus, R_t represents a set of uncorrelated regressors at a certain level of pairwise correlation t. For high values of t, variables in R_t are more likely to present multicollinearity, whilst this probability decreases with t. On the other hand, very low t values lead to the exclusion of most of the variables from R_t and thus to less accurate models. To find a good trade-off, the entire process described hitherto has been repeated changing the value of the threshold t from 1 downwards, until the signs of the regressors of the models were all consistent with their value of correlation with tie strength, that indicates that multicollinearity does not exist between the variables in R_t. The corresponding R_t contains the largest possible set of variables that do not present multicollinearity. The described procedure converged at t equal to $.4$. Then, for each of the 500 subsampled training sets, a regression model has been built using only the variables in R_t. The statistics of the regressors of the predictive model obtained averaging the 500 models are reported in Table 2.3 (this model is referred to as "model without pairwise products" in the table), using the same enumeration of Table 2.2. For each regressor, the estimate of its weight, the standard error and its p-value are reported.

In addition to the model using this set of regressors, other benchmark models have been considered. The first one is a very simple model used as baseline to assess the validity of the other models. It is a constant model which returns the average score of the evaluations used during the training phase for each possible input. The second model uses the set of uncorrelated regressors identified using the procedure described above and, in addition, it includes all the pairwise products between the regressors. Using pairwise products is a standard technique in regression analysis to improve the fitting introducing a set of simple non-linear terms. Moreover, a model using the *recency of communication* as the sole regressor is considered, as this is the variable that correlates most with tie strength (see Table 2.2). For completeness, the model with all the variables as regressors is reported. This model, although suffers from multicollinearity and could be heavily overfitted, represents a reference for the other models. The coefficient estimates of the models and the respective standard errors and p-values are reported in Table 2.3. The constant model has been omitted from the table since it does not have any coefficients. To not compromise readability the model with all the variables has been also omitted from the table. The regressors with a p-value greater than $.05$ have been excluded from the models since they were not statistically significant.

Table 2.4: Statistics of the regression models based on uncorrelated variables.

Statistics of the models			
Model	R^2	**stderr**	**rmse**
average value	0	$.211$	$.219$
one regressor	$.272$	$.180$	$.184$
uncorrelated w/o pairwise prod.	$.345$	$.171$	$.177$
uncorrelated with pairwise prod.	$.350$	$.170$	$.177$
all regressors	$.454$	$.156$	$.165$

The standard errors indicate that the coefficient estimates are sufficiently reliable and the small p-values indicate their statistical significance.

For each model, the standard indices R^2 and the estimated standard error are computed. Then, each model is tested on the test sets, computing the $rmse$. These indices indicate how well the models fit the data set and their prediction accuracy on the test set. Specifically, The R^2 index indicates how much variance of the tie strength in the training set the models are able to explain. The more this measure is close to 1, the more the model is able to correctly approximate all the different values of tie strength. On the contrary, a low value of R^2 indicates that the predictions made by the model could be centred on an average value and the model is not able to capture the entire variability of the tie strength. The estimated standard error is the average value of the error made by the model while fitting the training set. The $rmse$ measures the mean error made by the model during the prediction phase and is calculated comparing the output of the model and the reference values in the test set (i.e. the tie strength explicitly evaluated by the users). For a precise definition of these indices see [27].

The results of the models are reported in Table 2.4. The first model, by definition, has a R^2 equal to 0, since it is centred on the average value of the scores in the training set. The estimated standard error and the $rmse$ of the model are about 21% and should be considered as worst cases to test the effectiveness of the other models. Even though these values seem adequate for a model aimed at predicting tie strength between people, the model is not able to fit all the different values of tie strength (because of the null R^2) and fails to reproduce the typical tie strength distribution of social ego networks, depicted in Figure 2.4 and reported in [85]. The model with only one regressor is able to explain 27.2% of the variance of the tie strength in the data set, according to its R^2. This represents a rather good result, considering that only one variable is used to estimate the tie strength, that is influenced by many different sociological and psychological factors. Note

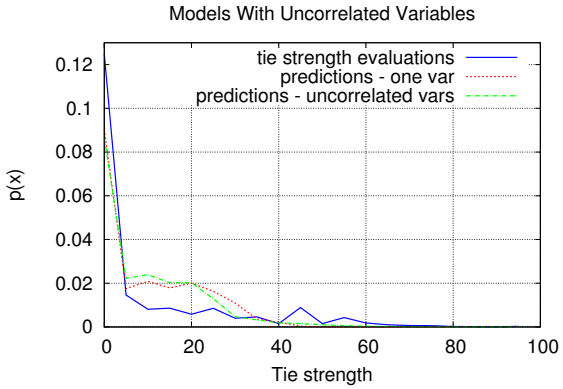


Figure 2.6: Distributions of the expected values of tie strength compared to the predictions made by the models built using uncorrelated variables.

that R^2 values around $.3$ are generally considered rather good results (e.g. [61]). Nevertheless, the remaining part of variance of the tie strength not explained by this first model is still large and this could limit the ability of the model to effectively predict all the different values of tie strength. The estimated standard error of the model is equal to 18%. This means that the model, on average, is able to fit the training set with a good accuracy. The $rmse$ of the model is really close to the estimated standard error. This is a good result, since indicates that the average error made on the test set has the same magnitude of the error made on the training set. Hence, the model seems not to be affected by overfitting and remains valid also when applied to data other than that used to train it. The model with the addition of the other uncorrelated variables to the *recency of communication* shows an improvement in terms of all the presented indices. Even if the improvements in terms of estimated standard error and $rmse$ are only $.9\%$ and $.7\%$ respectively, the R^2 is 7.3% higher than that of the model with one regressor. This improvement in terms of R^2 makes this model to be a better choice compared to the previous one, since it is more accurate in the fitting of all the different values of tie strength. The model with the introduction of the pairwise products of the variables does not bring a noticeable increment in terms of R^2 and $rmse$ to justify its higher complexity - represented by the higher number of regressors. Lastly, the model with all the variables as regressors shows the best performances, but, as stated before, it suffers from multicollinearity.

Figure 2.6 compares more in detail the predictions made by the models with respect to the explicit evaluations of tie strength in the training sets. Specifically, each curve in the plot shows the probability with which a given value of tie strength is predicted by the corresponding model, or appears in the explicit evaluations of tie strength. In other words, the curves show the empirical distributions of tie strength in the data set, and produced by the models. This provides a more detailed comparison with respect to the $rmse$ index, which is essentially an average accuracy index.

A graphical representation of the results of the model with the addition of pairwise products of the variables is not given in the figure, since the curve is really close to that related to the model without pairwise interactions. The constant model is also omitted from the figure, since its density function is a Dirac delta function centred in the average of the evaluations (i.e. 13.216). From the graphical representation in Figure 2.6 the reader can notice that the predictions made by the models using Facebook variables as predictors have similar distributions. Both models tend to overestimate tie strength when it is close to zero, since the density of the predictions is lower than the reference at zero and higher between 5 and 30. The data set is noisy in this particular region and this is likely to be the main reason for this inaccuracy. On the other hand, the models tend to overestimate tie strength for high values of tie strength. This prediction inaccuracy is likely to be due to the presence of few samples with a high value of tie strength. The both types of inaccuracy are likely to be mitigated using a larger data set. Nevertheless, Figure 2.6 qualitatively confirms that the estimations made by the models are in line with the explicit values.

The results described so far indicate that the models effectively predict tie strength using only a small set of Facebook variables (i.e. 4 in the model with all the selected uncorrelated variables). The first model, with only the *number of days since last communication* as regressor, already provides good prediction accuracy, confirming that this variable is a good predictor of tie strength. Nevertheless, using additional variables (i.e. the other regressors in the second model) provides even more accurate predictions.

Model With PCA Factors

A second approach has been used to build a model to predict tie strength, avoiding multicollinearity at the same time. A new model has been constructed by applying PCA on the 100 sub-sampled data sets, obtaining a set of orthogonal variables that has been used as regressors of the model. PCA is a standard technique that transforms a set of possibly correlated variables into a set of uncorrelated factors,

obtained as linear combinations of the original variables. The results of PCA are presented here in terms of factor loadings, that is to say the weights to be given to each original variable to obtain the factors themselves. Further details on PCA technique and a detailed description of the meaning of factor loadings and of other properties of the factors are given in [27]. The obtained factors have been used to build a regression model for each of the 500 training sets (100 sub-samples for each of the 5 combinations of training sets). Hence, the 500 models have been averaged obtaining a unique average model. Afterwards, the predictive power of this average model has been tested on the different test sets.

The results obtained using PCA, expressed in terms of factor loadings, are reported in Table 2.5. The numbering of the variables presented in the table is the same used in Table 2.2. Only the first 5 factors are considered, since they are the only ones with eigenvalue greater than 1, as suggested in [66]. In essence, all the factors with eigenvalue lower than one are dropped since they extract less than the equivalent of one original variable. Nevertheless, before putting aside the less important factors (in terms of explained variance) the correlation between all the factors extracted and the tie strength has been studied in detail. The first five factors, in addition to be the factors that explain the largest portion of variance of the data set (individually), are also the most correlated with tie strength. All the other factors show low values of correlation (i.e. below $.1$) or their p-value stays above $.05$. This means that they have a meaningless relation with tie strength.

Before continuing with the analysis and with the creation of a predictive model using the obtained factors, a characterisation of the physical meaning of each factor is given, based on the variables that determine it and their factor loadings (see Table 2.5). This gives a first broad idea on the nature of the principal dimensions contained in the data set and a preliminary comparison between the differences of these dimensions and those hypothesised by Granovetter in [57] can be performed.

The first two factors contain all the variables related to the communication between people, like the *frequency of contact*, the *time since last/first communication*, the *number of likes/posts/msg etc.* sent or received by egos. These factors are related to the time dedicated by two individuals to the social relationship that ties them together. Moreover, the factors embody the intensity of the communication related to the relationship. These factors are called “communication factors”. The first factor embodies the incoming communication and the overlap between the incoming and the outgoing communication (i.e. the incoming communication reciprocated by ego). The second factor contains the portion of outgoing communication not already contained in the first factor, that is to say the outgoing commu-

Table 2.5: PCA Factor Loadings.

Var	Factor				
	I	II	III	IV	V
1	$-.\mathbf{77}$	$-.29$	$-.07$	$-.17$	$-.12$
2	$.\mathbf{88}$	$.28$	$.09$	$.17$	$.11$
3	$.\mathbf{82}$	$.18$	$.10$	$.08$	$.03$
4	$.\mathbf{90}$	$.08$	$.09$	$.16$	$.14$
5	$.\mathbf{44}$	$.28$	$.22$	$.29$	$.30$
6	$.\mathbf{32}$	$.\mathbf{70}$	$.07$	$.22$	$.13$
7	$.21$	$.\mathbf{61}$	$.12$	$.\mathbf{31}$	$.28$
8	$.\mathbf{78}$	$.02$	$.18$	$.08$	$.07$
9	$.\mathbf{54}$	$.10$	$-.14$	$.24$	$.26$
10	$.\mathbf{35}$	$.\mathbf{51}$	$.10$	$.12$	$.06$
11	$.04$	$.\mathbf{56}$	$.14$	$.29$	$.24$
12	$.22$	$.23$	$.24$	$.29$	$.30$
13	$.16$	$.05$	$.22$	$.30$	$.\mathbf{34}$
14	$.06$	$.26$	$.15$	$.\mathbf{35}$	$.\mathbf{49}$
15	$.10$	$.05$	$.\mathbf{77}$	$.07$	$.05$
16	$.10$	$.09$	$.\mathbf{79}$	$.05$	$.03$
17	$.08$	$.19$	$-.03$	$.29$	$.\mathbf{45}$
18	$.06$	$.05$	$.\mathbf{46}$	$.13$	$.12$
19	$.01$	$.\mathbf{37}$	$-.09$	$.20$	$.21$

nication not reciprocated by alters. Although this requires better investigation, the fact that outgoing communication is split between the first two factors is likely to be the reason why they are uncorrelated. In general, incoming and outgoing communications are correlated, although less than expected. The correlation between the two types of communications in our data set is $.33$ ($p. < .01$). This fact is induced by the nature of Facebook, that allows people to consume information received by other users, but does not require that these people directly communicate with those specific users, as previously described by the authors of [16].

The third factor is a combination of the *number of groups* and *events* in common and the *number of tags on the same objects*. This factor represents how similar two Facebook profiles are and it can be called “social similarity factor”. The last two factors share broadly the same variables and they could be related to the intimacy and the emotional intensity of a relationship, since they also contain the number of pictures in which two users appear together, which are hypothesised as indicators of the emotional affinity of individuals.

Table 2.6: Coefficients of the regression models with PCA factors.

Regressor	estim.	std err.	p value
Model without pairwise products			
Intercept	13.342	.361	$< .01$
Factor I	10.565	.374	$< .01$
Model without pairwise products			
Intercept	13.143	.341	$< .01$
Factor I	9.224	.340	$< .01$
Factor II	5.444	.338	$< .01$
Factor III	4.418	.338	$< .01$
Factor IV	4.669	.339	$< .05$
Factor V	4.080	.339	$< .05$
Model with pairwise products			
Intercept	13.143	.336	$< .01$
Factor I	9.028	.367	$< .01$
Factor II	6.086	.453	$< .01$
Factor III	4.201	.355	$< .01$
Factor IV	5.167	.516	$< .05$
Factor V	4.718	.492	$< .05$
I*III	.913	.335	$< .05$
II*III	.530	.254	$< .05$
II*IV	$-.372$	.231	$< .01$
II*V	$-.471$	.204	$< .05$
IV*V	$-.242$	.208	$< .05$

Although it is necessary to extend this analysis to a larger number of users, these results suggest that the data set contains broadly the same dimensions hypothesised by Granovetter, even though the presence of variables indicating the intimacy and the reciprocal services in our data still remain as an hypothesis.

Using the factor scores obtained from PCA, three different regression models have been created. The first model uses only the first PCA factor as regressor (the factor with the strongest correlation with tie strength). A second model uses all the five PCA factors and a third contains all the factors and their pairwise products. Table 2.6 reports the coefficient estimates of the models along with their respective standard error and their p-values. The regressors with a p-value higher than $.05$ have been excluded from the models. These statistics indicate that all the regressors reported in the table are significant and their estimates are sufficiently accurate.

The R^2, the estimated standard error and the $rmse$ of the models are reported in Table 2.7. The first model has a noticeably lower value of R^2 compared to the other models and the error it makes during prediction is higher (almost 20%). The second model, instead, shows a good R^2, with a sensible improvement compared to the previous one. Also the $rmse$ and the average standard error indicate better performances, not far from the reference model built using all the possible regressors reported in Table 2.4. The third model introduces an additional improvement in terms of R^2, but its augmented complexity is not supported by a noticeable increment in terms of prediction accuracy. In fact, the $rmse$ is equal to that of the model without pairwise interactions and the presence of additional regressors could introduce overfitting. Hence, the second model turns out to be the best one, since it is simpler than the third one - maintaining a similar R^2 at the same time - and has a far better R^2 compared to the first model.

Figure 2.7 shows a graphical comparison between the distributions of the tie strength explicit evaluations in the test sets and the distributions of the tie strength predicted by the first and the second models, built using the PCA factors as regressors. The predictions made by the first model - with only the first factor as regressor - are not enough accurate, especially between 0 and 30. The model with the five PCA factors shows a good accuracy in the prediction, since the distribution of its output indicatively follows the distribution of the tie strength in the test set, even if the major part of the error is still concentrated between 0 and 25. The same hypothesis on the nature of the error made by the models, already highlighted for the models with uncorrelated variables in Section 2.2, holds also for Figure 2.7.

Comparison Between the Different Models

The models described so far have approximately the same predictive power in terms of $rmse$ ($M = .180$, $SD = .010$). This represents a good result, since all the models are able to predict tie strength with an accuracy greater than 80%. The little difference in terms of $rmse$ between the different models seems to indicate the model with only the *recency of contact* as regressor as the best choice, since it is

Table 2.7: Statistics of the regression models with PCA factors.

Model	R^2	**residual std err**	**rmse**
First PCA factor	.193	.189	.198
PCA factors I-V	.404	.163	.171
PCA factors I-V + pairwise prod.	.423	.161	.171

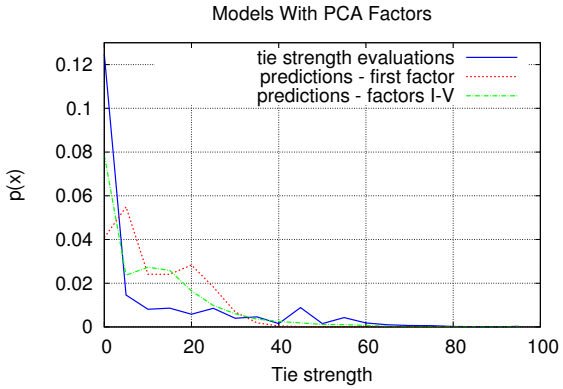


Figure 2.7: Distributions of the expected values of tie strength compared to the predictions made by the models built using PCA factors.

the simplest one. Nevertheless, the noticeable difference in terms of R^2 indicates that the models built using all the five PCA factors could be a better choice. In fact, the higher value of R^2 (not far from the R^2 of the reference model built with all the variables) makes this model able to better approximate all the possible values of tie strength and assure that the model does not produce always the same average score, but it effectively follows real tie strength distribution. For this reason the model with the five PCA factors is also the most general, and its validity is not limited to our particular data set. A drawback of this model is that it needs all relational variables. In cases where this is not feasible, the model using the four uncorrelated variables is a very good trade off. It is not much more complex than the one using only one regressor (*recency of contact*), and is able to provide higher R^2 and lower $rmse$, although it does not reach the performance of the model with all PCA factors.

It is noteworthy that the most important variable for tie strength prediction remains the *time since last contact* in all the models. The results also confirm that this variable is a good estimator of the frequency of contact, since it has a very high correlation with the *bidirectional frequency of contact* ($r = -.86$, $p < .01$). Moreover, it also represents a large portion of the first PCA factor. The *time since last contact* is also really simple to be obtained from Facebook and the model that uses only this variable as predictor requires only a small amount of information. In fact, it is sufficient to download only the last communication record and not the

whole history of interactions between the users to obtain the time at which the last contact between two online users occurred.

Discussions

In this section a detailed analysis of Facebook ego networks has been presented. This analysis has a double aim. On the one hand it provides a fine-grained characterisation of OSN ego networks properties and it studies the relation between these properties and the tie strength. On the other hand, the analysis is aimed at building a set of models able to predict tie strength from OSN data. To perform this analysis, a set of observations of Facebook variables concerning social relationships has been downloaded through a data acquisition campaign and the use of a dedicated Facebook application called FBA. The application also obtained tie strength estimation related to the involved users, asking the participants to explicitly evaluate their Facebook friendships through an electronic survey. From this data, it has been possible to find that the properties of Facebook ego networks are compatible with the findings regarding offline ego networks. In particular, the number of active relationships an individual can maintain - found in offline social networks (the well-known “Dunbar number”) is compatible the results found in Facebook. Facebook users in the sample have, on average, a maximum number of active friends equal to 105.14. This value falls inside the boundaries hypothesised for offline ego networks [49] and is similar to other active network mean sizes found in human social networks [105, 61]. Moreover, the distribution of active network size of the data set is similar to those found in offline ego networks [83, 61].

To study the composition of tie strength and to predict it using a set of Facebook variables, a series of regression models has been created, dividing the analysis in a first phase dedicated to the training of these models and a second phase in which they have been tested on another portion of the data set, different from that used for training. The predictive models are built following two different approaches. On the one hand, a group of variables not correlated with each other - discarding variables that can lead to multicollinearity - but having a sufficiently high correlation with tie strength has been taken, and a first regression model is created with these variables. On the other hand, PCA has been used to extract the principal factors of the data set - that are orthogonal and thus uncorrelated by definition - and the latter are used to create a second type of predictive model. Using PCA, it has been possible to compare the dimensions of tie strength hypothesised in the seminal work by Granovetter [57] and in [45] with the factors derived from the data set. The results of this analysis suggest that the collected variables represent all the dimensions of tie strength as hypothesised in [57].

The regression models perform quite well. They show R^2 indices comparable to other models in literature, namely to that presented in [61], regarding "offline" social networks, and that in [54], as far as online environments. Moreover, the validity of the models has also been also tested on a test set containing data different from that used to train them. They show good results in terms of prediction accuracy. On average, they achieve accuracy greater than 80%. The best one amongst them is the one using all the PCA factors as regressors. The main drawback of this model is that it needs to collect all the variables present in our data set, which may pose concerns in terms of practical applicability.

It is noteworthy that the most important regressors of the model, in terms of prediction power, are those implying the recency of communication and the frequency of contact between people involved in a social relationship, already used as tie strength predictors in other studies regarding "offline" social networks analysis [61]. Models using only this variable as predictor (which is similar to what has been typically done in the anthropology literature [61]) perform quite well, although do not match models based on the PCA factors. They are appealing, as they can be implemented by monitoring only one variable, which is a very low cost. An interesting trade off between one-regressor models and full-PCA models is achieved by a model that uses only four uncorrelated variables, and provides better fitting and prediction performance with respect to the model with one regressor. The 4-variable model keeps the complexity at a reasonable level, providing good (although sub-optimal) performance in terms of fitting and prediction accuracy.

In conclusion, the findings indicate that the characteristics of OSN ego networks are not so different from those found in offline ego networks, both in terms of their structure and tie strength composition. This means that, even if OSN like Facebook and Twitter give us many new and different ways to communicate, human social behaviour and the capacity to maintain social relationships with others seem to remain unaltered. These results clearly need further investigation on a much larger data set, more representative of the entire Facebook population, but they still represent a first interesting indication of the similarity existing between offline and online social networks. The tie strength models presented in this section clearly indicate that a lot of work still need to be done in OSN analysis to fully understand the global "social" properties of the networks. The obtained models are still preliminary and their performance must be improved, especially in terms of predictive power. Although this, this work demonstrates the feasibility of the creation of a general model for tie strength prediction that could represent the basis for more advanced studies in OSN analysis. Moreover, using the frequency of contact between users in OSNs provides a good estimate of their tie strength.

This is essential for the rest of the analyses presented in this thesis, which use the frequency of contact to study ego network structural properties on large-scale communication data sets.

2.3 Ego Networks in Twitter: an Experimental Analysis

This section contributes to give a detailed characterisation of the similarities between the structure of online and offline ego networks by analysing a large data set of Twitter communications. The data set, even though less detailed than the one described in Section 2.2, is very large and is a significant sample of the entire Twitter social network. The data have been filtered to obtain the frequency of contact of the relationships between pairs of users. For each user, her ego network has been built using the frequency of contact between her and her social contacts. Then, using clustering techniques, the presence of structures similar to those found in offline social networks has been assessed. The results show a striking similarity between the social structures in offline and online social networks. In particular, social relationships in Twitter share three of the most important features highlighted in offline ego networks: (i) they appear to be organised in four hierarchical layers; (ii) the sizes of the layers follow a scaling factor close to three; and (iii) the number of active social relationships is close to the *Dunbar's number*. These results, completely described in [26], further confirms what has been found in Section 2.2 and strongly suggest that the structural properties of offline and online social networks are similar, due to their direct relation with the properties of human brain.

To validate these results, the same technique has been applied on a different large-scale data set, collected from Facebook [24]. The results indicate that a similar structure can also be found in Facebook ego networks, suggesting that the similarities between offline and online ego networks are valid amongst different media and are thus general and not specific for a particular OSN.

2.3.1 Twitter

Twitter is an online social networking and microblogging service founded in 2006, with more than 500 million registered users as of 2012[2]. In Twitter, users can post short public messages (with at most 140 characters) called *tweets*. All the users' tweets are accessible by other users, unless the users' profiles are private or the

[2] According to Twitter CEO Dick Costolo in October 2012.

access is restricted by other specific settings. Users can also automatically receive notifications of new tweets created by other users by "following" them (i.e. creating a subscription to their notifications). People following a specific user are called her *followers*, whilst the set of people followed by the user are her *friends*.

Tweets can be enriched with multimedia content (i.e. URLs, videos, pictures) and by using special text characters to insert additional information. Specifically, a tweet can reference one or more users with a special mark called *mention*. Users mentioned in a tweet automatically receive a notification, even though they are not followers of the tweet's author. Users can also *reply* to tweets. In this case, a tweet is generated with an implicit mention to the author of the replied tweet. This implies that replies represent directional communications. Replies often require additional effort in terms of cognitive resources compared to other tweets since they presuppose that the user creating the reply has read the tweet she is replying. Twitter has also a private messaging system, however, since private messages are not publicly accessible, we did not collected them in our data set.

In addition to mentions and replies, Twitter provides a series of mechanisms for broadcast communication that represent the most popular features of the platform. Firstly, all the tweets are automatically sent towards all the followers of their authors. Moreover, tweets can also be *retweeted*. A user can make a retweet to forward a tweet it to all her followers. Each tweet can be assigned to a topic through the use of a special character called hashtag (i.e. "#") placed before the text indicating the topic. Hashtags are used by Twitter to classify the tweets and to obtain *trending topics*.

2.3.2 Data Set Description

A crawling agent has been created to download user profiles and their communication data from Twitter. The agent visits the Twitter graph considering the users as nodes and their contacts as links. In particular, it has been assumed that a link between two nodes exists if at least one of the users follows the other or an interaction between them has occurred. The presence of *mentions* in a tweet (i.e. the fact that a user explicitly mentions the other in a tweet) and *replies* (responses to tweets) have been used as indicators of the presence of social relationships between users. Replies represent directional communications between the user who is replying and the user who generated the original tweet. Replies require additional effort in terms of cognitive resources compared to other tweets since they presuppose that the user creating the reply has read the tweet she is replying. For this reason, the number of replies sent by one user to another has been used as an indication of tie strength.

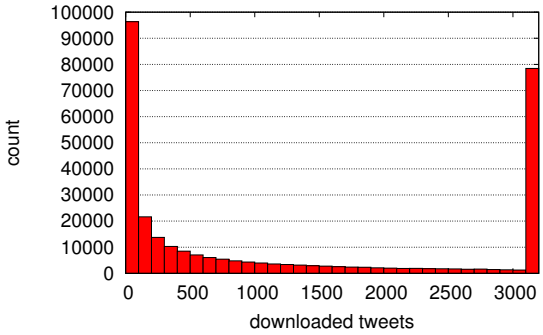


Figure 2.8: Downloaded tweets per user distribution.

The crawling agent started from a given user profile (seed) and visited the Twitter graph following the links. For each visited node, the Twitter REST API have been used to extract the user *timeline* (i.e. the list of tweets she posted including mentions and replies), her *friends* (i.e. people she follows) and her *followers* (i.e. people who follow her). Twitter REST API limits the amount of tweets that can be downloaded per user to $3,200$. This does not represent a constraint to our analysis since, as explained in the following, it was sufficient for the purposes of this study.

The crawling agent used 250 threads that concurrently accessed a single queue containing the ids of the user profiles to download. Each thread extracted a certain number of user ids from the queue, then it got the related profiles and communication data from Twitter using the REST API. Finally, after extracting new user ids from the communication data and from the friend/follower lists, the threads added them to the queue. The use of multiple threads both speeded-up the data collection process and avoided the crawler to remain trapped in visiting the neighbourhood of a node with a large number of links. The seed used to start the data collection is the profile of a widely know user (user ID: 813286), so that her followers represent an almost random sample of the network.

The collected data set contains $303,902$ Twitter users, whose data was downloaded in November 2012. In the column “all users” of Table 2.8 some statistics of the data set are presented, while in Figure 2.8 the distribution of the number of tweets downloaded per user is shown. In the figure the presence of a peak in correspondence of the value $3,200$ - that is the maximum amount of tweets downloadable using the Twitter REST API - can be identified. Cases where the number of tweets is lower than $3,200$ correspond to users that have generated less than $3,200$ tweets since their account has been created. The number of users

that posted an amount of tweets above this threshold is indicated in the table by $N_{3,200}$. The table also reports the average number of tweets, friends and followers per user and the average ratio of replies and tweets containing mentions (over the total number of tweets). Each average value is reported with 95% c.i. between square brackets. Data reported in the table indicate that around 20% of the tweets downloaded by the crawler contain direct communication between people, important for this study. This percentage is sufficient to perform significant analysis on the data set.

2.3.3 Classification

Differently from other online social networking services, Twitter is designed to encompass heterogeneous types of users. In fact, in addition to accounts used by persons mainly to communicate and maintain their social relationships with others (hereinafter referred to as *socially relevant users*), there exist Twitter accounts representing companies, public figures, news broadcasters, bloggers and many others, including spammers and bots.

Since the analysis is focused on the characterisation of social aspects of human relationships in online environments, an automatic procedure to distinguish between socially relevant users from all the other accounts has been implemented. To this aim, a supervised classifier has been used to divide Twitter accounts in two classes labelled “soc. rel. users” and “others” respectively. A sample of 500 accounts, randomly drawn from the data set, has been manually classified and these classifications have been used to train a Support Vector Machine [43]. This SVM uses a set of 115 variables: 15 of them related to the user’s profile (e.g. number of tweets, number of friends and followers, account lifespan) and 100 obtained from her timeline (e.g. percentage of mentions, replies and retweets, average tweets length, number of tweets made using external applications).

To test the generality of the SVM (i.e. the ability to categorise correctly new examples that differ from those used for training) 10 random sub-samples of the training set have been taken, each of which contains 80% of the entries, keeping the remaining 20% for testing. Then, the same methodology used to create the SVM generated from the entire training set has been applied on the 10 sub-samples. Doing so, several SVMs have been obtained, trained using different sub-samples of the training set, and of which the accuracy can be assessed. The average accuracy of these SVMs can be seen as an estimate of the accuracy of the SVM derived from the complete training set. Specifically, the *accuracy* index, defined as the rate of correct classifications, and the *false positives rate*, where false positives

are accounts wrongly assigned to the "soc. rel. users" class are used to evaluate the goodness of the result. In the analysis only users falling in the "soc. rel. users" class have been considered, thus it is particularly important to minimise the false positive rate[3]. Minimising the false negative rate is also important but less critical, as false negatives result in a reduction of the number of users on which we base the analysis.

The average accuracy of the classification system is equal to 0.813 $[0.024]$ and the average false positives rate is 0.083 $[0.012]$ (values between brackets are 95% c.i.). These results indicate that socially relevant users in Twitter are identified with sufficient accuracy, even if users have different behaviours and characteristics (e.g. different culture, religion, age). Moreover, the false positive rate is quite low (below 10%). The results are of the same magnitude as those found in a similar classification performed in Twitter [39].

After applying the classifier to the whole data set, $205,108$ socially relevant users have been isolated. Some properties of the classes "soc. rel. users" and "others" are reported in Table 2.8. It is worth noting that users in the class "others", on average, have a much higher number of friends and followers compared to the users in the class "soc. rel. users". Similarly, there is a higher number of tweets from the user belonging to "others" than from the users in "soc. rel. users" class. In the table there is also a comparison of other important variables, extracted from the classes, which exhibit significant dissimilarities. While the users in "soc. rel. users" class have a higher use rate of replies, user in "others" show a higher usage of mentions. Even though the number of tweets downloaded for the two classes do not significantly differ, the number of accounts with more than $3,200$ tweets is much higher in the "others" class. These results are aligned with the intuition about the different use of Twitter by humans to maintain social relationships, with respect to other type of users, in particular commercial and political ones. Specifically, "soc. rel. users" tweet less than "others" and have (far) less friends and followers. It is also interesting to note the higher percentage of replies, which is an indication of a more marked attitude towards bidirectional interactions, which is also an intuitive difference between the two classes.

2.3.4 Analysis

The "soc. rel. users" data set has been used to analyse the structure of ego networks in Twitter. Each Twitter user in the data set has been considered as an ego,

[3] False negatives are "soc. rel. users" with behaviour similar to the subjects in the "others" class. For this reason they have been considered as outliers, since the analysis is focused on Twitter average users.

Table 2.8: Data Set (all users) and Classes Statistics.

	all users	**soc. rel. users**	**others**
N	$303,902$	$205,108$	$98,794$
$N_{3,200}$	$77,196$	$38,107$	$39,088$
$(\%\ N_{3,200})$	(25.4%)	(18.6%)	(39.6%)
$\#\ tweets$	$1,234$ [5]	979 [5]	$1,764$ [8]
$\#\ friends$	$1,905$ [33]	673 [8]	$4,462$ [98]
$\#\ followers$	$11,335$ [529]	777 [107]	$33,254$ [$1,602$]
$\%\ tweets_{\mathrm{REPL}}$	17.4%	18.4%	15.4%
$\%\ tweets_{\mathrm{MENT}}$	22.7%	21.6%	24.7%

and *alters* are users in Twitter to whom the ego has sent at least one reply. A reply implies bi-directional communication and indicates that both the user and her friend has spent a certain amount of their cognitive and time resources to interact.

Users' Interaction as a Function of Ego Network Size

The first analysis performed is a study of the average number of replies sent by the users to their friends. Specifically, in [55], this was the main index used to conclude that a concept similar to that of the Dunbar's number (the maximum number of active social relationships a human can maintain) holds also in Twitter ego networks. By analysing this index it has been possible to understand whether the present data set is aligned with the one used in [55] as far as this index is concerned.

Figure 2.9 depicts the trend of the average number of replies per friend as a function of the number of friends of the user. Differently from [55], the analysis is divided for the two classes identified in the previous Section: "soc. rel. users" and "others". The results, supported by the figure, highlight a clear distinction between the properties of the two classes.

The class "soc. rel. users" shows a higher mean value of replies per friend and a maximum around 80 friends. This is an indication of the effect of the cognitive limits of human brain on the ability to maintain social relationships in online social networks. The peak of the curve identifies the threshold beyond which the effort dedicated to each social relationship decreases. This is due to the exhaustion of the available cognitive/time resources that, therefore, have to be split over an increasing number of friends. As discussed in [55], this can be seen as an evidence of the presence of the so called Dunbar's number in Twitter.

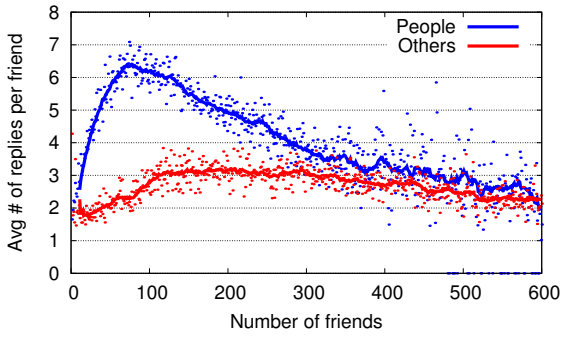


Figure 2.9: Average number of replies as a function of the number of friends; thick lines are running averages.

The class “others” shows a more random pattern, with lower average value of replies per friend without any significant discontinuities. This indicates that the accounts belonging to the class “others” are not influenced by cognitive capabilities. In fact they are often managed by more than one person or by non-human agents.

Structure of Ego Networks: Methodology

A refined selection of the ego networks in the data set permitted the identification of the most relevant set of accounts for the study. Specifically, a methodology similar to the one used in [24] to analyse ego networks in Facebook has been used to analyse ego networks in Twitter. Too recent accounts (i.e. with duration shorter than 6 months) have been removed, since they are not long enough to allow users to create a meaningful ego network, i.e. to select friends in the ego network and communicate with them long enough to well reflect the level of intimacy of the relationship. For the same reason, friendships with duration shorter than one month have not been considered in the analysis.

After the selection of the ego networks relevant for the analysis, their structure have been studied through the use of standard clustering techniques to find out if social relationships of the ego networks could be grouped according to their frequency of contact. The frequency of contact between users has been obtained by dividing the number of replies sent by egos to the considered friend by the duration of the friendship (i.e. the time since the first mention or reply sent to the friend). Hence, a quantitative analysis of the properties of the groups of relationships has been performed to highlight analogies and differences with offline ego networks structure and the results found in Facebook. To this aim the $k-$means

algorithm has been used. With the $k-$means algorithm, the frequencies of contact of each ego network are partitioned into a fixed number (k) of different clusters, according to their Euclidean distance. To find the number of clusters in each ego network, $k-$means has been repeatedly applied with increasing values of k. For each value of k, the standard k-means technique provides an index between 0 and 1 that measures the quality of the obtained clustering. This index monotonically increases with k. It is a standard technique to assume as optimal k the one beyond which increasing k yields an increase of the index below a given threshold. This threshold has been set to 0.1, as done in [24], to be able to obtain comparable results.

Structure of Ego Networks: Analysis

The distribution of the characteristic number of ego network circles in the data set, depicted in Figure 2.10, shows that most of the ego networks have 4 circles. Specifically, the average number of circles is equal to 3.14 and its median is 4. Moreover, Table 2.9 reports some statistics (with 95% c.i. between square brackets) about ego networks aggregated for different number of circles. It is worth noting that, as the number of circles increases, the average network size and the average Twitter use rate (defined as the average frequency of contact multiplied by the number of friends) also increases. The Twitter use rate is a proxy for the amount of time a user spends in Twitter, that is to say the budget of time the user allocates for socialising in Twitter. Another interesting finding is that the ego networks with 4 circles are those with the highest average number of replies sent per friend. According to the methodology used in [55], this marks the point where the cognitive "capacity" allocated to social relationship is saturated.

According to the results, four clusters, similar to the number of circles found in offline ego networks, have been found also in Twitter. Nevertheless, it is noteworthy that there is a non negligible amount of users in Twitter with only three clusters. This is a strong indication of the presence of two different kinds of users in online social networks: (i) *occasional users*, with a small three-clustered ego network

Table 2.9: Properties of Ego Networks with Different Number of Circles.

Circles	# ego nets	Avg net size	Avg use rate	Avg # replies
2	$3,819$	3.04 $[0.01]$	2.86 $[0.29]$	2.71
3	$27,788$	38.05 $[0.83]$	62.63 $[1.48]$	5.19
4	$53,982$	80.31 $[0.86]$	113.28 $[1.33]$	5.35
5	$1,073$	190.03 $[14.31]$	167.06 $[12.15]$	3.81

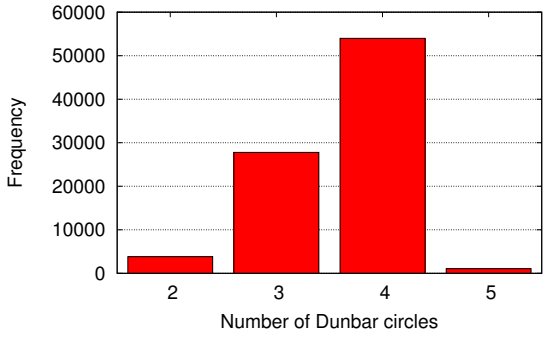


Figure 2.10: Distribution of the number of ego network circles.

and a low Twitter use rate, and (ii) *active users*, with an higher use rate and a number of clusters similar to that found in offline environments. This distinction is similar to the difference between more- and less-sociable individuals found in offline environment [51].

The properties of the identified ego network circles have been further analysed in Table 2.10, where k-means is applied, with $k = 4$, to all the ego networks in the data set. For each ego network, 4 clusters have been obtained, that are defined as S_1, S_2, S_3 and S_4, sorted by decreasing value of the centroid (i.e. the average frequency of contact of the cluster) so that S_1 represents the cluster of the social links with the highest frequency of contact.

The clusters obtained by using the k-means are not directly comparable with the circles of offline ego networks discussed in Section 2.1. In fact, whilst clusters are disjoint groups, social circles, as depicted in Figure 2.1, are hierarchically inclusive (i.e. the *support clique* is included in the *sympathy group* which is included in the *affinity group* which is included in the *active network*). For this reason, in order to compare social structures in online and offline ego networks, the clusters have been aggregated to form hierarchically inclusive circles. Specifically, the circles C_1, C_2, C_3 and C_4 have been defined as $C_k = \bigcup_{i=1}^{k} S_i$ so that $C_1 \subseteq C_2 \subseteq C_3 \subseteq C_4$.

The frequency of contact is measured in number of replies per month. The typical frequencies of contact of the circles (i.e. the minimum frequency needed for a relationship to be part of the circles) are $\sim$ once every two days for C_1, $\sim$ *weekly* for C_2, $\sim$ *monthly* for C_3 and $\sim$ *twice a year* for C_4. It is remarkable that some of the circles found in Twitter show properties similar to those found in offline social networks. In particular, C_2 and C_3 respectively resemble the support clique and the sympathy group in terms of size and frequency of contact. C_4, according

Table 2.10: Properties of ego network circles in Twitter.

	$\mathbf{C_1}$	$\mathbf{C_2}$	$\mathbf{C_3}$	$\mathbf{C_4}$
Size	1.74 [.03]	5.75 [.07]	17.56 [.21]	70.04 [.69]
Scaling factor	3.31		3.06	3.99
Contact freq.	17.28	6.00	1.77	0.20

to its properties, can be placed between the affinity group and the active network. These results are compatible with the ego network model also as far as the scaling factor between the circles is concerned, that is approximately equal to 3.

From the analysis performed, an additional circle (C_1) emerges. It is typically formed of one or two people strongly connected to the ego. This circle could be seen as a *super support clique*, and the alters contained in it are the most important relationships for the ego, perhaps a partner and/or a best friend. Scientists have long predicted the existence of this circle but they have never been able to prove it[4], due to the limitation of the methods used in offline analysis. The existence of such an additional circle, although needs to be supported by more detailed analyses, provides a very interesting result from the standpoint of the study of human social networks, and show a concrete example of the potential of characterising them through data collected on social networking sites.

The results also indicate that the size of the active network in Twitter is smaller than the reference value found in offline environments. Looking carefully at it, its frequency of contact appears to be lower than the affinity group ($\sim$ eight times a year as found in [24]), but higher than the active network described in the ego network model (once a year). The small size of this circle could be conditioned by the use of Twitter replies to weight social relationships. This index may not be the best choice to measure weak relationships, as it emphasises a lot the intentionality of interaction, which may be less present in weak relationships (than in strong ones). Nevertheless, the size of this circle is compatible with other results in literature about offline networks [83].

2.3.5 Validation on Facebook Data Set

The same analysis has been conducted on a different data set extracted from Facebook (completely described in [24]), to verify the results on a different medium. The same methodology applied on Twitter has been used to extract the ego networks from the Facebook data set and to select the set of ego networks relevant

[4] As stated by R.I.M. Dunbar in a private communication on June 19, 2012.

Table 2.11: Properties of ego network circles in Facebook.

	$\mathbf{C_1}$	$\mathbf{C_2}$	$\mathbf{C_3}$	$\mathbf{C_4}$
Size	(4.70)	(15.31)	(44.77)	(132.50)
Scaling factor.		3.26	2.93	2.96
Contact freq.	5.09	1.95	0.67	0.11

for the analysis, that are $91,347$. Hence, the k-means algorithm has been applied on these ego networks fixing $k = 4$. The results (reported in Table 2.11) are consistent, in terms of scaling factor and typical frequency of contact, with those found in Twitter. As explained in detail in [24], the Facebook ego networks extracted from the data set used for the analysis are not complete since they come from a random sample of the network. Despite this, since they represent a random sample of the respective ego networks, their size is proportional to the real size of the complete ego networks they represent. The size of all the layers in the ego networks have been scaled hypothesising that the size of the active network was equal to the size of the offline active network. The results are presented in Table 2.11, where the estimated size of each circle is reported in round brackets.

The results clearly highlight a similarity between Facebook ego network circles, the results found in Twitter, and the properties of offline ego networks. Specifically, the scaling factor found in Facebook is, on average, very close to 3. The typical frequency of contact of the circles are $\sim$ *weekly* for C_1, $\sim$ *twice a month* for C_2, $\sim$ *eight times a year* for C_3, and $\sim$ *yearly* for C_4. In this case there seems to be a match between all the four circles in Facebook ego networks and the circles found offline. Namely, C_1 is compatible with the support clique, C_2 with the sympathy group, C_3 with the affinity group, and C_4 with the active network. These results further confirm that ego networks in OSNs have the same structural properties of offline ego networks.

2.3.6 Discussions

This section presented an analysis of a real Twitter data set containing a high number of communication records to investigate the structure of ego networks in Twitter. The users have been divided in two categories, namely “soc. rel. users” and “others” to effectively study the properties of social relationships in Twitter. A standard clustering technique has been applied to the data set to characterise the structural properties of Twitter ego networks. The results indicate that Twitter presents a social structure qualitatively similar to that found by in offline ego networks. This suggests that the structure of ego networks in OSNs is consistent

among different social media. Moreover, Twitter ego networks show an additional small circle, not present in the taxonomy in the literature, formed of, on average, one or two people with extremely strong social relationships with the ego. In addition, the active network size in Twitter appears to be smaller than that found offline. These results indicate, on the one hand, a strong similarity between online and offline social networks and, on the other hand, the presence of additional properties not visible in offline networks. Even though these new properties have not yet been investigated in sociology, they have an intuitive meaning in humans and they should be further investigated to understand their role in social networks. The same analysis has been performed on a data set collected from Facebook, finding results qualitatively similar to those found in Twitter. This confirms that the results are consistend amongst different medium and are not specific of a particular OSN.

2.4 Dynamics of Personal Social Relationships in Online Social Networks: a Study on Twitter

Although analyses on OSNs conducted hitherto revealed some important structural properties of online social ego networks, there is still a lack of understanding of the mechanisms underpinning these properties, their relation to human behaviour, and their dynamic evolution over time. These aspects are clearly important to understand and characterise OSNs and to identify the evolutionary strategy that favoured the diffusion of the use of online communications within the society.

In this section a data set of Twitter communication records is analysed to assess the dynamic processes that govern the maintenance of social relationships online. The results reveal that Twitter users have highly dynamic social networks, with a large percentage of weak ties and high turnover. This suggests that this behaviour can be the product of an evolutionary strategy aimed at coping with the extremely challenging conditions imposed by the post-modern society, where dynamism seems to be the key to success [23].

2.4.1 Data set description

The data used in this section is an extension of the data set described in Section 2.3. The present data set contains many more profiles, representing four additional months of download. For this reason it has been possible to carry out a detailed analysis of the dynamics of Twitter ego networks.

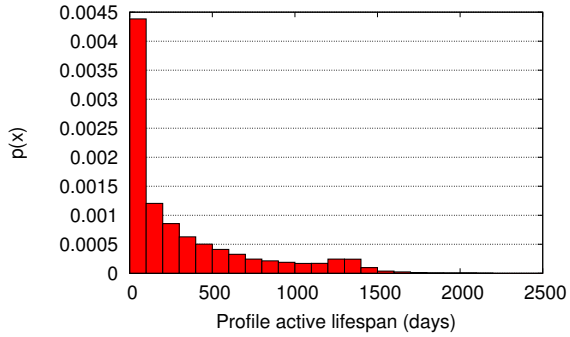


Figure 2.11: Distribution of active lifespan of Twitter ego networks.

The data set has been filtered with the SVM described in Section 2.3 to split the users in the two categories “soc. rel. users” and “others”. The data set contains $1,653,155$ “soc. rel. users”, that represents about 68% of the total number of downloaded users. The large number of users belonging to the “others” group gives a first interesting picture of Twitter. In fact, it indicates that Twitter is an online environment where different types of users coexist and interact. This feature makes OSNs like Twitter different from more traditional communication means, which often create a separation between different social environments. People using Twitter receive multiple social benefits at the same time, being able to manage more social domains in the same place.

The first column of Table 2.12 summarises the properties of the profiles in the data set, considering “soc. rel. users” only. The mean values of the indicated statistics, reported in the table, are averaged for all the users. Figure 2.11 depicts the active life span distribution of the “soc. rel. users” accounts (see Section 2.3 for the definition of active lifespan). The shape of the distribution in Figure 2.11 indicates

Table 2.12: Data Set Statistics.

variable	mean - all	mean - active
duration (days)	321.846 [0.628]	448.201 [0.762]
replies	208.923 [0.609]	290.885 [0.801]
mentions	103.882 [0.459]	144.634 [0.625]
retweets	151.492 [0.496]	210.924 [0.661]
plain text twts	280.037 [0.773]	389.810 [1.011]
twts w urls	4.813 [0.032]	6.698 [0.045]
twts w hashtags	56.411 [0.203]	78.529 [0.273]

that either most of the profiles have been created just before we downloaded the data or their activity on Twitter is very low. However, the long tail indicates that the obtained profiles have a tweet history of up to almost 7 years (i.e. the complete tweet history of some of the oldest profiles in Twitter), despite the limit of $3,200$ tweets imposed by the Twitter API. Indeed, only 0.02% of the "soc. rel. users" profiles in the data set exceed this limit. Nevertheless, the small peak in the distribution between $1,200$ and $1,400$ days could be ascribed to the presence of this limit, that prevented from obtaining the complete active lifespan of some of the downloaded profiles. Despite this, the number of profiles affected by this problem is very low and their last $3,200$ tweets are in any case a significant sample to describe their social behaviour. For this reason, the data set collected is well suited for the analysis.

In Table 2.12, it is worth noticing that the mean active lifespan of the "soc. rel. users" profiles in the data set (i.e. "duration" in the Table) is equal to 321.846 days. This indicates that, on average, almost one year of communications has been captured for each user and this is sufficient to conduct the analysis. Replies and mentions are about 39% of the total number of tweets made by "soc. rel. users" in Twitter. The exchange of these messages can be interpreted as a mechanism to actively maintain social relationships online and they should be strongly affected by the cognitive limits of human brain, since they require the users to spend cognitive resources to directly communicate with the involved people. Besides, non-direct messages take the largest part of the communication in Twitter. This kind of communication is controlled by a more *public* behaviour compared to direct messages and it should require less cognitive resources, since non-direct tweets are expected to contain a low value of emotional intensity.

The high number of replies could indicate a high number of communication threads between people. In fact, replies can also be used to reply to a previous mention and some communication threads are composed by an initial mention and a series of replies to that mention. The presence of communication threads is supported by the fact that the number of replies is, on average, broadly twice the number of mentions. This is another strong indication of the maintenance of social relationships online. Retweets are largely used by Twitter users and represent the willingness of people to spread messages they are interested in within the network. Seen from an evolutionary perspective, the diffuse usage of retweets could represent a strategy used by humans to receive a global benefit from having access to more information in the network, at the cost of being active in the diffusion process.

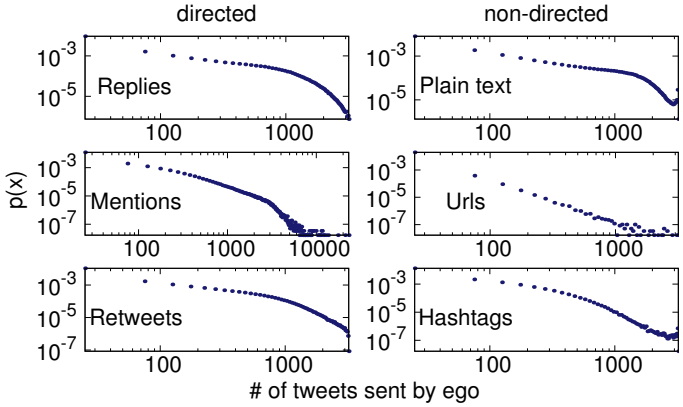


Figure 2.12: Distribution of the number of tweets divided by type.

Remarkably, non-direct tweets containing urls are less used than the other type of messages (only 4.813 tweets with urls sent on average by the users in our data set during their active lifespan). In addition, the small number of tweets with hashtags (i.e. 56.411 on average) could be ascribed to the fact that Twitter officially introduced hashtags only between 2009 and 2010.

After selecting "soc. rel. users" from the data set, all the profiles that have not sent any tweets (i.e. with null active lifespan) have been discarded, reducing the number of profiles to $1,187,105$. The statistics of these set of profiles are reported in the second column of Table 2.12. All the statistics of the profiles increase when users with null active lifespan are not considered, since the removed profiles do not contribute actively to the generation of content in the network.

Figure 2.12 depicts the distribution of the communication variables in the data set for the profiles with positive lifespan. Direct and non-direct communications have been separated, with the former identifying the explicit intention of the user to mention other users in the messages. In the figure retweets are labelled as direct communication, but their nature needs further investigation. In fact, retweets are more similar to non-direct tweets, with the exception that they contain the id of the user that initially generated the message and the ids of users that retweeted it.

Mentions show a very long tail (the scale of the x axis is different than for the other graphs), with some accounts generating up to $23,104$ mentions. This high number of mentions - apparently exceeding the limit of $3,200$ tweets - is due to the fact that a single tweet can contain more than one mention at the same time, or, in other words, many people can be mentioned in the same tweet.

Before continuing with the analysis the data set has been further refined, eliminating all the profiles created less than one year before the time of their download. This reduces possible artefacts due to including recently created accounts (with respect to the end of the download) as well as accounts that have been active only for a short amount of time. The data set, after this selection, contains $644,014$ accounts.

From tweets to ego networks

From the set of active soc. rel. users, a social ego network for each profile has been built. To do so, a measure of the strength of social links between people in Twitter has been defined. A social relationship exists between two users, A and B, if A sent at least a reply or a mention to B. This definition involves a cost in terms of cognitive effort spent for the maintenance of the relationship. As an estimate of the tie strength, the number of messages sent by A to B has been chosen. In this way, tie strength grows linearly with the number of messages exchanged between two users. Representing tie strength in this way is, at the moment, a good solution, since models to study the relation between tie strength and frequency of contact (described in Section 2.2) indicate that using linear approximations based on the frequency of contact leads to sufficiently accurate results.

Using the standard terms in ego network analysis, an *ego* has been defined as a user associated with a profile and her *alters* are all the people with whom ego has a social relationship. This definition gives a "static" view of the ego networks in the data set, aggregating all the communication of the egos, as already done in the previous sections. This definition gave a qualitative comparison between the ego network size in the data set and that found in the other sections. Hence, an analysis of the dynamic properties of social relationships has been performed. The total number of social relationships in the data set is $57,548,091$ with an average of 89.36 relationships per profile. This result is in accordance with the findings in Section 2.2, and 2.3.

To better understand how these ego networks evolve over time, the time series of the tweets sent by ego have been analysed, together with the composition of snapshots of the ego networks considering the communication occurred in time windows of one year each. This revealed important insights regarding human social behaviour in OSNs.

2.4.2 Methods

To perform the analysis, the time series of the direct tweets (replies and mentions) and of the non-direct tweets sent by each ego have been studied. For some performance indices (i.e. new users contacted per day and total number of new users contacted) the number of new alters contacted by ego each day until the network is active are counted. Instead, for analysing the dynamics of the ego network structure, the tweets time series have been sliced, taking snapshots of the duration of one year each, then assessing the size and the composition of ego networks in each snapshot. The one-year temporal window have been slid, taking steps of one day each, looking at how ego networks change over time.

By taking temporal windows of one year, all the active contacts maintained by each ego have been captured, according to the definition of active network introduced in Section 2.1 that identifies as "active" friends all the alters contacted by ego at least yearly. In this way, relationships that the users abandoned over time have been also identified. Note that abandoned relationships are not related to the notion of "unfollowing" (i.e. the explicit request of a user to remove a person from her friends), since unfollowing is an extreme action that does not capture the decline of a social link, but rather identifies sudden breach in the relationships, due to particularly negative and rare conditions.

The sympathy group in Twitter ego networks has been defined, according to the definition given in the literature, as the set of alters contacted at least once a month (i.e. contacted at least ~ 12.17 times in one year), and the support clique as the set of alters contacted at least once a week (i.e. contacted ~ 52.14 times in one year). By defining these circles, the changes in the different layers of the ego networks can be studied over time. Section 2.1 presents in detail the definitions of the different ego network circles.

To analyse the average behaviour of all the ego networks the first communication of each ego network (the time when ego started to actively communicate) has been shifted, so that the communication history of the ego networks start at the same point in time, specifically at the origin of the coordinate system of each graph reported in the following sections.

To deeply analyse the behaviour of different users in Twitter, the users have been classified in three categories on the basis of their active lifespan and their differences have been analysed in terms of social behaviour between these classes. To do so, the maximum lifespan in the data set has been divided it into three equal parts, obtaining three groups of 802 days of duration each. The choice to create three categories represents a good trade-off between the accuracy of the results

and their statistical significance. In fact, adding more categories would have decreased the number of users in each group, leading to low significance. The identified classes of users are the following: (i) occasional users (lifespan $<= 802d$) ; (ii) regular users ($802d <$ lifespan $<= 1604d$) and (iii) aficionados (lifespan $> 1604d$). These different categories of users should intuitively show different behaviours and different ego network properties. The data set is composed of 63.23% of occasional users, 35.22% of regular users and 1.55% of aficionados[5].

Note that in the graphs presented in the following, regarding the composition of ego networks in each one year snapshots (right-hand side plots in Figure 2.13, 2.14, 2.15, 2.18), the value of the x axis represents the starting point of each snapshot. Thus, the maximum value of the axis is equal to the maximum lifespan of the ego networks in the considered class, minus the duration of the snapshot (one year). The figures depict the average values as the curve in bold and the corresponding 95% c.i. as a lighter coloured area around the curve (barely visible, most of the time).

As another contribution of this study, the evolution of the recency of contact between users (i.e. time since last contact) has been analysed to understand how single social relationships evolve. To do so, the elapsed time between consecutive messages within each relationship has been analysed. The results have been averaged within the ego networks and then averaged for all the ego networks. While this clearly mixes the properties of different type of social relationships for a particular ego network, it provides a unique index to compare the ego networks of different classes of users, as explained in detail in the following.

After the analysis of the evolution of ego networks and personal social relationships over time, the stability of ego networks has been studied, assessing the proportion of alters that users maintain in their networks over time. This proportion has been estimated by comparing consecutive - but separated - one year snapshots and calculating their average Jaccard coefficient, then averaging the results for all the ego networks. The Jaccard coefficient is a measure of the percentage of overlap between sets defined as:

$$J(W_1, W_2) = \frac{|W_1 \cap W_2|}{|W_1 \cup W_2|} \tag{2.1}$$

where W_1 and W_2 are two sets, in the case of this study the one-year windows of the ego networks. The Jaccard coefficient can be a value between 0 and 1, with 0 indicating null overlap and 1 a complete overlap between the sets. The

[5] Note anyway that there are still about $10,000$ aficionados in the data set, which makes the analysis of also this class significant

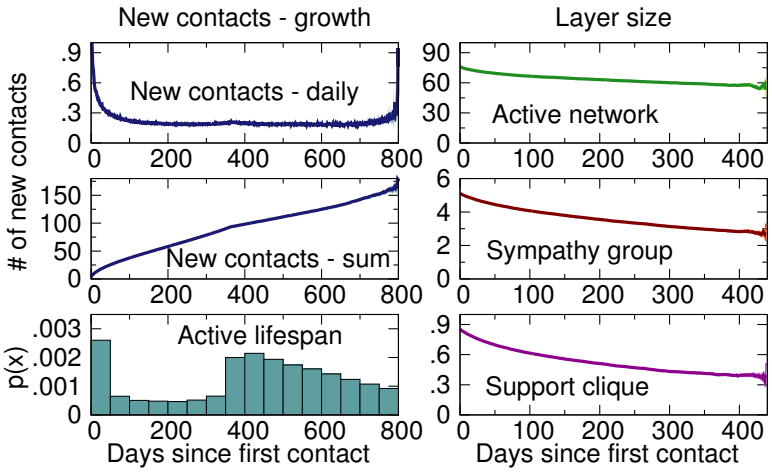


Figure 2.13: Ego networks properties for occasional users.

Jaccard coefficient has been calculated for the different layers in the ego networks to determine the "turnover" that takes place in the ego networks. This study is fundamental for understanding whether people maintain a stable network of contacts in Twitter or they prefer to vary their social relationships over time, and permitted a fine-grained characterisation of the users, dividing them into different classes: (i) users with structured ego networks, showing ego networks with composition and turnover similar to those found in other more traditional social networks and (ii) people without structured ego networks, showing higher turnover.

2.4.3 Results

This Section reports the results of the analysis and their interpretation from the point of view of human social behaviour. The main axes of the analysis, as previously identified, are the presence of different categories of users and, on the other hand, the presence/absence of a structured ego network.

Twitter abandonment

As a first contribution of the analysis, the behaviour of users that abandoned Twitter has been studied. A user abandoned Twitter if her active lifespan is followed by a period of at least six months of inactivity. In the data set, the average active lifespan of users that abandoned Twitter is 73.21 days, indicating that most of them are occasional users. In fact, over a total of $159,069$ accounts that abandoned Twitter

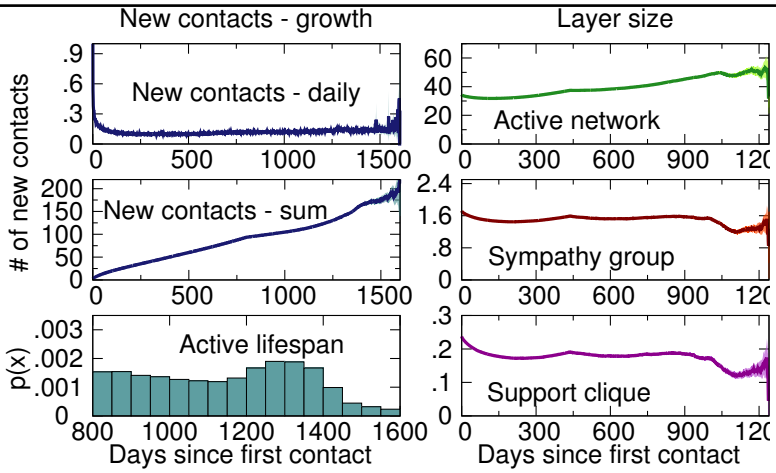


Figure 2.14: Ego network properties for regular users.

(i.e. 24.7% of the data set), 88.27% are occasional users, whilst only 11.6% are regular users and 0.13% are aficionados. From the distribution of the active lifespan of occasional users (depicted in the bottom left part of Figure 2.13) the reader can notice that there is a small number of accounts with duration between 50 and 365 days. Yet, there is a non negligible number of occasional users with a very short lifespan (i.e. $< 50d$). These accounts represent people that joined Twitter more than one year before the download, but that abandoned it after a short period of activity. This class of users can be seen as a sub-class of occasional users, who subscribed to Twitter only to "give it a try", but abandoned it very soon.

Ego networks evolution over time

Number of different alters contacted

The first result worth mentioning is that the number of new people that egos contact grows at a constant rate. This is true for all the categories of users and can be seen in the top left graphs in Figure 2.13, 2.14, 2.15. The graph labelled "New contacts - daily" depicts the number of new users contacted by egos during each day of their activity (averaged over all users still active at that day), whilst "New contacts - sum" represents the cumulative number of new users contacted by ego over time (again, averaged amongst all active users). From these graphs it is clear that, after a first phase in which ego contacts new people at a higher rate, this number quickly converges to a constant. The value of this constant is higher for

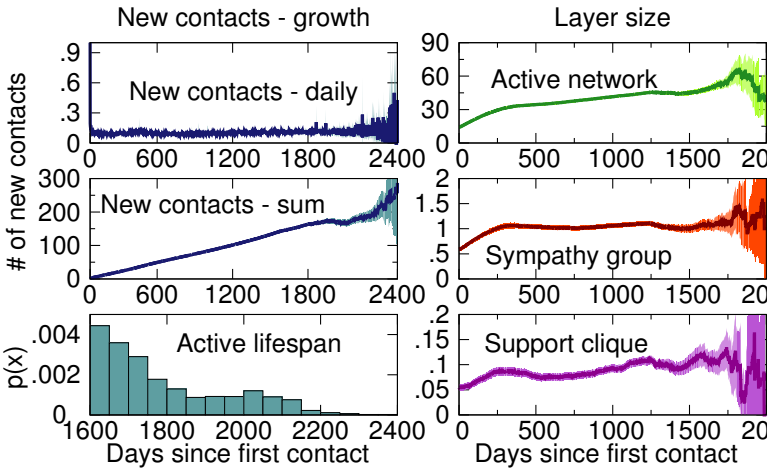


Figure 2.15: Ego network properties for aficionados.

occasional users than for the other classes. The mean over time is 0.222, 0.125 and 0.112 for the three classes, respectively. This indicates that occasional users have more dynamic ego networks, with a higher number of new social links added over time compared to the other categories. The total number of different people contacted by egos over time is, on average, about 200 and it is constantly growing, with little variation between the different classes, even though the duration of the ego networks changes considerably between classes. These results are in accordance with the findings in [104], where the authors found that users in RenRen (a popular Chinese OSN) are more active in creating new social links shortly after joining the network. The users eventually approach a constant number of edges created per time unit once most offline friends have been found and linked.

The presence of a constant growth rate is an important aspect of human social behaviour, indicating high dynamism in the ego networks of the users, that are constantly contacting new people rather than maintaining a limited number of stable relationships. This behaviour is confirmed by the analysis of the set of people actively contacted within the ego networks, reported below.

To understand how the constant addition of new contacts in the ego networks impacts on the communication level with the set of existing alters, the evolution of the size of the set of alters actively maintained over time has been studied, as reported in the following. Moreover, this section reports the analysis of the percentage of turnover (i.e. the degree of variation in the set of alters actively contacted) for the different layers in the ego networks.

Number of alters actively contacted

Even though the number of new alters contacted by egos increases over time, the number of alters that are actively maintained in the ego networks does not increase at the same rate. This fact reveals the presence of a turnover strategy within the ego networks, since the new contacts replace other relationships that are not maintained by ego. The size of the ego network layers are depicted in the right column of Figure 2.13, 2.14, 2.15, for the different categories of users. As far as occasional users are concerned, the size of all the layers significantly decreases over time. Specifically, the active network has a total decrease of 30.73%, the sympathy group of 45.91% and the support clique of 53.22%. Regular users show a different behaviour, with a considerable increase in the active network size (31.16% in almost 4 years), but with a decrease in the other layers (32.17% for the sympathy group and 30.42% for the support clique). It is worth noting that occasional users, compared to regular users, show a higher value of new contacts added in their ego networks daily and larger sizes in all the layers at the beginning of their lifespan, eventually approaching sizes compatible with the regular users. Aficionados show a considerable growth in size in all the ego network layers, even though the rate at which they contact new people is lower than for the other categories. These results highlight the different behaviour of the users in Twitter and indicate that occasional users have an initial boost of activity followed by a decrease or a sudden abandonment of the platform. Regular users and aficionados have a slower start, but they eventually increase the size of their active network over time. Aficionados even increase the size of their inner layers, indicating an investment in strong social relationships, maybe due to the longevity of such relationships, constantly reinforced through Twitter.

On average, the active network size lies between 30 and 80 for all the categories. This result suggests the effect of cognitive constraints of human brain in online environments, which limit the number of people that can be actively maintained over time, in line with the concept of the Dunbar's number. The small active network size, compared to offline social networks size found offline (equal to 132.5 [105]) can be related to the fact that Twitter is only a part of the complete social network of the users and the time spent on Twitter is still low compared to the time spent socialising in person, even though this discrepancy is constantly decreasing [46].

The lower growth rate showed by the sympathy group and the support clique compared to the active network (even negative for occasional users and regular users) suggests the presence of a strategy whereby people prefer dynamic ego

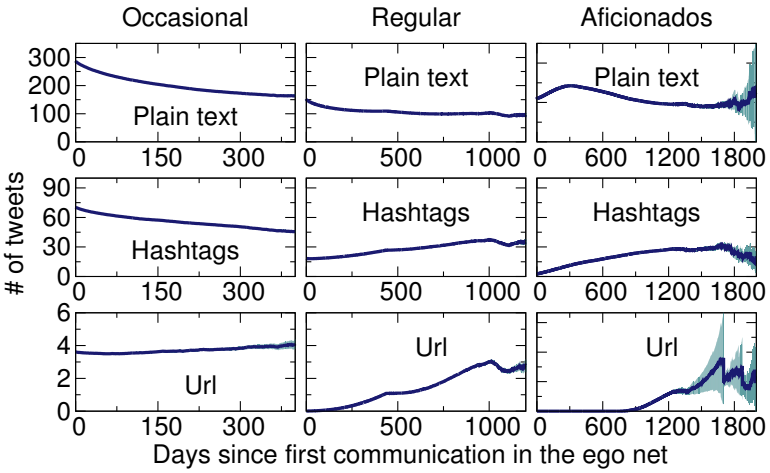


Figure 2.16: Non-direct communication divided by category.

networks formed of light-weight social relationships that give access to a larger amount of network resources [57], rather than more stable ego networks with stronger and well-consolidated relationships. However, for aficionados (i.e. users that spend a lot of time maintaining their social relationships in Twitter) this preference towards light-weight social relationships is way less marked, making their behaviour much more similar to the one highlighted in previous studies of social networks [105].

Finally, the rate at which egos contact new users is negatively correlated with ego networks growth rate, indicating that users spending a lot of their time adding new people to their networks do not have enough resources to maintain all these relationships over time and their layers inevitably decrease in size. This is in accordance with the idea that human social capacity is limited by cognitive constraints and going beyond the limits could even brake up a social network [88].

Non-direct communications

The rate of change over time of non-direct tweets (i.e. plain text tweets, tweets with hashtags and tweets with urls) has been studied for the different categories. The results are depicted in Figure 2.16. Occasional users significantly decrease the amount of non-direct tweets they send over time - apart from tweets with urls, although these are very limited. This category of users shows an initial boost of activity followed by a gradual decrease, as already found for direct communications.

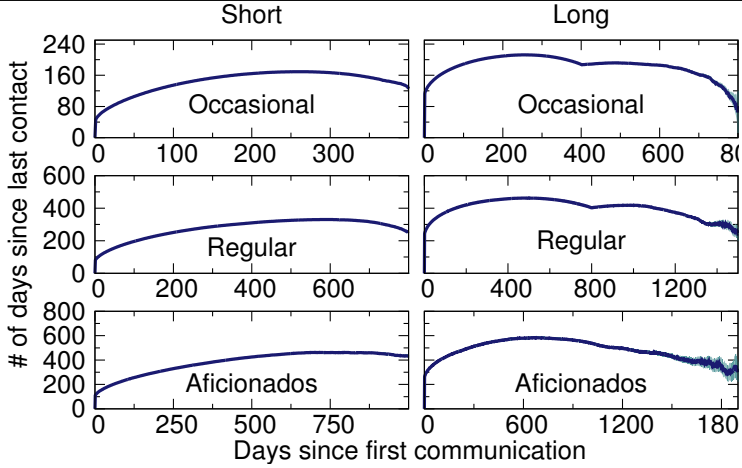


Figure 2.17: Days since last contact evolution over time.

Regular users show a much more stable trend for what concerns the number of plain text tweets, with a value asymptotically converging towards ~ 100 tweets sent in each one-year window. Yet, the number of non-direct tweets is noticeably lower than for the previous category, even though it is increasing over time. This indicates that regular users are less affected by an initial boost, and they rather have a slow start. Aficionados show a similar pattern, apart from plain text tweets, which show a peak in the first two years of their active lifespan. This peak could be due to an initial enthusiasm in the platform at a global level, since this category contains some of the oldest profiles in Twitter. After this initial phase, the number of plain text tweets converges asymptotically to a value similar to the other classes.

These results indicate that whilst some users abandon Twitter after a short period of time, the activity of the egos that continue to use the platform remain stable, rejecting the hypothesis of a convergence towards the OSN decline [89, 102]. This is in contrast with the results of [101], where the authors found that, in Facebook, users are more active when they join the network, decreasing their use rate over time. The present analysis reveals that this behaviour is true only for occasional users and that there is a non negligible amount of long-term users contributing to the survival of the OSN.

Evolution of personal social relationships

To better understand how personal social relationships evolve in Twitter, the average time since last contact has been analysed in terms of its change over time for

each single social link in the different categories. The social relationships in each category have been divided in “short” relationships, with duration shorter than half of the maximum duration of the category, “long” relationships, with duration longer than the same threshold. Figure 2.17 depicts the number of days since last contact between people involved in each social relationship (on the y axis) as a function of the time since the beginning of the relationship (x axis). From the figure the reader can notice that all the distributions show a “bow” shaped curve. This particular shape indicated that, on the one hand, social relationships have an initial phase in which they have a shorter time since last contact (i.e. higher frequency of contact) followed by a gradual increment. On the other hand, since some social relationships disappear as time passes, the remaining social relationships have shorter time since last contact, resulting in the gradual decay in the right most part of the graphs.

It is worth noting that there is a significant variation in the values of time since last contact in the different categories of users, with occasional users having lower values compared to the other classes. Once again, this supports the idea for which occasional users have an initial boost of activity, followed by abandonment or gradual decay.

Ego network turnover

Finally, the stability over time of each layer has been assessed for the different categories. To do so, the average Jaccard coefficient between separated one-year windows has been calculated for each ego network. To perform this analysis the number of ego networks in the data set has been further reduced, since at least two years of active lifespan are necessary to calculate the Jaccard coefficient between two different non overlapping one-year windows. Thus, $190,249$ ego networks with active lifespan greater than two years have been selected. The average Jaccard coefficients for the different layers are reported in Table 2.13 under the label “all ego networks”. The low values of Jaccard coefficient for all the layers indicate a percentage of turnover higher than 75%, with a maximum of 98.8% for the support clique of aficionados. This reveals that the average turnover in each layer is really high. Interestingly, the turnover in the inner layers is higher than the turnover in the active network. This result is in contrast with the findings on phone call records analysed in [85], where the authors found that for the top 20 ranking alters in ego networks - formed of social links weighted with the number of calls between people in a fixed time period - the turnover is lower than for the rest of the ego network. It is also worth noting that occasional users show higher stability

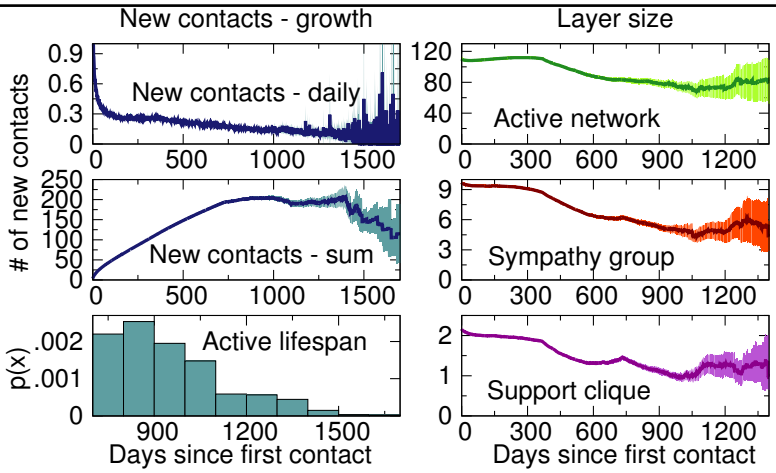


Figure 2.18: Ego network properties of structured ego networks.

compared to the other classes. This result could be explained by the fact that the longer the lifespan, the higher is the probability that the social relationships in the ego network change due to turnover.

The low values of Jaccard coefficient in the inner layers (i.e. 0.057 for occasional users, 0.024 for regular users and 0.012 for aficionados) could be influenced by the presence of small support cliques and sympathy groups, that for many egos do not even exist. For this reason the analysis has been refined calculating the Jaccard coefficients considering only users that always maintain a structured ego network, or, in other words, that show a non empty support clique in all the sampled one-year windows. The results are reported in Table 2.13 under the label "structured ego network". In this case the values on the Jaccard coefficient for the

Table 2.13: Average Jaccard coefficient of different network layers.

layer	Occasional	Regular	Aficionados
All ego networks			
active net	0.124	0.098	0.103
sympathy gr.	0.122	0.075	0.072
support cl.	0.057	0.024	0.012
Structured ego networks			
active net	0.191	0.190	0.193
sympathy gr.	0.287	0.309	0.362
support cl.	0.346	0.395	0.488

different layers are higher than in the previous case and are compatible with the findings in [85]. The values of the percentage of turnover of the active networks are similar for all the different categories and are about 81% (Jaccard coefficient ~ 0.19). For what concerns the other layers, the sympathy group show a percentage of turnover between 71.3% and 63.8%, whereas the support clique 65.4% and 51.2%. These results denote a behaviour similar to other social networks, where the inner layers contain stronger relationships that should be intuitively less affected by the turnover in the network. Nevertheless, as already found in [85], also the inner layers are strongly affected by turnover. The number of ego networks that show a turnover pattern similar to those found in other social environments is $10,307$, only 5.42% of the analysed egos. This is another strong indication that the dynamic properties of Twitter ego networks significantly differs from other social networks involving more traditional and dyadic communications. Remarkably, in structured ego networks the categories of users with longer lifespan have higher values of Jaccard coefficient, especially for the inner layers. This indicates that users that maintain structured ego network tend to reinforce their close relationships over time, instead of devoting their time to supporting weak relationships. Note that this is in accordance with the analysis of the evolution of the sizes of the layers over time for aficionados, previously discussed.

The properties of these $10,307$ ego networks have been further analysed applying the same technique used in Section 2.4.3. The results are shown in Figure 2.18. The active lifespan of these ego networks ranges between 730 and $1,749$ days. These are the minimum and maximum active lifetimes of ego networks in the data set that always presented a non-empty support clique. With this definition, users with behaviour similar to that showed in "offline" environments have been identified, where the support clique is maintained over time by the majority of people as the most important part of their networks.

Interestingly, the layers of the structured ego networks are larger than the average, resembling the layers found in Section 2.3, where a "super support clique" has been identified in Twitter, as a set containing one or two alters with very strong relationships with ego, perhaps a partner and/or a best friend. Also the sympathy group and the active network sizes are compatible with the findings in the previous section. Remarkably, all the layers decrease in size as time passes and so does the number of new alters contacted by ego. This could be explained by the presence of the initial boost of social activity of occasional users. Nevertheless, egos with longer lifespans prefer to consolidate their social relationships than adding new contacts, as indicated by the decrease in the top left graphs in Figure 2.18. This is in accordance with the results presented in the previous sections.

2.4.4 Discussions

This section presented a detailed analysis of the dynamic processes of ego networks and personal social relationships in Twitter. The results indicate that the dynamic properties of ego networks in Twitter significantly differs from other social networks studied in the literature in different research fields. On average, compared to more traditional social networks, Twitter presents a really high turnover. This fact suggests that the general behaviour of Twitter users is to maintain a lightweight ego network formed of weak social relationships suitable to maximise the amount of resources accessible trough the network and limiting the number of strong relationships. This type of user shows an initial phase of very high activity that is inevitably followed by a gradual decay or abandonment. On the other hand, a small but noticeable set of users prefer a "slow" start with a gradual increase of activity and more stable networks. This type of user shows ego networks much more similar to those found in previous analyses of social networks, with more stable inner layers and larger active networks (with respect to the first type of users). Moreover, the results also indicate that users that do not immediately abandon Twitter tend to use it at a regular rate in terms of direct and non-direct communication. This suggests that the hypothesised decline in the use of OSN might not be present, at least in Twitter.

Seen from an evolutionary perspective, the presence of a vast majority of users of the first type, and the resulting difference between the properties of their Twitter networks and conventional models of ego networks represents an interesting fact, since their behaviour seems to be adapting to the dynamism of the society, reflected in the need of new ways of acquiring information in a very dynamic way through OSNs like Twitter.

3

CAMEO Middleware

The analyses presented in the previous Chapter allowed to clearly identify the structural and dynamic properties of ego networks in OSNs. The results indicate that online and offline personal social networks share the same characteristics in terms of structural properties and size. Moreover, the findings presented in Section 2.4 suggest that the ability of social media to make the users constantly add a higher number of social contacts in their networks compared to more traditional communication systems in offline environments is of great importance to have access to a broader range of social resources (e.g. news, emotional support, new ideas, job opportunities) and makes OSNs effective media for maintaining social relationships. These properties are important to identify the requirements of future communication systems and for optimising MSNs.

To provide a better support for the development of MSNs, this chapter presents a novel middleware architecture for mobile devices called CAMEO (Context Aware MiddlEware for Opportunistic Mobile Social Networks), able to collect and share multidimensional context information, derived both from physical and virtual worlds. The aim of CAMEO is to provide MSN application developers with a set of APIs that give access to common functionality in terms of opportunistic networking facilities and context and social data management, to improve the social experience of the users whilst using MSNs, to enrich their interactions, and to stimulate the *collective awareness* concept by using personal mobile devices.

Context-awareness has become a fundamental requirement in the design of mobile and pervasive computing systems, trying to improve the impact of new technological solutions on the experience and quality of life of single individuals, of groups of people sharing interests and/or habits and, as a final goal, of the entire society. In the last few years, research organisations and IT companies are

investing over multiple application domains, from personal health to family life, environmental monitoring and social inclusion, each of them trying to contribute to the general *well-being* condition of the people and the society. All these domains are characterised by a high dynamism mainly due to: the mobility of the users, the interoperability of devices, the interactions between applications and the external environment and, last but not least, interactions amongst users and devices.

In this scenario, the notion of context for mobile and pervasive systems must be enlarged including both social and environmental conditions of the physical world in order to create autonomic, self-managing and self-adaptive systems, customised on the user's profile.

From a technical point of view, CAMEO allows personal mobile devices, which occasionally meet each other in a physical location, to automatically discover users' common interests, available services and resources through opportunistic communications. To this aim, it implements optimised networking protocols, resource management mechanisms and context data processing features. By exploiting these properties, MSNs are then able to generate and share content with peer-to-peer communications based on the characteristics of the users and their devices. In this scenario, the network of devices becomes a proxy of the networks of their human owners, generating thus *real-time MSNs* related to the real time needs of the users. Several application domains can benefit from this new paradigm, providing the users with new communication and comparison opportunities and generating additional context information that further enriches mobile systems' functionality. Figure 3.1 shows a group of application domains focused on the well-being condition of users.

This Chapter presents CAMEO architecture and prototypical MSNs created to show the potential of the middleware. Before delving into the details of CAMEO, some practical example of MSNs are presented to better understand what type of context information is relevant for MSN scenarios and identify middleware functionality and requirements to efficiently support the development of MSNs.

3.1 Application scenarios

As a first example the reader can consider tourists as the reference user category of a MSN application. Currently, tourists are more and more autonomous in planning their trips and sharing their experiences through the Web, but they need to search in advance for general information on dedicated websites (e.g. www.tripadvisor.com) or through their social network, in case some of their friends have useful information on that topic. In this way they can access information that

has been generated in the past and which is typically not tailored on the current user context. However, simply moving around the city, they can encounter (possibly unknown) people that have just visited interesting attractions and that can provide useful information, not available on the Internet and recently updated (e.g. “At 3PM there was 2 hours queue to visit Colosseum”, or “Yesterday dinner at La Maison was awful!”). In this scenario, context information is represented by the preferences of the user, her profile, location, attractions to visit inside the city, user-generated contents, but also information related to the surrounding users and their devices (the available types of connectivity, their interests, contents and so on). By collecting and managing all this information, CAMEO implements optimised content dissemination protocols through opportunistic communications, making the application able to help interested users to re-schedule their visit, optimise their time and avoid unpleasant experiences.

Another possible scenario is represented by the use of MSN as novel solutions for resource/information sharing in mobile systems. The reader can consider the emerging trend of Participatory Sensing applications [17]. Users generate contents related to air pollution, traffic monitoring, risky areas in a participatory fashion by exploiting their own resources (e.g. embedded sensors and camera) and then data are stored on a web server to be shared with others. These applications are typically web-based and exploit the single user with her smartphone as a mobile sensor node to collect useful data from the environment. The introduction of opportunistic communications and MSNs can further enrich participatory sensing applications through the cooperative use of resources belonging to physically connected devices (i.e. devices that are within the same wireless communication range). For example, if node A measures a temperature of $30\,^{\circ}\mathrm{C}$ but it is not able to measure the environmental humidity, it can ask node B for this information, since it has the humidity sensor embedded in the smartphone. Then, node A can locally correlate the collected information. In the same way, a node can delegate to another node the computation of a complex operation or the collection and elaboration of data provided by external sources unreachable from the local device. In this case the MSN offers an opportunistic computing service [42] and the definition of context is further extended considering information related to both the user and her device, in addition to data generated by external sources and those related to social interactions.

Several other application domains can be improved by exploiting MSN paradigm and in all the scenarios it is essential to collect, elaborate and integrate multidimensional context information in order to provide highly personalised, efficient and effective services in a really dynamic environment. CAMEO completely addresses

Figure 3.1: Well-being application domains.

this issue by providing a comprehensive definition and model of *well-being context*, and functionality to collect, manage and reason upon it on mobile devices, guaranteeing efficient and sophisticated context- and social-awareness features to MSN applications. Before describing CAMEO and the created prototype MSNs, the related work in the literature related to existing middleware solutions are presented, with particular attention to the management of social context, one of the hottest research topics in this area.

3.2 Related Work

In the last years context-awareness has become a key topic in pervasive mobile computing, evolving from the need of modelling external and objective conditions and situations, to the identification and modelling of subjective parameters (e.g. personal opinions, interests, character, feelings), largely influencing the interaction of the user with her device, and with the external world (both in terms of physical and virtual interactions). With the increasing success of social network applications, personal and social information of the user has further enriched the original notion of context, paving the way for a completely new area in context management and in the development of mobile social applications, known as *social-awareness*. Most of the work presented in the literature identifies social context only with information derived from virtual social interactions of the users by using social media, messaging applications, online gaming, and others. This information is used to: (i) identify and categorise social relationships between the local user and the others; (ii) define the profile of the user in terms of habits, contacts and interactions, in

order to personalise the application and provide appropriate feedback; (iii) to extend the social ego network of the user through trust mechanisms like FOAF[1]. The collection and management of this *virtual* social context is complex and resource consuming, not suitable for mobile environments.

Recent works like SCIMS [65] and Social Hourglass [63] are practical examples of infrastructures for virtual social context management. In SCIMS, in order to support the variety of social information and their semantic reasoning, the social context is stored and processed on a centralised server, requiring heavy loading procedures and long query processing times. These conditions are inconvenient for mobile environments and, even more, in case of opportunistic communications, in which a stable connection amongst nodes or with a remote server cannot be guaranteed.

In Social Hourglass, the authors present a multi-layered architecture for social context management based on the interaction of multiple applications with different roles (called *social sensors* and *personal aggregators*) aimed at collecting, filtering and personalising social data derived from multiple sources. This architecture requires the definition and development of a social sensor, as stand-alone application, for each type of social source (generally an application) the user is registered with (e.g. social networking service, chat), and a set of aggregators to tune collected data on the personal profile of the user. All the collected information is then stored in a persistent Social Knowledge Management Server (SKS) that, in this case, is designed as a structured peer-to-peer system [69]. This solution requires the execution of multiple concurrent applications for context elaboration generating a huge amount of data. In addition, as largely demonstrated in the mobile peer-to-peer systems literature, especially for mobile ad-hoc networks [47, 36], standard peer-to-peer architectures are not suitable for highly dynamic environments, in which the mobility of the users affects the network performances, causing also intermittent connectivity conditions.

In addition to performances issues of these solutions when applied to mobile environments, virtual social context information often does not reflect the actual behaviour and needs of the user in daily activities or in real situations. For this reason it is fundamental to collect and analyse social information derived from physical interactions between the users and the devices, integrate it with additional context related to the local user and the surrounding environment, and export them to mobile applications. In this direction, Yarta middleware [94, 93] introduces the concept of Mobile Social Ecosystems (MSE) as the set of interactions occurring amongst users and devices in a specific physical location. The middleware exploits

[1] http://www.foaf-project.org/

the current location of the user as a “social filter”, selecting relevant social information. Yarta considers each MSE as an independent entity, not analysing possible correspondences with users belonging to different ecosystems, and without any historical management of context information related to previously visited MSEs.

In a scenario characterised by high mobility, in which the physical network continuously reconfigures based on new and already known contacts, the social characterisation of the user cannot be limited to a single MSE. On the contrary, it must be related to all the visited MSEs, at least for a predefined amount of time. In addition, multidimensional context information is necessary to completely characterise the current situation a user is involved in, in order to optimise applications as much as possible. The work presented in this thesis about context and social-awareness in opportunistic networks started in the framework of Haggle project[4], initially focusing on context-aware forwarding protocols [34]. In that case context information, derived from the physical interactions of mobile users and their devices, were used to select, hop-by-hop, the best path from a source to a destination node, efficiently supporting intermittent connectivity conditions. Haggle project proposed a data-centric, event-driven architecture for the generation and management of opportunistic networks in terms of networking protocols, resource sharing, and mobile applications. However, real experiments and simulations demonstrated severe performance issues related to its software architecture [30].

CAMEO provides a much more detailed and comprehensive definition of context with respect to Yarta and Haggle. In addition, CAMEO implements the context exchange amongst neighbour nodes through opportunistic communications in order to: (i) detect both the local conditions of the user, her device and the surrounding context, and (ii) maintain a historical context profile of the interactions between the users. This allows CAMEO to identify the current situation the user is involved in and to implement optimised and automatic procedures for context- and social-aware content dissemination, contributing to improve a general collective awareness. CAMEO provides an API to MSN applications developers to allow a full access to context storage, elaboration and reasoning of context information derived from multiple and heterogeneous sources to further improve the applications.

3.3 CAMEO

This Section presents the general architecture of CAMEO middleware [20]. CAMEO is designed to collect and reason upon multidimensional context information, derived by the local device, the local user and their physical interactions with other

Figure 3.2: Well-being context.

devices and users. It provides a common API to MSN applications through which they can exploit context- and social-aware functionality to optimise their features. Validation and performance evaluation have been conducted through an experimental testbed, presented in the last part of this Section, along with a real example of MSN application.

3.3.1 CAMEO Well-Being Context

CAMEO introduces the general notion of *well-being context* to provide a two-fold awareness to the mobile system: context- and social-awareness. By correlating local information of both user and her device (like user's profile, interests, activities, local resources and running applications) with information derived from other users and devices, CAMEO is able to define new physical communication patterns and new social interactions amongst users. In this way, context information can be used to optimise both internal services, like networking and resources sharing, and MSN applications.

The ensemble of a user and her mobile device represents the core entity of CAMEO and MSN applications, and in this thesis is referred to as "node". The context of a node is defined as the integration of three main components: the *local* context, the *external* context and the *social* context (see Figure 3.2).

The *local* context represents all the information related to the local user and her mobile device including:

- the *user's personal profile* (i.e. all the information that describes interests and behaviours of the user, like type and place of work, habits, life style, timetables), generally provided directly by the user through dedicated interactions with the mobile device and its applications (e.g. through a digital agenda);
- The *resources* of the device, as the description of the local resources (e.g. battery level, storage capacity, CPU occupancy, tasks management, available connectivity, embedded sensors);
- The *embedded-sensing context*, as the information collected by phone-embedded sensors (e.g. GPS, cameras, accelerometer, sound sensor) that are generally used to characterise the activity of the user and her location;
- The *application context*, as the context information specified by each application running on the mobile device that can be used to optimise its performance. Some examples can be represented by specific characteristics associated with the application contents (e.g. type of content, user-generated tags), and physical and parameters for the correct execution of the application on the mobile device (e.g. the utility of the contents for a specific user, the available resources).

The *external* context represents the set of information collected by the mobile device of the user through a direct interaction with external sources. These sources can be fixed or mobile sensing stations (also wearable sensors), aimed at monitoring specific parameters (e.g. air pollution, traffic level, safety conditions), or remote services dedicated to data collection from one or more sensor networks. This type of information can then be integrated and correlated with specific components of the local context, such as phone-embedded sensing information and/or application contents directly generated by the users and their devices. In this way, the system is able to implement *participatory and opportunistic sensing*, aimed at increasing the quantity of information related to specific events and possibly improving the accuracy and fairness of environmental sensing information.

As far as the definition of CAMEO *social* context is concerned, both the identification of the physical interactions of the local node with other nodes in proximity and the exchange and collection of information belonging to the local contexts of the neighbour nodes are considered by CAMEO[2]. CAMEO is thus able to identify the membership of the local node to a *physical community*. In addition, since the local context of each node contains both *people-centric* and *content-centric* information (e.g. user's habits, personal profile and interests in specific topics, content

[2] Due to the temporal constraints characterising some information of the local context, nodes periodically exchange only a subset of the local information in order to build the social context of each node. More detail will be given in the next section.

types), the social context identifies also the membership of the local node to *virtual communities*. For example, tourists visiting Rome during Christmas holidays belong to the virtual community of tourists declaring interest in Rome and to the physical community of users visiting Rome in the same period of time and currently in proximity.

It is worth noting that, due to mobility, physical communities can have high dynamism, causing temporary and partial overlaps between physical and virtual communities. Connections amongst separate physical communities can be established through *traveller nodes* (i.e. nodes that, whilst moving, become members of multiple physical communities), which can be used as message ferries to disseminate contents to interested users/devices that could not be directly connected with the source of the information. This assumption relies on the definition of the HCMM mobility model presented in [35]. Figure 3.3 shows a simple example in which nodes belonging to the same physical community are characterised by different interests (highlighted with different colours), defining thus different virtual communities. Nodes moving between these physical communities can be used to carry data relevant to the different virtual communities they are in touch with. In this scenario, a *home* physical community is associated with each node as specific information of the personal user's profile. It represents the physical community in which the node spends more time and has stronger social links. The travelling condition of a node is then considered as a temporary visit to other communities. CAMEO distinguishes between the social context related to the *current* physical community of a node (a snapshot obtained by 1-hop context exchange amongst nodes), and the social context related to nodes encountered in *previously visited* physical communities, generating thus a historical characterisation of the social behaviour of the user (in terms of visited communities, frequency, etc.) and her social contacts.

CAMEO is in charge of collecting and reasoning upon the entire well-being context, in order to provide mobile application developers with context- and social-aware functionality aimed at optimising both the system performance and the user experience. It is also able to identify the *current situation* the local node is immersed in and to take autonomous decisions for the entire system optimisation.

Context Modelling

The choice of the context model to be implemented in CAMEO can heavily influence system performance in terms of processing overhead and response times of context evaluation in order to support the multidimensional and heterogeneous

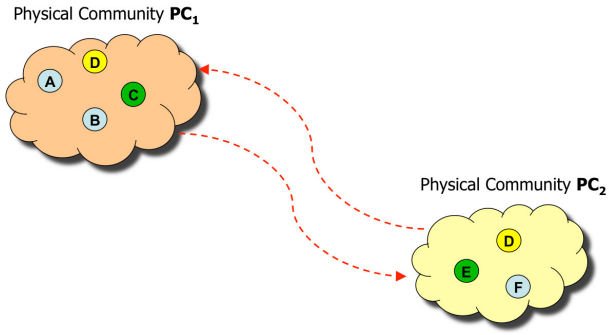


Figure 3.3: Example of physical communities with traveller nodes.

nature of well-being context. These represent a limitation both for the local context management and for opportunistic communications. In fact, since contact times of users and devices and related communication opportunities strictly depend on mobility, high processing times can cause an opportunity loss for data exchange amongst mobile nodes. This issue has been tackled by analysing advantages and drawbacks of context modelling and reasoning techniques for pervasive computing presented in the literature.

As showed in [29], context models can be compared in terms of efficiency (to access data and execute reasoning procedures), scalability and usability of the formalism. Specifically, approaches based on key-value pairs use a basic context representation resulting in a low-cost and easy to implement model. However, they present limited capability in defining different context types, relationships and dependencies, supporting limited reasoning, especially on context uncertainty. Ontological approaches build on key-value models by introducing the description of concepts and relationships to provide a semantic meaning to context data. The main issue in using ontology-based models is to find the correct tradeoff between the expressiveness of context data and the complexity of reasoning. The integration of Description Logics [37] partially solves this problem by providing optimised automatic tools for reasoning, but their use on mobile devices presents performance issues in terms of computational load, scalability and processing times with the increasing quantity of data to be modelled.

An alternative and interesting solution is represented by Context Modelling Language (CML), proposed by Henricksen et al. in [60, 59] as an extension of Object-Role based model (ORM). CML provides a formal basis for the representation of *object types* and *fact types*. A fact type represents a relationship between

two object types and it specifies the role assumed by the object type within the specific fact type. An object type is classified according to its origin (i.e. profiled, sensed or derived[3]) and persistence (i.e. static or dynamic). The context model is also enriched by quality metadata, like accuracy and freshness, cardinality and dependencies between fact types, and histories in order to support temporal analysis of the temporal evolution of the context. Furthermore, CML reasoning extends the evaluation of simple assertions typical of SQL-like queries by introducing a three-valued logic to query over uncertain information (i.e. the evaluation of an assertion as "User A is located at X" can provide different results if one or more location values are associated with the subject, generating thus a *possibly true* result). This model provides a high expressiveness of context information (comparable with ontology-based models), whilst implementing efficient reasoning techniques. For these reasons it has been chosen to represent CAMEO well-being context.

Figure 3.4 shows the CML representation of well-being context[4]. In this model the main entities and relationships of well-being context are highlighted, with particular attention to the interactions amongst local, external and social components. It is worth noting that the core of the model is represented by the local context and by the Person object type in particular. It includes all the types of information associated with the local user's profile. The Person object type is also involved in several fact types that describe the activity and location of the local user (both derived and sensed fact types), her relationships with the local device and the running applications, and her interests in content types associated with specific applications. In fact, the model is designed to support multiple applications (running on the same device and used by the same person), not necessarily belonging to the same application domain but sharing some common context requirements. However, since application requirements are not known a priori, each application must register to the middleware to specify its context types. In this way the model support context reuse even though the context evaluation cannot be completely decoupled from applications. In fact, it relies on additional fact types like "Application receives/generates Content" and "Person uses Application".

[3] *Profiled* in case of information supplied by users, *sensed* if derived from sensors or *derived* if obtained from one or more associations using a derivation function, such as a mathematical computation or a complex algorithm.

[4] Object types are graphically represented as ellipses and fact types as boxes with appropriate characteristics as detailed in the legend based on CML specifications. Different colours are not specified by CML. We use them only to highlight the interactions of object types belonging to different well-being context components as represented in Figure 3.2.

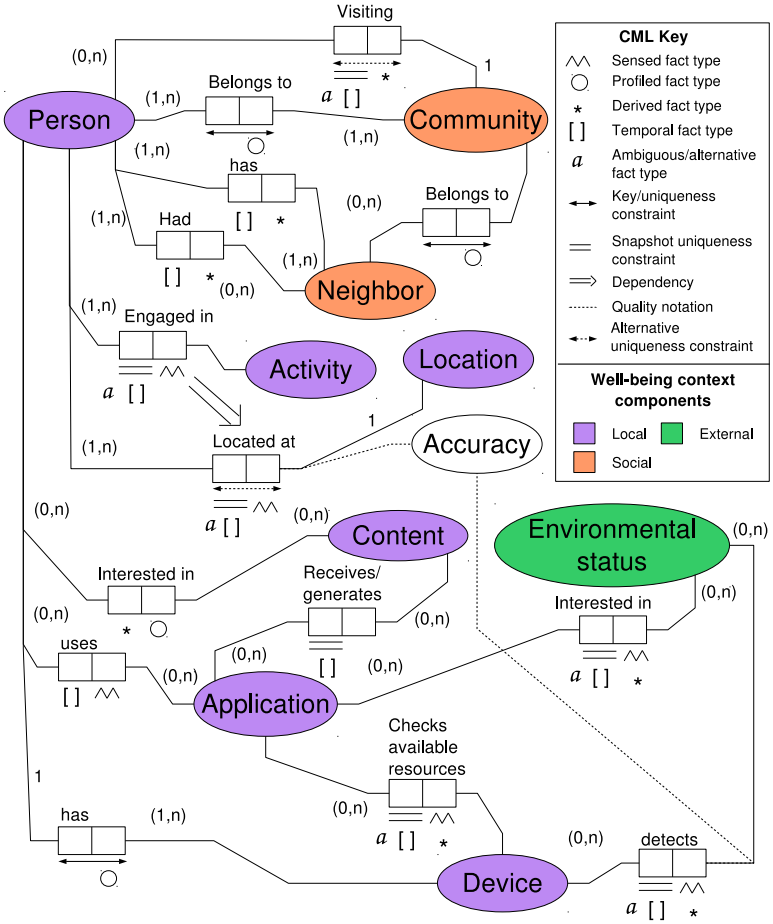


Figure 3.4: CML model of well-being context.

The Application object type represents also the join between local and external context components, together with the Device object type. In fact, “Application interested in Environmental status” represents the relationship between a specific application and the set of context types derived by an external environmental monitoring service (e.g. remote web services or external sensor networks), whilst “Device detects Environmental status” represents the capability of the local device

to collect and process environmental parameters derived from external sources and the context type of the result.

As far as the social context components are concerned, the Person object is again at the centre of the process, specifying the relationships of the user with both current and past neighbours. These fact types and the Neighbour object type result from the context exchanged amongst nodes in proximity. The Neighbour object represents a subset of the context types characterising a user, her device and applications, used to identify main characteristics, interests and capabilities of a remote node. Neighbours are also grouped by their "home" community. Therefore, two additional fact types can be defined for the social context component: (i) "Person belongs to Community", which describes the membership of the local user to a "home" community in terms of community id number and a list of neighbours belonging to the same community; (ii) "Person is visiting Community", which represents the set of information related to the community currently visited by the local node and the related neighbours (as the result of the context evaluation through the algorithm explained in the following).

By observing this model, it can be noted that the combination of some fact types can lead to a context change and the consequent adaptation and/or notification to the middleware. The occurrence of these combinations is generally defined as a *situation* and can be represented and analysed in several ways. In the next Section the current implementation of CAMEO situation awareness is presented (i.e. situation modelling and middleware adaptations).

CAMEO situation awareness

Situations can be defined as high-level context abstraction representing the elaboration of one or more context types and the consequent generation or triggering of system actions. Some work has been presented in the literature for situation modelling and reasoning [103, 44, 29]. Due to the heterogeneity of situations identified by CAMEO in the well-being context, it is difficult to identify a single model suitable for all of them. As a first step, the situations that represent a specific temporal state of one or more context types have been identified, and then they have been manually specified. However, the system could benefit from learning capabilities or probabilistic analysis in order to automatically recognise the occurrence of additional situations, based on the analysis of context information dynamically collected by CAMEO (e.g. information related to the mobility patterns and/or user activity).

By and large, two sets of situations can be identified: (i) *local* and (ii) *social* situations, involving local and social context components respectively. Both of them

imply a notification to upper-layer applications (previously registered to the specific event) or an adaptation of the middleware involving the execution of a consequent action. Several local situations strictly depend on the internal status of the main object types (e.g. available resources, new interest of the user) and on application requirements and constraints, triggering thus really specific actions. However, there are two general situations that significantly impact the behaviour of the middleware: *new registered application* and *application closed*. The former represents the starting point for the whole context processing related to the specific application, from the specification of the application context types up to the definition of the application specific situations (e.g. new content, new service). On the other hand, the latter situation, representing the end-of-running of an application, forces the middleware to a global context update, considering all the context types related to the application and their relationships.

Social situations are mainly independent of upper-layer applications and their context types. In fact, they represent general conditions of the system and the network in which all the applications are interested. Specifically, the following social situations have been identified in CAMEO:

- *New/Lost Neighbour*
- *New available content*
- *New available remote service*
- *Community change*

All these situations rely on the analysis of the context exchanged amongst neighbours. The *new neighbour* situation occurs when the local node receives a context message from an unknown node. This invokes the analysis of the message content in order to identify the subsequent situations (i.e. new available content and/or new available remote service, community change). To this aim, CAMEO compares the received information with that stored in the *current social context* of the local node (i.e. information related to the current neighbours). In case of a new service, CAMEO simply notifies upper-layer applications that can decide when and how to exploit it. As far as the availability of a new content is concerned, it is strictly related to the community change situation. In fact, if the local node experiences a community change and according to the reference mobility model used by CAMEO, it assumes the role of traveller node and it can decide to share its own resources to download available contents on behalf of previously encountered neighbours, based on the probability to meet them again in the future. Therefore, the occurrence of a community change, and the consequent availability of new content, triggers the evaluation of the utility of the content with respect to the inter-

ests of the neighbours belonging to the *historical* social context of the local node. Otherwise, if the local node still participates in the current community, CAMEO evaluates the utility of the content only for the local node, leaving to the final user the decision to obtain it.

The detection of a community change situation is currently implemented in CAMEO through the analysis of the *current social context* of the local node and, specifically, of the "home" communities declared by the current neighbours. The analysis of the social situations and related context processing represent the main innovation of CAMEO with respect to other middleware solutions, integrating the local context information with that directly derived from network communications.

Physical community detection

As previously explained, CAMEO identifies community change situations as the movement of a traveller node from a physical community to another. To identify the community change, every time CAMEO detects a new neighbour by receiving its context information, it executes the community detection algorithm that identifies the current community of the local node as the *home* community declared by the majority of the neighbours. The detection of this situation leads to: (i) update the social context of the local node (both current and historical profiles) and (ii) notify upper layer MSN applications.

To give a formal definition of the algorithm, consider N as the set of nodes of the entire network, and C as the set of physical *home* communities declared by nodes of N inside their local contexts. The characteristic function $I_c(n, c)$ is defined to indicate the membership of node $n \in N$ to the physical community $c \in C$, where $c \subseteq N$ as:

$$I_c(n, c) = \begin{cases} 1 & \text{if } n \in c \\ 0 & \text{otherwise} \end{cases} \tag{3.1}$$

Assuming also that $Neigh_{n,t}$ defines the neighbour set of node n at time t and that $C_{neigh_{n,t}} \subseteq C$ is the set of communities declared by those neighbours at time t, we can define the current physical community of node n at time t as

$$c_{cc}(n, t) = \underset{c_i \in C_{Neigh_{n,t}}}{\operatorname{argmax}} \sum_{n_j \in Neigh_{n,t}} I_c(n_j, c_i) \tag{3.2}$$

Therefore, every time the local node receives a message from a new neighbour, CAMEO computes (3.2) to discover the current physical community.

Context-based utility function for content dissemination in virtual communities

In MSN scenarios, context-based utility functions are generally designed to match content attributes with user interests in order to optimise content dissemination protocols in opportunistic networks. The main idea is to maximise content dissemination in virtual communities (reflecting common user interests) also by exploiting traveller nodes moving amongst different physical communities. To this aim, each application specifies a set of general context information characterising its contents (e.g. type, subject, size), the same information is also selected by the users as interests in the application contents. Therefore, the utility of a content represents the degree of interest of one or more users in that content.

In CAMEO, every time a community change is detected and/or a new content is available, the middleware evaluates the utility of the content with respect to the interests of the users in its social context. Specifically, in case of a community change, CAMEO computes content utility with respect to the historical social context, otherwise it computes the utility with respect to the interests of the local user. However, before selecting the content to download on the local node, CAMEO also checks resource availability on the node.

CAMEO exploits utility-based content dissemination framework proposed in [32]. Specifically, the utility function for a given content can be defined as follows:

$$U(c) = u_l(c) + \sum_{i \neq l} \omega_i u_i(c) \tag{3.3}$$

where $u_l(c)$ is the utility of a specific content for the local user, $u_i(c)$ is the utility for the i^{th} community the user is in contact with, and ω_i is a weight that defines the willingness of the user to cooperate with the i^{th} community (i.e. to spend its own resources to increase content availability for that community). In this way the local node offers its own resources to download contents available in its current physical community, that can be useful for users belonging to the previously encountered communities, assuming that in the next future the local node will visit again those communities.

In case there is no community change, CAMEO evaluates $u_l(c)$. This is used to notify upper-layer applications of the utility of the available contents for the local user, but the decision to download a content from a 1-hop neighbour is left to the direct interaction of the application with the final user. Otherwise, $u_l(c)$ is set to 0. In this case a traveller node can be characterised by different social behaviours influencing the content dissemination protocol. This is reflected in the definition of the weights associated to different communities. In [32] the authors

have introduced and investigated a set of social-oriented policies describing five types of behaviours: i) *Uniform Social (US)*, in which all the visited communities assume the same weight; ii) *Present (P)*, which favours only the current community; iii) *Most Frequently Visited (MFV)*, evaluating the frequency of community changes; iv) *Future (F)* and *Most Likely Next (MLN)* which require a probabilistic prediction of future visited community. Currently, CAMEO supports both US and MFV policies. Specifically, in case of US, all the ω weights are set to 1, so that CAMEO calculates the utility function for all the previously encountered communities, whilst in case of MFV, the ω weights are set proportionally to the number of visits of the local node to each community. Therefore, US policy facilitates data dissemination, because each node picks up all the contents found to be interesting for any previously encountered communities. On the other hand, MFV policy reduces the data dissemination rate optimising the local resources dedicated to preventive download procedure.

The same mechanism of utility functions can be used by CAMEO internal services to optimise their features (e.g. context-aware forwarding protocol, such as HiBop [31], and opportunistic resource sharing mechanisms).

The following sections describe in detail the software architecture of CAMEO, its implementation on Android OS, and the API provided to upper-layer application developers, with particular attention to context management features. In addition, a practical example of MSN application developed on top of CAMEO is presented, to better understand the middleware functionality and experimentally evaluate CAMEO performances.

3.3.2 CAMEO Software Architecture

CAMEO is designed as a light-weight and modular software architecture consisting of a single software package containing two sub-packages (as shown in Figure 3.5):

- **Local Resource Management Framework (LRM-Fw)**, aimed at implementing features strictly related to the interaction with the local resources of the device, both hardware (e.g. embedded sensors) and software (e.g. communication primitives and software libraries). It is also in charge of managing the interactions between the node and the remote sources (e.g. single external sensors, sensors networks, centralised repositories)
- **Context-Aware Framework (CA-Fw)**, aimed at storing and processing all the collected context information.

In addition, CAMEO provides an API towards MSN applications and it directly interacts with an external module for the user's profile definition called User Profile Module.

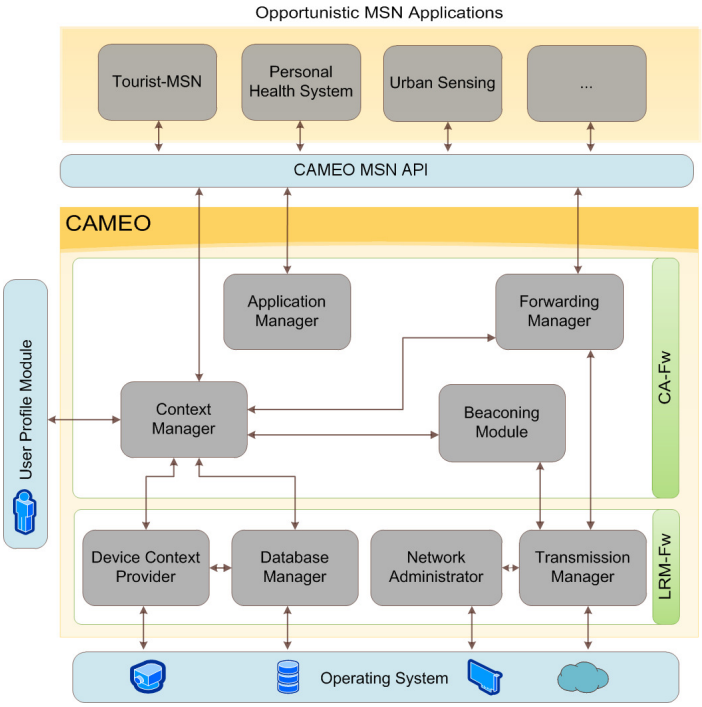


Figure 3.5: CAMEO software architecture.

Local Resource Management Framework

LRM-Fw is composed of three software modules:

Network Manager. In order to deploy real-time MSN, CAMEO allows mobile devices to exploit all the opportunities to communicate and exchange data through opportunistic communications. To this aim, the Network Manager interacts with all the available wireless communication interfaces (e.g. WiFi ad-hoc, WiFi infrastructure mode, Bluetooth) selecting the best communication medium under specific conditions. It is also in charge of notifying other interested CAMEO components

(specifically, the Transmission Manager) about the status of the connectivity between the local mobile device and her neighbours (e.g. WiFi link status and quality information, Bluetooth active/not active status).

Transmission Manager. After the Network Manager has selected the wireless interface for the transmission of a specific message (either middleware or application messages), the Transmission Manager is in charge of establishing the communication channel between a source and a destination node through the use of standard communication primitives (e.g. socket, TCP/UDP protocols and related parameters). It also receives notification messages from the Network Manager in case of link errors or disconnection events towards the message destination node.

Database Manager. It is responsible for the interaction of LRM-Fw with a SQL database that implements the CML model of well-being context.

Device Context Provider. It is in charge of collecting context data derived from internal components of the mobile device (e.g. embedded sensors, storage capacity level, battery level, resources consumption). Data specification and related parameters (e.g. sampling frequency for GPS or accelerometer, CPU occupancy threshold) are provided by the interaction of this module with the CA-Fw, following the directions provided by upper-layer applications or internal modules. In addition, Device Context Provider is able to manage data collected from heterogeneous sources (either internal or external to the mobile device) as specific support to participatory and opportunistic sensing services. It is worth noting that both embedded and external sensors are generally characterised by proprietary specifications and data formats. Thus, in order to guarantee the interoperability of CAMEO with those sources, a new light-weight standard for efficiently identifying and encoding heterogeneous sensing information on mobile devices, called *Sensor Mobile Enablement* (SME), has been defined. SME is implemented in CAMEO as a software library managed by Device Context Provider. SME is compliant with OGC Sensor Web Enablement (SWE) standards [81] designed for web services dedicated to collect sensing data. In this way CAMEO is also able to establish bidirectional communications with sensor web services and to forward this information to the opportunistic network. SME is described in detail in Section 3.4.

Context-Aware Framework

CA-Fw represents the core of CAMEO, being responsible for the collection, management and processing of all the context information (local, external and social) and the development of internal context- and social-aware services (e.g. forwarding protocols, resource sharing services). It is composed of the following software

modules:

Beaconing Module. It implements the periodical context exchange amongst 1-hop neighbours. This procedure allows CAMEO to discover new neighbours inside the current physical community and to build up and maintain the social context of the local node. As previously explained, only a subset of the local context components are disseminated through the network. Specifically, the device context is not exchanged with other neighbours due to its real-time nature. In fact, it represents local measurements of internal resources with limited temporal validity, and it is generally locally processed to evaluate the feasibility of specific actions (e.g. the local node receives a download request from a neighbour and it checks its local resources, such as its battery lifetime, before accepting and managing it). As far as the sensing context is concerned, only a summary of the available information on the local node is included in the beaconing message so that interested nodes and applications can directly request specific information. In order to avoid the periodical transmission of large quantity of data, the Beaconing Module implements an optimised data exchange procedure.

Forwarding Manager. It is responsible for the implementation of end-to-end communications. Specifically, it is designed to implement optimised forwarding protocols for opportunistic networks to successfully deliver a message to a multihop destination in case of intermittent connectivity (e.g. HiBOp forwarding protocol [34]).

Application Manager. It is in charge of establishing a communication channel between each MSN application and CAMEO through MSN API.

Context Manager. The main functions of Context Manager can be summarised in the following points: i) management of the well-being local, external and historical contexts; ii) interaction with the Database Manager to retrieve/store context data from/to the database; iii) implementation of algorithms and procedures for context reasoning, identification of specific situations and related middleware adaptations.

In order to provide efficient and reliable access methods to context information at run time, the Context Manager implements four separate data structures that partially reflect the database content:

well-being Local Context (wbLC): it contains the context information related to local context components divided in the following structures: *User Profile* derived from Context Manager interactions with the User Profile Module[5] (see Fig-

[5] This module has been designed externally to CAMEO in order to be implemented as a stand-alone application dedicated to the collection of user's personal information inde-

ure 3.5); *Application/Service Context* specified by each application developed on top of CAMEO and by single internal services; *Device Context* derived from the interaction of Context Manager with DeviceContextProvider. The latter contains all the information related to the device's status, such as sensor data and the available resources.

External Context (EC): it contains references to the external context types available in the database (e.g. pollution data, noise data, weather data). In this way, EC maintains a list of the available sensing services on the local node and a set of pointers to the respective data actually stored in the local database. Since the external context can be represented by a large quantity of data, it is not efficient to manage it at run time through simple data structures.

current community Social Context (ccSC): it contains the list of current 1-hop neighbours of the local node and their context information disseminated through periodical beaconing messages. The content of ccSC represents thus a snapshot of the physical community of the local node in a specific instant.

historical Social Context (hSC): it contains the list of communities previously visited by the local node associated with a timestamp as the temporal information of the last visit and a counter to maintain the number of visits in a predefined period of time. In addition, for each visited community hSC maintains the list of encountered nodes and their context. This information is essential to implement social-oriented policies for the evaluation of context-based utility functions. Information in hSC are periodically updated, maintaining at run time a subset of all the historical context based on predefined configuration parameters (e.g. time threshold, frequency). All the other information are stored in the database.

3.3.3 CAMEO APIs towards MSN applications

CAMEO APIs provides a full access to CAMEO context- and social-aware functionality for the development of MSN applications. Since communications between CAMEO and MSN applications are bidirectional, two distinct APIs have been defined for application requests and CAMEO notifications (e.g. messages, events, errors). In the following a high-level description of the possible interactions between MSN applications and CAMEO is presented. The complete specification of CAMEO APIs can be found in [19]. A practical example of MSN application and its implementation is then presented in the following.

---

pendently of the running applications and services. However, the information provided by User Profile Module are integrated in wbLC with information provided by other services and applications that are mainly related to the user interests, habits and so on.

- **Registration**. Each application must register to CAMEO in order to access its internal functions. During the registration a unique identifier is assigned to the application and a callback interface is established. Applications must provide the type of service they are interested in, so that CAMEO will be able to provide them with specific sensors data coming from both internal and external sources.
- **Application Context specification**. Each application specifies the set of context information relevant for its execution in order to be evaluated by CAMEO Context Manager. This information characterises the running application on the local node and it is disseminated over the network through the beaconing procedure as part of the local context.
- **Utility function evaluation**. Each MSN application can define an algorithm for the utility function evaluation. The utility function is expressed in the form of a set of criteria to be applied to a given content through logical operations. The application can ask CAMEO to evaluate the utility function by passing the utility function, the id of the content to be evaluated (stored by CAMEO) and the social policy for further details about social policies).
- **Message sending/receiving**. MSN applications can send/receive messages through peer-to-peer communications and they are notified in case of failure during a message sending[6]. Through these messages applications can send/receive messages using their own communication protocols, as well as messages to request the exchange of context data (both local and remote) and services.

CAMEO notifications towards MSN applications are implemented using the callback interfaces created during the registration procedure. To manage different concurrent applications, CAMEO maintains the list of registered and currently active callback interfaces assigning a logical communication port to each of them. A special communication port is used by the Context Manager for the context exchange over the network and its interactions with the other CAMEO internal modules. CAMEO notifications are related to the following events:

- **New application content discovery**. Every time CAMEO finds a new application content from a remote node it informs the interested application.

[6] In case of exchange of large application contents, system primitives for the standard message exchange can present overload problems due to the predefined memory size assigned to each Android process (32MB). To overcome this issue, CAMEO implements a file segmentation procedure splitting the requested content into fixed length data chunks (512Kb each). The correct reception of each chunk is acknowledged, so that the sender node can manage automatic re-transmissions of not acknowledged chunks.

- **New service discovery**. Every time CAMEO finds a new service available on a remote node (including sensor web services or external sensing devices) it informs the interested applications. In case of a sensing service, CAMEO informs the interested applications by sending them the description and the capabilities of the involved sensors.
- **New neighbour discovery**. CAMEO informs the applications when a neighbour enters/exits the 1-hop area.
- **New community detection**. CAMEO informs the applications when the community detection algorithm results in a physical community change.

3.3.4 Android implementation

To validate CAMEO functionality and evaluate its performances, it has been implemented on Android platform. Android has been chosen due to its constantly rising popularity and because it naturally supports Java-based distributed and concurrent applications in addition to an easy access to system information such as those related to embedded devices (e.g. GPS, sensors, camera). To better understand the implementation details, the definition of basic Android software components provided to developers are briefly discussed. For additional technical details the reader can refer to [1].

The Application Framework is the main Android component provided to the developer. On top of this framework, developers can design their own applications accessing the same APIs used by the core applications. An Android application is composed of four components: (i) *Activities*, representing Graphical User Interfaces; (iii) *Services*, which permit the execution of tasks in background; (iii) *Broadcast Receivers*, which listen for broadcast communication events amongst different applications and Android internal modules; (iv) *Content Providers*, which make a specific set of application data available to other applications. The activation of the three components and their following interactions are implemented through the *intent* mechanism: asynchronous messages exchanged between Application Framework components and containing the definition of the action to be performed.

Since CAMEO is a middleware platform supporting multiple concurrent applications, it has been implemented as an Android Service. Thus, MSN applications, designed and developed to interact with CAMEO, are implemented as Android applications. A single instance of CAMEO, running on a separate process, is shared amongst all the applications. To support the communication between CAMEO and MSN applications, an Android technique based on the inter-process

communication paradigm (IPC) has been adopted. The interfaces defined for IPC are based on Android Interface Definition Language (AIDL) similar to other popular languages based on CORBA or COM specifications (e.g. Java CORBA IDL [9] and C++ CORBA IDL [8]). The data that can be transferred through AIDL interfaces is limited to Parcelable objects, which are designed as high-performance IPC transport objects. This mechanism permits fast data exchange to the detriment of limited design flexibility due to the lack of standard functions for the marshalling of Parcelable objects.

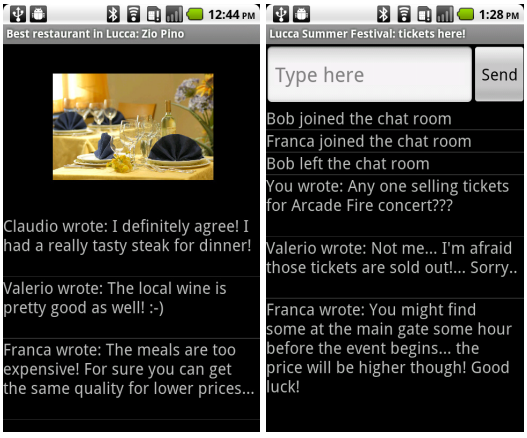
Tourist-MSN Application

Tourist-MSN [18] is a real example of MSN application developed on top of CAMEO. It is aimed at improving people experience during tourist visits by allowing individuals to create, collect and share useful information, related to geo-located points of interest (POIs). Contents are exchanged through opportunistic communications amongst mobile devices. Tourist-MSN provide users with two main functions:

- generation and sharing of multimedia contents, denoted as *post* and characterised by a title, a textual content (to comment or express impressions related to the POI), and optional information such as audio files, images or videos. Posts are divided into categories (e.g. event, cultural visit, transportation) in which users can express their interests.
- real-time textual communications through an *opportunistic text chat* inside a limited group of users in proximity. Each chat is identified by a title and a category.

Tourist-MSN specifies (through MSN API) the following information as application context to be disseminated over the network by CAMEO beaconing procedure: (i) title and category of each post and chat generated by the local user; (ii) user interests in specific categories of posts and chats. In this way, each node becomes aware of other nodes running Tourist-MSN in its current physical community and the list of available contents. Even though each node maintains a historical profile of neighbours and contents encountered in different physical communities, the management of a real-time chat is limited to the current physical community due to intermittent connectivity conditions characterising opportunistic networks. However, since posts distribution results in an asynchronous content exchange, Tourist-MSN provides CAMEO with the utility function algorithm designed to implement the context- and social-aware dissemination of posts amongst different physical communities. Moreover, since users can increase the content of a post

(a) List of available posts. (b) Creating new posts.



(c) Post Comments. (d) Chat.

Figure 3.6: Tourist-MSN GUI

by adding their own comments, CAMEO is also able to manage and distribute the content updates to the interested nodes.

Every time a new post or a new chat matching the interests expressed by the local user becomes available in the neighbourhood, CAMEO notifies Tourist-MSN with an event message. The application then notifies the user through the GUI about the availability of the new content and the user can decide whether to download the post (or join the chat room) or not. In case the local node experiences a

community change and there is one or more new available contents, CAMEO evaluates the Tourist-MSN utility function on each content with respect to the interests of users previously encountered in a different physical community by implementing US or MFV social-oriented policies. CAMEO provides then the application with a ranked list of the available contents and checks the feasibility of the related download procedures with respect to the local node physical requirements (e.g. memory availability, permanence time in the current community). In case a new content with a higher utility becomes available but additional resources are required, CAMEO will discard the content with minor utility. The main purpose of this mechanism is to maximise the utility of contents to be disseminated in the network following specific social-oriented policies. Figure 3.6 shows some screenshots of Tourist-MSN application running on Google Nexus One.

To better understand how to develop a MSN application on top of CAMEO, some code snippets related to the development of Tourist-MSN are presented. As a first step, Tourist-MSN starts using CAMEO establishing a connection between its Service component and CAMEO platform.

```
@Override
public int onStartCommand(Intent i, int startId, int flags) {
 Log.i(tag, "Connecting to CAMEO...");
 //Connect to CAMEO and start it if not already running
 bindService(new Intent("cnr.CAMEO.PLATFORM"),
   CAMEOConnection, Context.BIND_AUTO_CREATE);
 return START_STICKY;
}
```

Then, Tourist-MSN has to make a specific registration request to CAMEO in order to access CAMEO functionality (e.g. to send and receive messages over the network, to exploit content dissemination protocols). In the following code the basic instructions used by Tourist-MSN to require a registration are reported. The application provides a logical port and a callback interface used by CAMEO for event notifications and replies. Note that the variable called `CAMEO` represents the interface used by Tourist-MSN to interact with CAMEO. Once connected, Tourist-MSN can use local messages with predefined values to access CAMEO functionality (in the example below Tourist-MSN asks CAMEO for the local user context).

```
@Override
public void onServiceConnected(ComponentName name,
  IBinder service) {
 CAMEO = PlatformInterface.Stub.asInterface(service);
```

```
if (!registered) {
 try {
  resp = CAMEO.registerClient(PORT, callback);
  ...
  if (resp.getType() == LocalMessage.SUCCESS) {
   registered = true;
   key = (Long) resp.getContent();
   try {
    resp=CAMEO.sendLocalRequest(new LocalMessage(
    LocalMessage.GET_USR_CONTEXT, key));
    ...
```

As far as CAMEO notifications are concerned, Tourist-MSN must define the events for which it wants to be notified and the subsequent operations by using the `callback` interface defined during the registration procedure. In the example below, Tourist-MSN is notified by CAMEO when a new message is received, thus it must implement `onReceiveMessage` function specifying the operations to be executed for each type of message received.

```
private final CallbackInterface.Stub callback =
  new CallbackInterface.Stub() {
 @Override
 public void onReceiveMessage(LocalMessage packet,
   byte[] source) throws RemoteException {
  TouristMessage msg = (TouristMessage) packet.getPayload();
  switch (msg.getType()) {
   case TouristMessage.CHAT_MESSAGE:
   String chatMSG=(String)msg.getContent();
   ...
```

As a last example, the specification of the utility function by Tourist-MSN is presented. The application can request CAMEO to evaluate the utility function after a specific event (e.g. new available content, community change). In general, the utility function is expressed as a list of criteria to be applied to a given content and its properties through a logical operation. Each criterion is independently evaluated, then the results of all the criteria are combined together as a weighted sum (according to the given weights and the specific social policy), to obtain the overall utility of the content. As an example, in the following piece of code, Tourist-MSN specifies the utility function as a match between the property “category” of a post and the interest of the user, defined by the preference value “museum”,

and it requests CAMEO to evaluate it every time a new content is available in the neighbourhood. If the result of the logical operation is true, the criterion assumes the value specified by the last parameter (weight) passed to the function addEvaluationCriterion (1 in this case).

```
private final CallbackInterface.Stub callback =
  new CallbackInterface.Stub() {
 @Override
 ...
 //property = 'category', preference = 'museum'
 //opType = 'match', weight = 1
 utilityFunction = new ContentEvaluator()
  .addEvaluationCriterion(property, preference,
  opType, weight);
 socialPolicy = 1 //MFV policy
 public void onReceiveMessage(LocalMessage packet,
   byte[] source) throws RemoteException {
  TouristMessage msg = (TouristMessage) packet
   .getPayload();
  switch (msg.getType()) {
   case TouristMessage.NEW_POST:
   CAMEO.evaluateUtility(utilityFunction,
    msg.getId(), socialPolicy)
   ...
```

Experimental evaluation

In order to validate CAMEO functionality and its performance, experiments on a real testbed with up to seven Google Nexus One Android smartphones have been performed. The opportunistic network is represented by a WiFi ad-hoc network deployed by simulating the presence of disjoint physical communities, defining a priori their number and their labels to be declared through the context dissemination procedure. To simulate the separation between physical communities an IP-based content filter has been used. Traveller nodes periodically switch their IP-filters to move from one community to another. To be more realistic and to desynchronise intra-community movements, travellers wait for a random time before changing their community.

The functionality of both CAMEO and Tourist-MSN application has been assessed in terms of context and content messages exchanged through simple func-

tional tests in several scenarios involving an increasing number of devices (up to seven) and communities (up to three). Then, as far as the performance evaluation is concerned, three reference scenarios have been set up to measure, respectively, (i) the time necessary to CAMEO to detect a change in the physical community of a traveller node, (ii) the average time spent by CAMEO for the evaluation of the utility function for a given Tourist-MSN application content (by varying the number of interested users in the historical context), and (iii) the average content transfer delay measured by CAMEO by varying the content size. These three performance components provide additional information to CAMEO in terms of the average time that a node should spend inside the current physical community to permit a successful download of one or more contents. In addition, (ii) and (iii) performance components compared CAMEO performances with those of Yarta [93] and Haggle [4], respectively.

Community Detection Time

In order to evaluate the performances of community detection algorithm in a realistic scenario, different experiments have been performed by varying the number of involved nodes and physical communities. According to (3.2), a node realises that its current physical community is changed when the majority of its neighbours declare (through beacon messages) to belong to a different community with respect to that the node currently belongs to. To evaluate CAMEO reaction time to this event, three sets of experiments have been set up, reflecting a basic scenario in which one or more nodes belonging to the same community move towards another community. For each run, at time $t = 0$ nodes are distributed in separate communities maintaining this configuration for a fixed period of time T. Then, at time $t = T$ each node waits for a random period of time $\delta \in [0, T_b]$ (where T_b is the beaconing interval) before updating its IP filter to receive messages from nodes of the new community they move to. The community detection time is measured as the time period starting from $t = T + \delta$ and ending when CAMEO detects the community change event. The experiments are divided into three configurations:

1. Two disjoint communities (A and B), the first one with one single node (a_1) and the second with three nodes (b_1, b_2, b_3); node a_1 moves to community B.
2. Two disjoint communities assigning three nodes to community A (a_1, a_2, a_3) and four nodes to B (b_1, b_2, b_3, b_4); all the nodes of community A move to B.
3. Three communities (A, B and C), with one node assigned to A (a_1), two nodes to B (b_1, b_2) and three nodes to C (c_1, c_2, c_3); nodes of A and B move to C.

Table 3.1: Community Detection times for different experiment configurations and beaconing intervals.

Communities				Community Detection Time	
Comm	#H	#T	T_b	Avg	SD
2	1	3	500ms	390.51ms	123.29
2	1	3	1,000ms	704.10ms	63.30
2	1	3	5,000ms	4,290.72ms	197.66
2	3	4	500ms	539.11ms	228.99
2	3	4	1,000ms	1,097.54ms	183.77
2	3	4	5,000ms	5,136.96ms	870.32
3	1	3(2)	500ms	438.63ms	124.79
3	1	3(2)	1,000ms	903.1ms	114.29
3	1	3(2)	5,000ms	4,877.81ms	81.16

In all these configurations each node updates its IP filters at time $t = T + \delta$, starting to receive messages from nodes of the target community denoted as c_{target}. In the first two sets, $c_{target} = B$ while in the third set $c_{target} = C$. For each experiment, the community detection time measured by a node of community A (denoted as a_i) is reported.

All the experiments have been run with different beaconing intervals (i.e 500 ms, $1,000$ ms, and $5,000$ ms) and the results have been averaged over $1,000$ run tests.

Table 3.1 presents the detailed measures for each type of experiment. Specifically, the first column represents the number of communities participating in the experiment followed by the number of nodes of the starting community (#H) and the initial number of nodes of the target community (#T). When a third community is involved in the experiment (last three rows in the table), its size is specified within round brackets next to the cardinality of the target community.

In all the configurations the average community detection time is strictly related to the beaconing interval, since all the nodes use the same interval for their neighbour discovery procedure. However, the distribution of the community detection time computed by a_i after the IP filter update is highly influenced by the ratio between the number of nodes of c_{target} that a_i needs to discover as new neighbours in order to detect the community change (this number is equal to the number of neighbours in its home community plus one) and the total number of nodes belonging to c_{target}. The impact of this ratio can be observed in Figures 3.7, 3.8, 3.9 showing the comparison between the distribution of community detection times

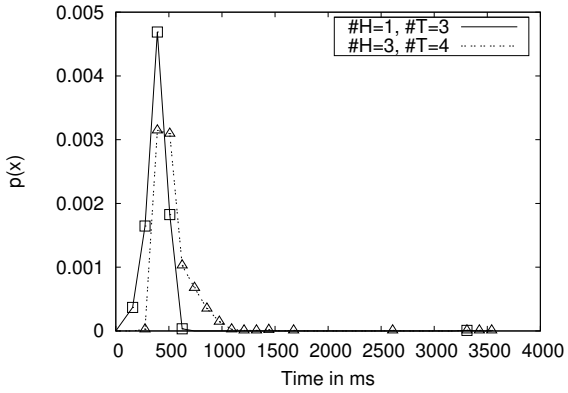


Figure 3.7: Distribution of Community Detection time $T_b = 500ms$.

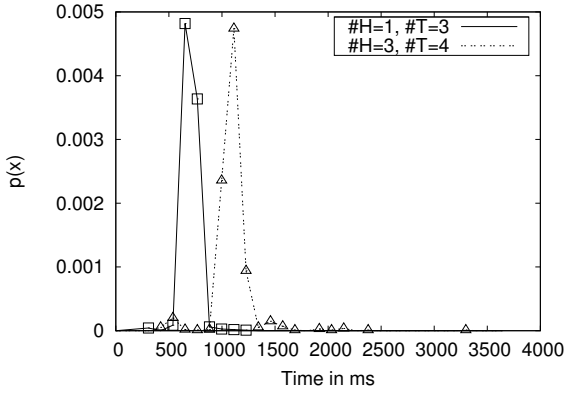


Figure 3.8: Distribution of Community Detection time $T_b = 1,000ms$.

for the first two sets of experiments ($\#H = 1$, $\#T = 3$ and $\#H = 3$, $\#T = 4$, respectively) measured for three different beaconing periods (i.e 500ms, $1,000$ms, $5,000$ms). Even though the average community detection time approaches the beaconing interval in all the experiments, the reader can notice that an increasing value of the previously described ratio implies an increase of the community detection time (see also numerical results in table 3.1). This is mainly due to the increasing minimum number of neighbours to be discovered in order to detect the community change. In addition, in case one or more beacon messages are lost, the a_i must wait for at least another beaconing interval to discover the target community, further increasing the community detection time.

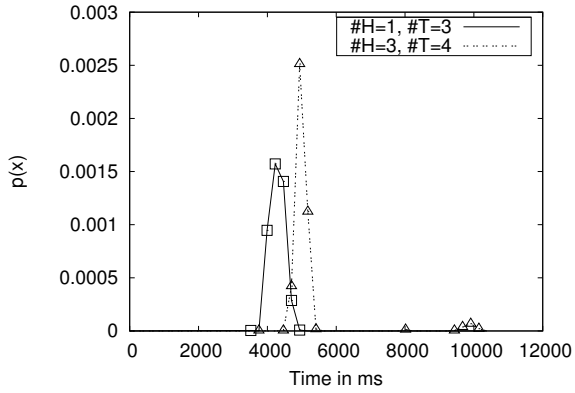


Figure 3.9: Distribution of Community Detection time $T_b = 5,000ms$.

In the third set of experiments, the presence of an intermediate community with less nodes with respect to the target community can generate on node a_i the detection of a temporary current community, different from the target community. This increases the community detection time. As shown in Table 3.1, in this case the increase of the delay is lower than in the case of two communities with the same number of nodes in the home and target communities. However, this behaviour depends on the speed of single nodes entering the target community, simulated in the experiments through the random interval of time waited by each node before updating the IP filters.

Utility Function Evaluation Time

In order to evaluate CAMEO performance in terms of management and elaboration of context information, a set of experiments have been performed to measure the time spent by CAMEO in evaluating the utility function for a given Tourist-MSN application content. To this aim, two different communities, A and B, initially composed of three and two nodes respectively have been set up. Two nodes of A are interested in an application content owned by a node of B. The third node is a traveller, which moves to community B and eventually comes back to its home community A. When the traveller reaches the nodes of B, CAMEO becomes aware of the availability of contents shared by the new neighbours and notifies Tourist-MSN application. Hence, the application requests a utility function evaluation to CAMEO for the new available contents with respect to the preferences of previously encountered nodes, passing the utility function and the social policy as parameters. In tourist-MSN, the utility function has a single criterion and a MFV

social policy. The time spent by CAMEO in the evaluation of the utility function of an application content is composed of (i) the time for the retrieval of the contexts of the nodes of A from the database through SQL queries, and (ii) the time needed to locally execute the utility function algorithm. These times have been evaluated by increasing the number of nodes interested in the content and, consequently, the number of contexts to be retrieved from the database. Specifically, ten experiment configurations have been considered, by increasing the interested nodes from 100 up to $1,000$ nodes, with steps of 100 nodes. 10 runs for each experiment have been executed, eventually averaging the results. Figure 3.10 shows the average time for context retrieval and the overall time to complete the utility function evaluation. Both quantities increase with the number of interested nodes with a greater impact of the data retrieval procedure. However, the time for data retrieval grows as a sublinear function, while the overall time grows linearly with the number of nodes. This is mainly due to optimised procedures implemented for database querying. In fact, considering that CAMEO spends an average time of 18.89ms to recover the context of a node with a single query, SQL queries for context retrieval have been grouped in sets of up to 100 nodes, reducing the impact of context history size on the utility computation. Of course, these values also depend on the social context size. In these experiments single node contexts of 9 KB have been used.

Results presented in this section can also be used to qualitatively compare CAMEO performances with those of Yarta presented in [93]. Actually, a complete comparison is not possible since there is no detailed description neither of context information stored by Yarta in its Knowledge Base, nor of the context size for each node of the Mobile Social Ecosystem. However, the authors show that Yarta requires a time in the range of $(2, 3)$ seconds to retrieve the context of a selected node by varying the Knowledge Base size from 100 to $1,000$ nodes. This procedure can be compared to a single query executed by CAMEO to retrieve the context of a node from the historical table in the database by varying the table size in the same way of the Knowledge Base in Yarta. CAMEO spends at most 30 ms in this procedure in case of $1,000$ historical contexts. In addition, the time required by Yarta to get the list of "known persons" from the Knowledge Base (grouped by sets of 50) overtakes 90 seconds in case of 100 nodes population and it further increases with the number of nodes in the Knowledge Base. As showed in Figure 3.10, CAMEO requires less than 6 seconds to retrieve $1,000$ contexts, largely outperforming Yarta.

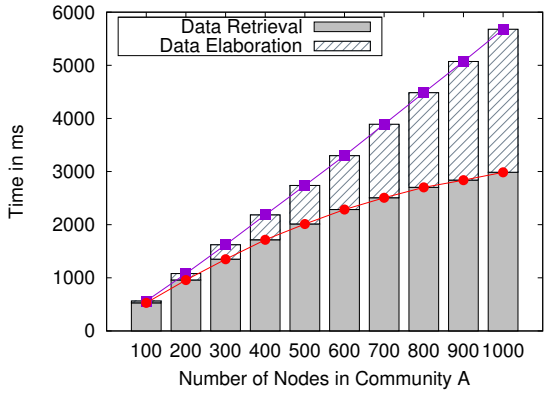


Figure 3.10: Utility evaluation time by varying historical context size.

Context-aware file transfer delays

File transfer represents a basic functionality for content sharing in MSN applications. To evaluate CAMEO performances in executing this procedure and compare it with Haggle platform, the file exchange experiments conducted in [30] have been reproduced. Specifically, the delay for a file transfer between two nodes connected through a 1-hop ad-hoc network have been measured for variable file sizes (from 28KB up to 6.5MB). The transfer delay is measured as the time interval starting when the application sends a request to CAMEO (or Haggle) to download a specific application content, which has been created with arbitrary size, and ending when CAMEO (or Haggle) notifies the completed transfer to the application. Figure 3.11 shows the file transfer delays averaged over 10 independent experiments for each file size with their respective 95% c.i. Specifically, Figure 3.11 shows CAMEO performances and the same results are compared in Figure 3.12 with those of Haggle measured in [30]. These results show the effectiveness of CAMEO in supporting MSN applications with an efficient management of context information, clearly outperforming Haggle. This is mainly due to internal characteristics of Haggle architecture, in which software modules designed to implement internal services (e.g. connectivity management, resource management, networking protocols) exploit a Publish/Subscribe mechanism through a centralised event queue. This permits the definition of interactions amongst all the software components to guarantee as much generality as possible in the definition of protocols, services and applications. However, the excessive modularity of this architecture resulted in a overload of internal messages that affected the scalability of the sys-

tem in terms of number of involved nodes and concurrent services active inside the Haggle platform. In addition, performance problems are further negatively affected by the centralised implementation of the event management procedure through a single kernel process becoming thus a bottleneck for both internal and networking communications. Haggle delays are about 7 times larger than those of CAMEO for the transfer of a 6.4MB file.

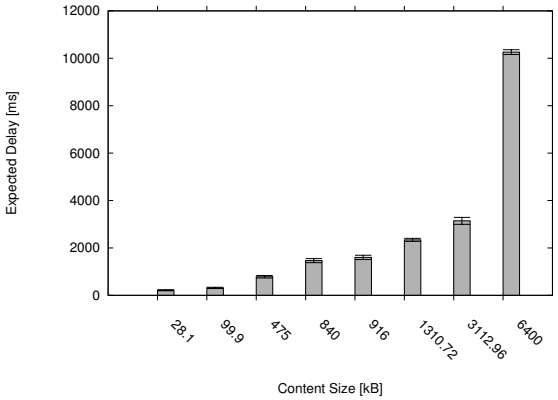


Figure 3.11: CAMEO performance.

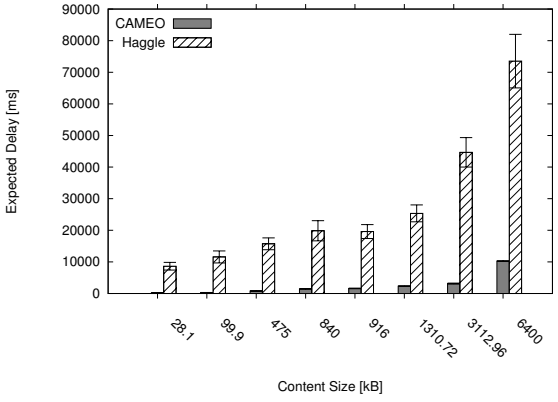


Figure 3.12: Haggle performance.

Another parameter that has been evaluated in this set of experiments is the memory used by CAMEO and by the application with respect to the total amount of available memory assigned by Android to each process. Results show that CAMEO uses $6,280$KB of which $3,640$KB are dedicated to the shared memory for the inter-process communication. This portion of memory is set by Android proportionally to the number of communicating processes. As far as the Tourist-MSN application is concerned, it uses $9,379$KB of which $5,339$KB are for shared memory. Consider that Android launches a dedicated instance of Dalvik JVM for each process with a maximum memory size of 32MB. Note also that the overall amount of memory used by CAMEO, the application and the Android operating system with its standard applications is $184,564$KB over a total amount of $416,132$KB available memory. This shows the light-weight implementation of this novel context-aware middleware architecture.

The three performance components showed in the previous subsections not only highlight the efficiency of CAMEO in real testbeds, but their aggregation also provides an estimation of the minimum time a node has to spend in a physical community to complete one or more download procedures. This information could be correlated by CAMEO with information on the local user mobility patterns. In fact, by knowing an estimation of the time the local user generally spends in a physical community, CAMEO could automatically decide to execute the download of a content or not, ranking contents both for their utility index and download time.

3.3.5 Discussions

This section presented CAMEO, a light-weight context-aware middleware platform aimed at allowing easy and efficient development of MSN applications in opportunistic networks. CAMEO is able to collect, manage and reason upon multidimensional context information, derived both from physical and virtual worlds, characterising user's profile, her social behaviour, the available services and resources, and the surrounding environmental conditions. Several application domains can benefit from the analysis and correlation of this context information, contributing to the general well-being condition of users and their society. For this reason, the well-being context has been defined as the ensemble of context information related to the local user, her device, her social interactions, and the external environment. A general CML model for well-being context has been proposed with a set of situations, involving both local and social context components, that lead to specific middleware adaptations and notifications to upper-layer applications. CAMEO defines a detailed API in order to provide a complete access to context- and social-aware

functionality to MSN applications. Results presented in this thesis showed the efficiency of CAMEO in collecting and managing well-being context and supporting the development of MSN applications through real experiments. Tourist-MSN application has been presented as a prototypical MSN application able to enrich tourists social experience during their visits. Tourist-MSN allows tourists to create on-demand social relationships with other tourists in proximity to exchange useful information when needed.

3.4 Sensor Mobile Enablement (SME): a Light-weight Standard for Opportunistic Sensing Services

This Section presents *Sensor Mobile Enablement (SME)*, a light-weight standard for efficiently identifying, encoding and decoding heterogeneous sensing information on mobile devices [22]. SME has been developed to allow CAMEO to enlarge the opportunities to collect sensor data from external sources and to provide MSN applications and services with a more complete view of the environment. This Section describes SME features and advantages, as well as its integration with CAMEO. Moreover, SME performance has been tested on Android smartphones. The results highlight that SME does not heavily impact on the performance of mobile systems, whilst efficiently supporting new opportunities for opportunistic sensing services.

SME is a light-weight version of SWE framework [10], released by the Open Geospatial Consortium (OGC), suited for mobile devices and compliant with the standard. SWE is composed of a set of interoperability interfaces and metadata encoding (based on XML schemas) that enable real time integration of sensing information into a server web infrastructure. SWE defines three standards for encoding, respectively, sensor descriptions (Sensor-ML), observations and measurements (O&M) and transducers (TML). In addition, it provides four web service interfaces for accessing the related information. Generally, the most used interface is the Sensor Observations Service (SOS), that enables applications to access observations and sensor system information stored on remote web servers. SWE relies on the general assumption that each sensor is a web-connected device and all sensed data must be remotely managed through a web service. To this aim, SWE defines sensor discovery procedures, processing and correlations of sensor observations in a completely centralised way. SWE is currently becoming the reference standard for remote sensing services and for the emerging paradigm of Internet of Things. In fact, OGC formed a new standard working group called *Sensor*

Web for IoT [11] aimed at developing new standards based on Web of Things protocols whilst also leveraging the existing SWE standard. However, to consider the Web as the only reference scenario does not reflect the actual distributed nature of current mobile systems, especially involving smartphones as complex sensing devices.

Today smartphones are able to produce several sensing information, both as raw sensing data derived from phone-embedded sensors and application level sensing events, derived from the correlation of multiple sensing data originated by local and remote nodes (e.g. wireless proximity information, crowded places). Through opportunistic communications, mobile devices can also share their own resources (e.g. processing and sensing capabilities) extending the use of local sensors also to remote nodes. The personal mobile device of each user is thus able to generate and share useful sensing information not necessarily passing through a web service. However, to maintain the interoperability of opportunistic sensing services with external sensors implementing SWE or with centralised SOS servers, mobile devices (and CAMEO) must be able to efficiently elaborate SWE data.

Currently, mobile operating systems do not support the efficient elaboration of large XML files, such as those defined by SWE, due to limited hardware and software resources. For these reasons, SWE standard cannot be used as they are on mobile devices and this currently represents an open issue in this research area. Limited work on this topic has been presented in the literature and the extreme inefficiency of managing SWE standard on mobile devices is in most cases at the centre of the debate. Specifically, [90] revealed that processing SOS XML files on mobile devices can be 30 to 90 times slower than their elaboration on a PC. To overcome this limitation, the same authors proposed to use different file formats to encode SWE messages on mobile devices [92] trying to reduce the overhead of XML processing and their impact on the network usage. Even though some formats (e.g. JSON [5], EXI and EXI with compression [14]) are able to slightly improve the system performances, they introduce an additional overhead due to the local conversion of SWE data into the new format and vice versa. In this section, in addition to SWE performances limitations, technical limitations are showed to prevent SWE implementation on mobile devices, especially on Android OS.

3.4.1 Technical limitations of SWE on Android

Implementing SWE on mobile devices presents two main issues: (i) the lack of efficient software tools for managing XML files on mobile OS, and (ii) the large size

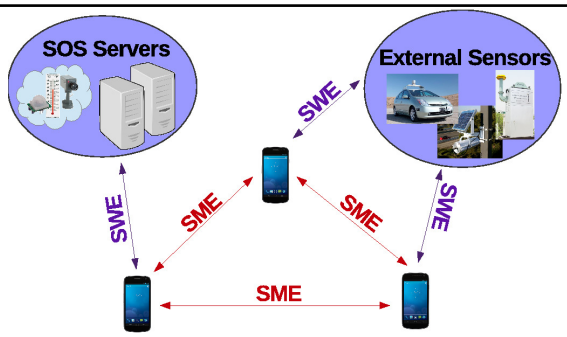


Figure 3.13: Smart city scenario involving heterogeneous sensing devices.

and complexity of SWE XML schemas (most of them including several dependencies). To manage and process XML files, the first investigated approach was XML file parsing by using standard APIs like DOM (tree-based) [13] and SAX (event-based) [2]. In both cases, application developers have to know a priori the specific XML file structure to implement serialisation and deserialisation procedures, manually selecting relevant tags and values (e.g. sensor descriptions and measurements fields). As an alternative, there exist some software tools able to automatically serialise and deserialise XML files into a predefined set of Java classes, reflecting the specific XML file structure (i.e. XML data binding). In this way, the application development is completely independent of the XML file interpretation and a unique library can be provided for a specific set of XML files. Nevertheless, the large size and the complexity of SWE XML schemas make really difficult the use of this procedure. In fact, even desktop tools like JAXB [12], XMLBeans [15] and XBinder [6], are not able to provide a complete support for the automatic generation of SWE classes (as witnessed by [7]). Specifically, XBinder has been tested, which is the only tool able to generate Java classes for Android, but the results indicated that it is able to support only Sensor-ML schemas.

In order to overcome these limitations, SME has been designed as a Java library able to support XML data binding of a light-weight standard compliant with SWE, customised for opportunistic and participatory sensing services.

3.4.2 Sensor Mobile Enablement (SME)

In order to completely understand SME potential in mobile environments, consider a *smart city* reference scenario as the one depicted in Figure 3.13. Several heterogeneous sensing devices ca be envisioned around the city, either located at

fixed positions (e.g. traffic lights, panels) or in movement (e.g. through vehicular networks, as buses carrying air pollution sensors), thousands of users with their smartphones, equipped with additional embedded sensors, and remote sensor data repositories. In this scenario, smartphones and mobile users can dynamically aggregate forming social networks, making users able to generate and share useful sensing information through opportunistic communications. At the same time, smartphones can interact with external sensing devices (e.g. sensors) and remote sensor web services (e.g. SOS server collecting air pollution data of the city). To permit the interaction of all these heterogeneous sensing systems it is necessary to define a common standard suitable for mobile device management and elaboration.

SME defines standard models, compliant with SWE, to encode information related to both phone-embedded sensors and application-level sensing events. In this way it permits to implement opportunistic sensing services and their integration with external sensing components. As far as SME Android implementation and CAMEO integration are concerned, a set of primitives for efficient XML data binding of both SME and SWE XML files and for the interaction with SOS servers (and the efficient elaboration of related data) are provided to mobile application developers. In order to reduce XML processing and transmission overhead on mobile devices, SME data structures are based on a subset of SWE schemas, appropriately selected for mobile environments. Specifically, SME mainly refers to Sensor-ML and O&M as relevant standards for describing sensors, their possible operations and the related measurements, and SOS interfaces to support mobile access to sensor web services. Starting from sensor descriptions and observations derived from real SOS servers, a subset of SWE XML tags relevant for mobile sensing services have been defined, along with the related XML schemas. SWE schemas reduction does not affect the interoperability of SME with the original standard. In fact, experimental studies on a sample of SOS servers in [91] witnessed that most of them actually use less than 30% of SWE functionality. Moreover, SME defines data structures for encoding information related to phone-embedded sensors, which are not included in the original SWE. In fact, currently the only support to the development of mobile applications involving phone-embedded sensors is the implementation of operating system APIs, which allow developers to recover a predefined set of information (e.g. sensor type, vendor) and to register to the sensing event specifying the desired sampling frequency. SME defines ex-novo XML schemas for all phone-embedded sensors derived from the interaction with Android APIs, and implements a set of Java classes for data binding operations.

These schemas contain the minimum set of tags to be compliant with SWE standard.

As far as the interaction of the mobile device with external SWE sensors and repositories is concerned, SME implementation presents a two-fold functionality. On the one hand, it implements an efficient data binding procedure of complete SWE XML files derived from download operations. In fact, to reduce XML processing overhead additional fields not defined in SME classes are discarded. On the other hand, SME converts local objects into SME XML files, compliant with SWE, allowing mobile device upload operations to remote servers.

In this way, SME is able to overcome technical limitations of using SWE on mobile devices and to efficiently support the development of opportunistic and participatory sensing services, maintaining their interoperability with SWE standard services.

3.4.3 SME and CAMEO

Phone-embedded sensors represent additional sources of information and sharable resources on the opportunistic network. In fact, nodes running CAMEO can ask their neighbours for already available sensor measurements or for specific sensing operations (e.g. a node not having pressure sensor can ask to one of its neighbours to measure the related data, even if the remote node is not interested in it). To this aim CAMEO, by interacting with SME library, extends the context of the local node by introducing the notion of *sensing context*. In fact, CAMEO directly retrieves sensor information through operating system APIs and exploits SME data structures to initialise the sensing context. Sensing context includes the main information related to the available local sensors (i.e. sensor descriptions including both hardware and software capabilities) and their measurements (i.e. observations) and it is disseminated on the network through the beaconing procedure. In this way, nodes running CAMEO have a complete view of available contents and resources amongst their neighbours[7].

SME definition of general data structures for descriptions and observations of embedded sensors has been designed to support both existing and future integrated devices, supporting thus heterogeneous sensing components even inside the same mobile device. In addition, SME implementation allows CAMEO to interact with external sensing components based on SWE standard, like independent

[7] Naturally, the successful implementation of opportunistic resources sharing mechanisms is subject to dynamic constraints of the involved nodes (i.e. local resources availability to host remote service requests).

sensors or SOS services. In this way, CAMEO can support both standard upload-/download operations to/from SWE services and export local node sensing capabilities to external entities. This permits to include complex sensing systems, like smartphones or tablets, into the emerging paradigm of Sensor Web for IoT [11].

Within CAMEO, the Device Context Provider is in charge of interacting with Android APIs for embedded sensors management. It directly inherits SME data structures to create Java objects that represent sensor descriptions and observations according to Android information. Java objects are then passed to the Context Manager, the core of the middleware, which manages and elaborates all context information. The same objects are also included in CAMEO messages in case of data exchange over the opportunistic network, amongst nodes running CAMEO. This allows CAMEO to further optimise sensor data exchange over the opportunistic network simply transmitting serialised Java objects instead of SME XML files. On the other hand, Java objects are converted in SME XML files if they must be transmitted to independent nodes.

Referring to CAMEO interactions with external SWE services, CAMEO APIs include SWE standard interfaces (e.g. SOS GetCapabilities, GetDescriptions, GetObservations). When a node running CAMEO receives SWE XML files from the network, the Context Manager exploits SME data binding procedure to filter relevant information and deserialise the file. Instead, in case CAMEO and/or upper-layer applications need to export local information as SME XML files, the Context Manager uses SME deserialisation procedures. Even in this case, CAMEO reduces the data transmission towards SWE services, since SME XML files are light-weight codifications with respect to SWE standard.

Therefore, the introduction of SME as reference standard for mobile sensing applications largely extends CAMEO functionality, both in terms of additional sensing opportunities in mobile networks and interactions with remote standard solutions. In addition, the use of CAMEO as middleware platform to support multiple context-aware mobile applications presents the fundamental advantage of local sharing of multi-dimensional context information amongst different applications running on the same node. In this way, independent mobile applications can access and correlate heterogeneous context information, provided both by local and external services.

3.4.4 Experimental results and Performance evaluation

In order to evaluate real performances of using SME on mobile devices, a set of experiments have been performed on Google Galaxy Nexus smartphones

(HSPA+), equipped with Android 4.1.1 (Jelly Bean) and 1.2 GHz dual-core processor. Google Galaxy Nexus includes six hardware sensors and seven software sensors (as highlighted in Table 3.2). Hardware sensors are physical components integrated into the device, measuring specific environmental properties. Software sensors represent the elaboration of data derived from one or more hardware sensors. Android manages hardware and software sensors without distinction. It defines Sensor objects as the abstraction of sensing operations, involving one or more hardware sensors. Sensor objects are characterised by: event type, event name, event description, measurement unit, resolution, sensor name, sensor vendor, maximum range, maximum sampling frequency, consumed power. SME sensor descriptions are defined as data structures reflecting SME adaptation of Sensor-ML standard and they are populated by Sensor object values. As far as data derived from sensor measurements are concerned, Android generates SensorEvent objects according to the sampling frequency. A SensorEvent is characterised by: accuracy, sensor name, timestamp, and values. SME sensor observations are defined as data structures based on SME adaptation of O&M standard and they are populated by Android SensorEvent characteristics.

Through real experiments SME performance has been evaluated in terms of:

- SME library size with respect to desktop implementations.
- XML processing overhead on mobile devices related to serialisation and deserialisation procedures of both SME and SWE files.
- impact of SME processing on battery consumption.

SME size and XML processing overhead

As a first result it is worth noting that SME library, implementing a light-weight version of SWE standard, largely reduces the size of files in current desktop implementations, such as *JAXB for OGC* project [7]. Specifically, SME requires $38.9\mathrm{KB}$ for classes definitions and additional space for the use of a data binding tool (in this case SIMPLE [3], that requires $384\mathrm{KB}$). On the other hand, the library provided by *JAXB for OCG* requires $3.43\mathrm{MB}$ just for classes definitions, although not supporting the entire SWE standard.

In order to evaluate XML processing overhead introduced by SME standard and data binding procedures, the classification of XML files provided by W3C [99] has been taken as a reference. A measure called *Content Density (CD)*, representing the ratio between the value size of XML attributes and elements (real content of the file measured in number of characters) and the total size of the XML file (content and XML overhead) has been used to classify the files. XML files are

Table 3.2: Google Nexus embedded sensors.

#	Sensor Name	Type
1	Sharp GP2A Light Sensor	HW
2	Sharp GP2A Proximity Sensor	HW
3	Bosch BMP180 Pressure Sensor	HW
4	Invensense MPL Gyroscope	HW
5	Invensense MPL Accelerometer	HW
6	Invensense MPL Magnetic Field	HW
7	Invensense MPL Orientation	SW
8	Invensense MPL Rotation Vector	SW
9	Invensense MPL Linear Acceleration	SW
10	Invensense MPL Gravity	SW
11	Google Rotation Vector	SW
12	Google Gravity	SW
13	Google Linear Acceleration	SW
14	Google Orientation	SW
15	Google Corrected Gyroscope	SW

classified into two main categories depending on CD value: *High CD* if CD $> 33\%$ and *Low CD* if CD $< 33\%$. *Low CD* files are additionally divided into *Large* files if size $> 100\mathrm{KB}$, *Small* files if $1\mathrm{KB} < \mathrm{size} < 100\mathrm{KB}$ and *Tiny* files if size $< 1\mathrm{KB}$.

In the performed experiments, sensor descriptions and observations derived from local embedded sensors and remote SOS servers have been taken into account. Specifically, three reference scenarios have been envisioned:

1. Two nodes meet each other and they exchange their own sensor descriptions and observations encoded in SME standard.
2. The local node encounters up to 10 remote nodes and receives their sensor descriptions and observations encoded in SME standard.
3. The local node contacts a SOS server asking for a single observation including an increasing number of values (from 0 to $50,000$ values related to independent sensor measurements). The received observations are encoded in SWE standard. In a second step, the local node converts the elaborated data in SME standard in order to send it to other remote nodes.

As far as the first scenario is concerned, Figure 3.14 depicts the time required by SME library to deserialise the description of each local sensor from SME XML files to Java objects. Sensors are identified by sequential number as in Table 3.2. The first time the local node applies the deserialisation procedure to a sensor description, the processing time includes also the time necessary for loading SME

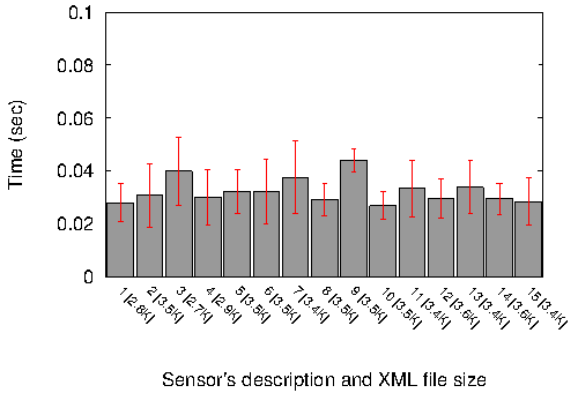


Figure 3.14: Scenario 1: mean deserialisation time.

library into memory. On average, the total time is equal to 0.618s with a 95% c.i. equal to $(0.521\text{s}, 0.7150\text{s})$. Instead, if the library is already loaded, deserialisation time for each embedded sensor description is in the range $[0.027\text{s}, 0.044\text{s}]$, as showed in Figure 3.14. XML file size of each local sensor description ranges between 2.7KB and 3.5KB with CD between 9.8% and 30.1%. These results have been compared with the time needed by SME library to deserialise a real SOS sensor description. The size of the original SWE codification of the used SOS sensor description is 10.5KB with CD equal to 23.81% (i.e. small file - Low CD). During SME deserialisation procedure, the XML file is filtered, discarding not relevant tags and values. As a result, the size is reduced to $6.6KB$ and the CD increased to 43.48%, becoming thus a High CD XML file. In this case the deserialisation time of the original description takes 0.085s on average, with a 95% c.i. (calculated on 10 experiments) in the range $(0.067\text{s}, 0.102\text{s})$. These results witness the advantages of using SME on mobile devices for efficient elaboration of SWE XML files. In fact, SME is able to reduce SWE codification size while increasing their content density. In addition, processing time is proportional to the size of resulting SME codification, independently of the original file size.

In the same scenario, the processing overhead related to the management of an increasing number of observations exchanged between two nodes has been assessed. Specifically, both serialisation and deserialisation times of up to 15 sensor observations have been measured on one of the involved nodes. The experimental results are showed in Figure 3.15. In both cases, the last value represents the worst case in which the local node sequentially processes all the sent/received observations. In case of deserialisation, SME library requires at most 0.24s to com-

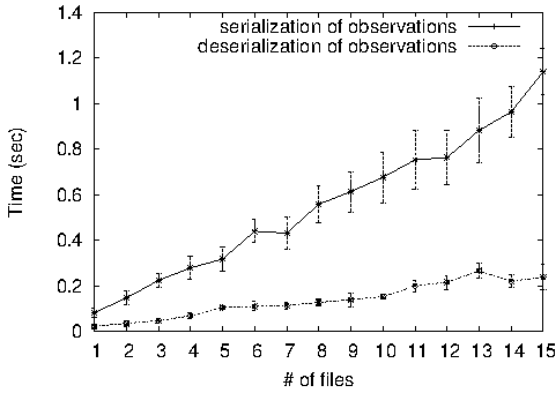


Figure 3.15: Scenario 1: XML processing time.

plete the operation. Instead, in case of serialisation it requires at most 1.14s. The difference between these processing times is mainly due to I/O operations of SME encoding. Specifically, read operations on mobile devices equipped with NAND memories are generally faster than write operations. In both cases, SME processing overhead does not heavily affect mobile device performance.

As far as the second scenario is concerned, Figure 3.16 shows the experimental results concerning the SME deserialisation procedure applied to an increasing number of sensor descriptions and observations generated by up to 10 remote nodes. Also in this case, SME library performance is really advantageous for mobile devices, requiring only a maximum of $1.782s$ and $3.662s$ to deserialise observations and descriptions respectively. The difference between these times is mainly due to the different size of single observations and descriptions of embedded sensors (on average, 1.9KB and 3.347KB respectively).

Results related to the first scenario showed that serialisation and deserialisation of a relatively large number of small XML files do not overload the mobile system processing capability, making SME library able to support mobile sensing services that require medium/high frequency of I/O operations on XML files. This is an important characteristic in opportunistic computing scenarios, in which limited contact times require efficient processing especially in case of real-time services.

In the third scenario, serialisation and deserialisation times of a real sensor observation derived from a SOS server have been measured. A single observation containing an increasing number of values related to independent measurements have been considered (from 0 up to $50,000$ with steps of $1,000$). The observation was originally encoded in SWE O&M standard and each value corresponded

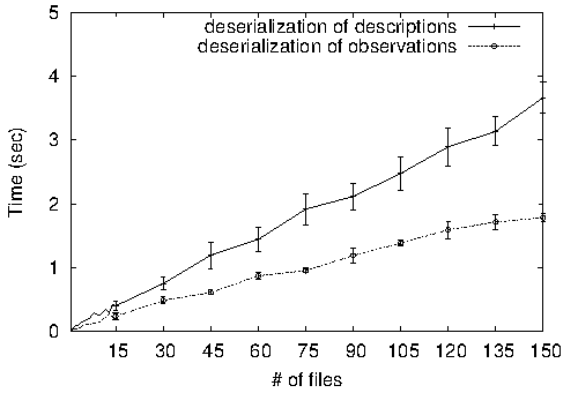


Figure 3.16: Scenario 2: deserialisation time.

to a sensing event represented by the following fields: a date, a number in double precision and a string describing the event. The XML file size of the considered observation ranges between $1.9\mathrm{KB}$ (i.e. containing only SWE XML overhead and 0 values) and $2.3\mathrm{MB}$ (i.e. containing $50,000$ values). The content density ranges between 18.4% and 99.9% respectively. Figure 3.17 highlights the efficiency of SME library in the management of complex SWE observations, corresponding to large XML files, in few seconds. Also in this case the difference between serialisation and deserialisation times is mainly due to I/O operations.

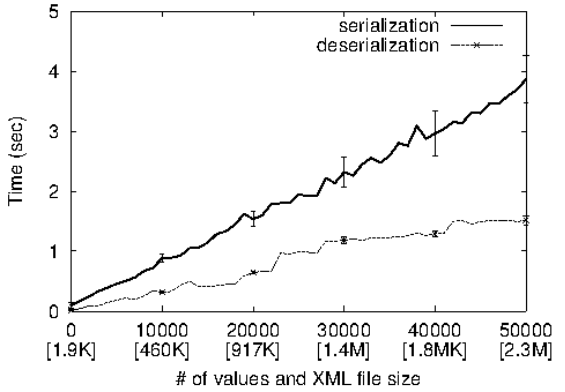


Figure 3.17: Scenario 3: XML processing time.

Battery Consumption

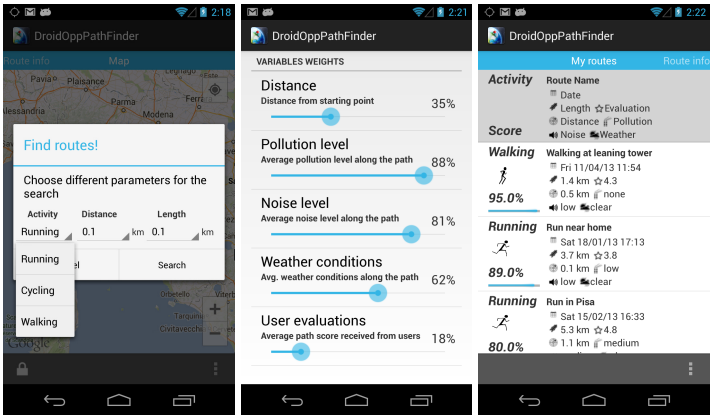
As a last set of experiments, the impact of XML processing on the battery consumption of mobile devices has been analysed. Firstly, the power consumption related to the continuous execution of serialisation and deserialisation procedures have been considered. At the end of a 15 hours test, the battery level moved from 100% to 98%. Considering the reference scenario involving both a group of CAMEO nodes and external SWE services, a second set of experiments has been run involving the use of CAMEO Wi-Fi communications. 5 tests have been performed, lasting one hour each, and the battery level reduced to 97% for all the 5 tests. To evaluate the impact of XML processing on battery consumption with respect to a power consuming service, the previous experiments have been performed including also the use of GPS. In this case the battery level reduced to 88% for all the 5 tests. These results clearly show that XML processing does not significantly impact on battery consumption, while the main consumption is related to the use of Wi-Fi, and GPS especially.

All the experimental results demonstrated the efficiency of SME introducing a limited overhead for XML processing on mobile devices.

3.5 DroidOppPathFinder: A Context and Social-Aware Path Recommender System Based on Opportunistic Sensing

This section presents DroidOppPathFinder, a MSN application based on CAMEO and designed to generate and share contents about paths for fitness activity [21]. The application is able to recommend the best path in a specific area by analysing user preferences and real-time environmental characteristics collected by heterogeneous sensing devices and services through opportunistic sensing mechanisms and exchanged using SME. To this aim, DroidOppPathFinder has been developed on top of CAMEO, which provides context- and social-aware functionality to improve both application performance and the user experience. This work represents a real example of opportunistic sensing service as additional support to the development of MSN applications. In addition, it demonstrates an efficient management of heterogeneous sensing data and services on mobile devices through the use of SME data format in order to further enrich the context of both local and remote nodes.

DroidOppPathFinder is enriched with opportunistic sensing functionality that involves users interested in outdoor physical activities (i.e. cycling, walking, hiking or running) by stimulating their social interactions and their active participation



(a) main (b) settings (c) list view

Figure 3.18: DroidOppPathFinder GUI

in generating and sharing useful information on possible paths. Paths are characterised by multiple context information derived by both embedded and external sensing capabilities to identify the environmental condition of the single path (e.g. real-time pollution, weather and noise conditions on the path) and by both objective and subjective evaluations provided by local and remote users (e.g. presence of holes or mud on the path, subjective impression of the difficulty, perceived temperature). DroidOppPathFinder exploits communication and sensing facilities provided by CAMEO and SME to sense the environment and to share the collected data amongst nodes in proximity. In this way, the application is able to collect context information characterising each path and locally rank the possible solutions based on the local user preferences.

Each mobile device running DroidOppPathFinder is able to collect real-time sensing data both from embedded sources (e.g. GPS and other local sensors) and from available external sources (e.g. air pollution sensor networks, weather stations). In addition, the application collects user generated contents regarding the paths as multimedia posts and comments and the personal user evaluation. The user can define her interests in paths related to a specific geographical area. When two nodes running DroidOppPathFinder meet each other, CAMEO automatically exchanges all the context information related to interesting paths (i.e. those related to the geographical area specified by the other node) through opportunistic communications. Each user can then require to the application a recommendation for a specific path or the best path in an area (with some additional information,

such as the distance from the original location, the path length) and the application is able to provide the information by requiring to CAMEO a specific context elaboration. In this way, DroidOppPathFinder exploits the interactions between users and devices through opportunistic communications to enrich the context of the local node and to improve path recommendation results. In addition, mobile nodes can cooperate to share the collected data and their resources in case the nodes have access to diverse types of information (e.g. some mobile nodes which are not able to connect to centralised weather or pollution web services can exploit the ability to directly interact with other nodes to obtain that information for a specific area).

DroidOppPathFinder provides the local user with additional sensing features based on noise level in the interested area through the analysis of data derived from the phone-embedded microphone. Therefore, each path experienced by the local user is further characterised by noise level. This information can be shared on the network together with the original characteristics of the path.

In this way, opportunistic communication allows users to collect real-time information about available paths. However, in order to extend the availability of collected information also to other users not in proximity, each node can decide to upload its piece of information to a SOS web server to collaborate in building sensing data of a specific area, depending also on the availability of an infrastructured connectivity.

Figure 3.18a presents some screenshots of the graphical user interface proposed in DroidOppPathFinder through which the user can request a path in a specific area and receive recommendations. As a starting point, the user can click on the map to select a point where to start searching paths. Then, a popup window requests some additional information to the user such as the desired type of fitness activity, the size of the area in which to search for the paths, and their maximum length. The following context information, either acquired by the local node or received from the network, is locally used to decide which path should be presented to the user: (i) the distance between the selected point and the path; (ii) the average pollution level on the path; (iii) the average noise level on the path; (iv) the average weather conditions on the path; (v) the average evaluation received from other users through comments and posts. The application specifies these parameters to CAMEO and a weighted sum of related values - to be normalised between 0 and 1 - is computed on each path based on the user requirements. In fact, the user can directly set the weights of the previous variables representing the personal relevance of each property of the path in the overall evaluation, as shown in Figure 3.18b. The results are then displayed as a list of entries (see Figure 3.18c)

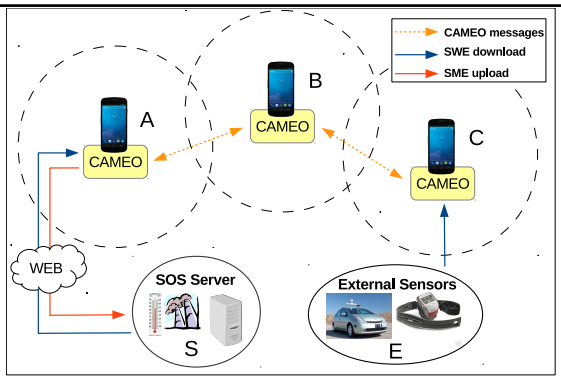


Figure 3.19: DroidOppPathFinder Scenario.

ordered by their overall evaluation. The application also maintains a history of the paths experienced by the local user and a cache of the paths collected through the network with an associated temporal validity.

For each path, map areas are highlighted with different colours and symbols associated with their average pollution, noise and weather conditions. The user can activate/deactivate layers to visualise the different categories of information on the map.

The social experience of the users is enriched by DroidOppPathFinder thanks to the multimedia feedback left to evaluate the paths. In fact, users can interact commenting the feedback and cooperating in building a collective awareness. This feature represents one of the most important characteristics enabled by CAMEO.

To better describe DroidOppPathFinder functionality, and opportunistic sensing and sharing features in particular, the reader can consider the use case scenario depicted in Figure 3.19. The scenario is composed of three mobile devices running DroidOppPathFinder (nodes A, B and C), a SOS web server (S) maintaining air pollution and weather context information, and some external sensing device (E). B has some paths stored in memory, but no information regarding pollution or noise. C has some noise data already obtained from an external sensor (E). A is in direct contact with S, from which it gets pollution and weather data. During previous encounters, B has collected some interesting paths. Then, when B encounters C and A they exchange their paths and context data and common paths are updated with the new information (e.g. noise data from C and pollution and weather data collected by A from the SOS server). Therefore, when the user of node B

asks for the best path available near her current location, DroidOppPathFinder recommends the best path considering also data provided by A and C.

3.5.1 Discussions

This section presented DroidOppPathFinder, a MSN application built on top of CAMEO and that exploits the availability of external context data derived from sensor data repositories accessed through the use of SME. The application is able to recommend the best path to follow to the user, during fitness activity. To do so, it combines context data coming from different sources and elaborated through CAMEO. The results indicate that CAMEO enables MSN applications with new features able to effectively enrich the social experience of the users, creating social relationships and exchanging content when needed. This helps to enlarge the collective awareness of the users.

4

Conclusions

With the increasing success of pervasive and ubiquitous communication systems enabled by the presence of technologically advanced mobile devices (e.g. smartphones and tablets of the last generation) and OSNs (e.g. Facebook and Twitter), new opportunities for Future Internet systems are arising. For example, wireless interfaces can be used to enable direct communications between devices in proximity. A novel paradigm based on this concept, called Mobile Social Networking, has been proposed. MSNs inherit the communication mechanisms provided by OSNs, but they exploit opportunistic networking to enable content exchange amongst nodes in proximity through direct communications (e.g. via Wi-Fi or Bluetooth). Currently, MSNs lacks a common platform to provide opportunistic networking functionality on mobile devices and to ease the collection and management of context and social data. These data are essential for the optimisation of MSNs, since they can provide information about the node running the application, the surrounding environment, and the state of other nodes in the network, that can be used to optimise content exchange and to personalise the social experience of the users.

Since MSNs are both people- and content-centric, the analysis of context and social data is essential for their design. In particular, the analysis of the structural properties of online personal social networks (i.e. ego networks) of the users could provide important insights on communication patterns, interests, habits, etc. regarding both the virtual and the physical worlds. Despite this, whilst a lot of work has been done to characterise ego networks in offline social networks, there is no detailed information regarding the structural properties of ego networks in OSNs.

To identify one of the theoretical bases for the design of new MSNs, this thesis provided a detailed analysis of the structural and dynamic properties of ego

Table 4.1: Comparison between the properties of offline and online ego networks.

		super support clique	support clique	sympathy group	affinity group	active network
Twitter	circle	C_1	C_2	C_3		C_4
	min freq.	17.28	6.00	1.77		0.20
	size	1.74	5.75	17.56		70.04
Facebook	circle	–	C_1	C_2	C_3	C_4
	min freq.	–	5.09	1.95	0.67	0.11
	size	–	(4.70)	(15.31)	(44.77)	(132.50)
Offline	min freq.	–	4.29	1.00	–	0.08
	size	–	4.6	14.3	42.6	132.5

networks in OSNs, assessing the differences between offline and online social networks. The results indicate that ego networks in the two worlds share the same structural properties, in terms of size and ego network circles composition. In particular, a set of ego networks have been extracted from Twitter and Facebook data sets, by calculating the frequency of contact between pairs of users. The same clustering technique (i.e. k-means fixing the number of clusters to 4) has been applied on the frequencies of contact of the ego networks. The resulting clusters have been combined to be comparable with the circles defined by the ego network model. Specifically, since ego network circles are inclusive (i.e. support clique $\subseteq$ sympathy group $\subseteq$ affinity group $\subseteq$ active network), the clusters obtained by using k-means (i.e. S_i with $i \in 1, 2, 3, 4$, ordered according to their frequency of contact, so that S_1 has the highest value and S_4 the lowest) that are disjoint by definition, have been merged into circles $C_k = \bigcup_{i=1}^{k} S_i$ so that $C_1 \subseteq C_2 \subseteq C_3 \subseteq C_4$. Doing so it has been possible to compare the ego network circles in online and offline environments in terms of size and typical frequency of contact, as reported in Table 4.1. In the table, the circles found in Facebook and Twitter are mapped to the circles in the ego network model according to their properties. The size of Facebook ego networks represents an estimate of the real size since, as detailed in Section 2.3, the analysed data is related to a random sample of the corresponding Facebook ego networks.

Most of the online ego network circles have a good match with the offline circles. This is a strong indication of similarity between offline and online ego networks and suggests that human social capacity (i.e. the maximum number of social contacts people can actively maintain in their networks), identified by the Dunbar's

number, and the structure of human ego networks are not influenced by the use of a particular social medium and they are instead controlled by the cognitive constraints of human brain. Nevertheless, Twitter ego networks present a new internal social circle, formed of one or two people on average, that was already hypothesised in sociology but its existence has never been proved yet due to the lack of sufficiently detailed data. This demonstrates the potential and the importance of ego network analysis in OSNs and contributes to the characterisation of human social behaviour in both online and offline environments.

The last circle in Twitter (C_4) appears to be something in between an affinity group and an active network. This particular result could be influenced by the choice of using Twitter replies to estimate the tie strength between ego and alters. Replies involve high intentionality since a user usually have to read the original tweet before replying to it. This requires additional cognitive resources compared to other types of messages (e.g. normal tweets) and could be too much to measure weak social relationships in the active network. Nevertheless, the size of this circle in Twitter is compatible with active networks found in other work on offline social networks [83].

Through a detailed analysis of the temporal evolution of ego networks in Twitter, it has been possible to demonstrate that, even though ego networks in OSNs show the same structures found in offline ego networks, they have a higher level of turnover compared to more traditional social networks. In fact, alters contacted by egos in Twitter are quickly abandoned to make room for new social contacts. This suggests that the key property to understand OSNs could be the ability of their users to constantly add new people in their networks, resulting in an improved access to a broader range of social resources.

Another contribution of this thesis is the description of a platform for the development of MSNs, called CAMEO, aimed at giving a set of APIs to application developers to access opportunistic networking functionality and common methods to collect and manage social and context information, shared by multiple MSNs at the same time. MSNs, thanks to CAMEO, further stimulate the social interactions between users based on the ability of their devices to communicate in proximity and exploiting interests and needs they have in common. Experimental results showed the effectiveness of CAMEO and MSNs. Real examples of MSNs have been presented to better understand the potential of CAMEO and its functionality. CAMEO has also been extended to support standards for the exchange and elaboration of sensor data trough external services, defined by the Sensor Web Enablement framework (SWE).

Moreover, to improve the collection and elaboration of SWE data, a new lightweight data format called Sensor Mobile Enablement (SME) has been defined and integrated in CAMEO. SME is compatible with SWE, but inherits only a small part of its properties, thus lowering the complexity of XML files used for encoding sensor data. The experimental results on SME showed the efficiency of CAMEO in the elaboration of context and social information encoded in the new format, compared to SWE data formats. A prototypical MSN application called DroidOppPathFinder has been presented to show how the social-awareness of the users can be improved by using MSNs and CAMEO. In fact, the application recommends to the user the best track to follow during fitness activity based on context and social data and on user's personal interests. The creation of new social interactions between users are stimulated by the application to improve user social experience and to allow to share useful contents regarding sport tracks when needed.

In conclusion, this thesis provided both theoretical and practical instruments for the design of new MSNs, creating a new framework for the Future Internet. In fact, whilst the analysis of the structural properties of ego networks in OSNs provides on of the bases for the design of future MSNs centred on human sociality, CAMEO represents a concrete support for the development of MSNs.

References

1. http://developer.android.com/guide/topics/fundamentals.html.
2. http://sax.sourceforge.net/.
3. http://simple.sourceforge.net/.
4. http://www.haggleproject.org.
5. http://www.json.org.
6. http://www.obj-sys.com/xbinder.shtml.
7. http://www.ogcnetwork.net/jaxb4ogc.
8. http://www.omg.org/spec/CPP/1.2.
9. http://www.omg.org/spec/I2JAV/1.3.
10. http://www.opengeospatial.org/projects/groups/sensorwebdwg.
11. http://www.opengeospatial.org/projects/groups/sweiotswg.
12. http://www.oracle.com/technetwork/articles/javase/index-140168.html.
13. http://www.w3.org/DOM/.
14. http://www.w3.org/XML/EXI/.
15. http://xmlbeans.apache.org/.
16. http://overstated.net/2009/03/09/maintained-relationships-on-facebook, 2009.
17. Tarek Abdelzaher, Yaw Anokwa, Peter Boda, Jeff Burke, Deborah Estrin, Leonidas Guibas, Aman Kansal, Samuel Madden, and Jim Reich. Mobiscopes for Human Spaces. *IEEE Pervasive Computing*, 6(2):20–29, 2007.
18. Valerio Arnaboldi, Marco Conti, and Franca Delmastro. Implementation of CAMEO: a Context-Aware MiddlEware for Opportunistic Mobile Social Networks. In *WoWMoM*, pages 1–3, 2011.
19. Valerio Arnaboldi, Marco Conti, and Franca Delmastro. CAMEO: a novel Context-Aware MiddlEware for Opportunistic Mobile Social Networks. Technical report, IIT-CNR, 2012.
20. Valerio Arnaboldi, Marco Conti, and Franca Delmastro. CAMEO: A novel context-aware middleware for opportunistic mobile social networks. *Pervasive and Mobile Computing*, October 2013.
21. Valerio Arnaboldi, Marco Conti, Franca Delmastro, Giovanni Minutiello, and Laura Ricci. DroidOppPathFinder: A Context and Social-Aware Path Recommender System Based on Opportunistic Sensing. In *IEEE International Symposium on a World of Wireless Mobile and Multimedia Networks*, pages 0–2, 2013.

22. Valerio Arnaboldi, Marco Conti, Franca Delmastro, Giovanni Minutiello, and Laura Ricci. Sensor Mobile Enablement (SME): a Light-Weight Standard for Opportunistic Sensing Services. In *International Workshop on the Impact of Human Mobility in Pervasive Systems and Applications*, 2013.
23. Valerio Arnaboldi, Marco Conti, Andrea Passarella, and Robin Dunbar. Dynamics of Personal Social Relationships in Online Social Networks: a Study on Twitter. In *COSN*, 2013.
24. Valerio Arnaboldi, Marco Conti, Andrea Passarella, and Fabio Pezzoni. Analysis of Ego Network Structure in Online Social Networks. In *SocialCom*, pages 31–40, 2012.
25. Valerio Arnaboldi, Marco Conti, Andrea Passarella, and Fabio Pezzoni. Analysis of Ego Network Structure in Online Social Networks. Technical report, 2012.
26. Valerio Arnaboldi, Marco Conti, Andrea Passarella, and Fabio Pezzoni. Ego Networks in Twitter: an Experimental Analysis. In *INFOCOM*, pages 3459–3464, 2013.
27. Valerio Arnaboldi, Andrea Guazzini, and Andrea Passarella. Egocentric Online Social Networks: Analysis of Key Features and Prediction of Tie Strength in Facebook. *Computer Communications*, 36(10-11):1130–1144, 2013.
28. Valerio Arnaboldi, Andrea Passarella, Maurizio Tesconi, and Davide Gazzè. Towards a Characterization of Egocentric Networks in Online Social Networks. In *OTM Workshops*, volume 7046, pages 524–533, 2011.
29. C Bettini, O Brdiczka, K Henricksen, J Indulska, D Nicklas, A Ranganathan, and D Riboni. A survey of context modelling and reasoning techniques. *Pervasive and Mobile Computing*, 6(2):161–180, 2010.
30. C Boldrini, M Conti, F Delmastro, and A Passarella. Context-and social-aware middleware for opportunistic networks. *Journal of Network and Computer Applications*, 33(5):525–541, 2010.
31. C Boldrini, M Conti, I Iacopini, and A Passarella. HiBOp: a History-based Routing Protocol for Opportunistic Networks. In *Proc. of IEEE WoWMoM 2007*, Helsinki, Finland, 2007.
32. C Boldrini, M Conti, and A Passarella. ContentPlace: social-aware data dissemination in opportunistic networks. In *ACM MSWiM*, pages 203–210, Vancouver, British Columbia, Canada, 2008.
33. C Boldrini, M Conti, and A Passarella. Context and resource awareness in opportunistic network data dissemination. In *IEEE AOC Workshop*, 2008.
34. C Boldrini, M Conti, and A Passarella. Exploiting users social relations to forward data in opportunistic networks: The HiBOp solution. *Pervasive and Mobile Computing*, 4(5):633–657, 2008.
35. Chiara Boldrini and Andrea Passarella. HCMM: Modelling spatial and temporal properties of human mobility driven by users' social relationships. *Computer Communications*, 33(9):1056–1074, 2010.
36. E Borgia, M Conti, and F Delmastro. MobileMAN: integration and experimentation of legacy mobile multihop ad hoc networks. *IEEE Communications Magazine*, 44(7):74, 2006.
37. A Borgida, M Lenzerini, and R Rosati. *The Description Logic Handbook: Theory, Implementation and Applications*. Cambridge Univ Pr, 2003.
38. Danah M. Boyd and Nicole B. Ellison. Social Network Sites: Definition, History, and Scholarship. *Journal of Computer-Mediated Communication*, 13(1):210–230, 2007.

39. Zi Chu, Steven Gianvecchio, Haining Wang, and S Jajodia. Who is tweeting on twitter: human, bot, or cyborg? In *Annual Computer Security Applications Conference*, 2010.
40. A L Comrey and H B Lee. *A first course in factor analysis*, volume 2. Lawrence Earlbaum Associates, 1992.
41. Marco Conti, SK Das, Chatschik Bisdikian, and Mohan Kumar. Looking ahead in pervasive computing: challenges and opportunities in the era of cyber-physical convergence. *Pervasive and Mobile Computing*, 8(1):2–21, 2011.
42. Marco Conti and M Kumar. Opportunities in opportunistic computing. *Computer*, 43(1):42–50, 2010.
43. C Cortes and V Vapnik. Support-vector networks. In *Machine learning*, pages 273–297, 1995.
44. Patricia Dockhorn Costa, Giancarlo Guizzardi, Joao Paulo A Almeida, Luis Ferreira Pires, and Marten van Sinderen. Situations in Conceptual Modeling of Context. In *EDOC Workshops*, page 6, 2006.
45. O. Curry and RI Dunbar. Why birds of a feather flock together: the effects of similarity on altruism. *Personal and Social Relationships*, 2011.
46. John Delaney, Nathan Salminen, and Eunice Lee. Time americans spend per month on social media sites - sociallyawareblog.com, 2012.
47. F Delmastro. From Pastry to CrossROAD: CROSS-layer Ring Overlay for AD hoc networks. In *PerCom Workshops*, pages 60–64. IEEE, 2005.
48. Peter Sheridan Dodds, Roby Muhamad, and Duncan J Watts. An experimental study of search in global social networks. *Science*, 301(5634):827–9, 2003.
49. R I M Dunbar. The social brain hypothesis. *Evolutionary Anthropology*, 6(5):178–190, 1998.
50. R I M Dunbar and S G B Roberts. Communication in Social Networks: Effects of Kinship, Network Size and Emotional Closeness. *Personal Relationships*, 2010.
51. R. I. M. Dunbar and M. Spoors. Social Networks, Support Cliques and Kinship. *Human Nature*, 6(3):273–290, 1995.
52. Martin Ester, HP Kriegel, and Jörg Sander. A density-based algorithm for discovering clusters in large spatial databases with noise. In *KDD*, pages 226–231, 1996.
53. Massimiliano La Gala, Valerio Arnaboldi, Andrea Passarella, and Marco Conti. Ego-net Digger: a New Way to Study Ego Networks in Online Social Networks Categories and Subject Descriptors. In *KDD Workshops*, pages 9–16, 2012.
54. Eric Gilbert and Karrie Karahalios. Predicting tie strength with social media. In *CHI*, pages 211–220, 2009.
55. Bruno Gonçalves, Nicola Perra, and Alessandro Vespignani. Modeling users' activity on twitter networks: validation of Dunbar's number. *PloS one*, 6(8):e22656, 2011.
56. Neil Zhenqiang Gong, Wenchang Xu, Ling Huang, Prateek Mittal, Emil Stefanov, Vyas Sekar, and Dawn Song. Evolution of Social-Attribute Networks: Measurements, Modeling, and Implications using Google+. In *IMC*, pages 131–144, 2012.
57. Mark S. Granovetter. The Strength of Weak Ties. *The American Journal of Sociology*, 78(6):1360–1380, 1973.
58. Keith N Hampton, Lauren Sessions Goulet, Lee Rainie, and Purcell Kristen. Social networking sites and our lives. Technical report, Pew Internet & American Life Project, 2011.
59. Karen Henricksen and Jadwiga Indulska. Developing context-aware pervasive computing applications: Models and approach. *Pervasive and mobile computing*, 2(1):37–64, 2006.

60. Karen Henricksen, Jadwiga Indulska, and Ted McFadden. Modelling context information with ORM. In *OTM Workshops*, pages 626–635, 2005.
61. R. A. Hill and R. I. M. Dunbar. Social network size in humans. *Human Nature*, 14(1):53–72, 2003.
62. Shawndra Hill, Foster Provost, and Chris Volinsky. Network-Based Marketing: Identifying Likely Adopters via Consumer Networks. *Statistical Science*, 21(2):256–276, 2006.
63. A Iamnitchi, J Blackburn, and N Kourtellis. The Social Hourglass: An Infrastructure for Socially Aware Applications and Services. *Internet Computing, IEEE*, 16(3):13–23, 2012.
64. Jason J. Jones, Jaime E. Settle, Robert M. Bond, Christopher J. Fariss, Cameron Marlow, and James H. Fowler. Inferring Tie Strength from Online Directed Behavior. *PLoS ONE*, 8(1):e52168, 2013.
65. M Kabir, J Han, J Yu, and A Colman. SCIMS: a social context information management system for socially-aware applications. In *CAiSE*, volume 7328, pages 301–317, 2012.
66. HF Kaiser. The varimax criterion for analytic rotation in factor analysis. *Psychometrika*, 23(3):187–200, 1958.
67. R Kanai, B Bahrami, R Roylance, and G Rees. Online social network size is reflected in human brain structure. *Biological sciences / The Royal Society*, 279(1732):1327–34, 2012.
68. DJ Ketchen and CL Shook. The application of cluster analysis in strategic management research: an analysis and critique. *Strategic management journal*, 17(6):441–458, 1996.
69. N Kourtellis, J Finnis, P Anderson, J Blackburn, C Borcea, and A Iamnitchi. Prometheus: User-controlled P2P social data management for socially-aware applications. *Middleware 2010*, pages 212–231, 2010.
70. Hans-Peter Kriegel, Peer Kröger, Jörg Sander, and Arthur Zimek. Density-based clustering. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 1(3):231–240, 2011.
71. Jure Leskovec and Eric Horvitz. Planetary-Scale Views on an Instant-Messaging Network. Technical report, 2007.
72. Peter V Marsden and Karen E Campbell. Measuring Tie Strength. *Social Forces*, 63(2):482–501, 1984.
73. Giovanna Miritello, Rubén Lara, Manuel Cebrian, and Esteban Moro. Limited communication capacity unveils strategies for human interaction. *Scientific Reports*, 3:1–7, June 2013.
74. Alan Mislove, Massimiliano Marcon, Krishna P Gummadi, Peter Druschel, and Bobby Bhattacharjee. Measurement and analysis of online social networks. In *IMC*, volume 40, page 29, 2007.
75. Michael Muller, David R Millen, N Sadat Shami, and Jonathan Feinberg. We are all Lurkers: Toward a Lurker Research Agenda. In *CSCW*, pages 1–10, 2010.
76. MEJ Newman and Juyong Park. Why social networks are different from other types of networks. *Physical Review E*, 68(3), 2003.
77. J.P. Onnela, J. Saramäki, J. Hyvönen, G. Szabó, D. Lazer, K. Kaski, J. Kertész, and A.L. Barabási. Structure and tie strengths in mobile communication networks. *PNAS*, 104(18):7332–7336, 2007.

78. Andrea Passarella and Marco Conti. Characterising aggregate inter-contact times in heterogeneous opportunistic networks. In *IFIP Networking*, pages 1–12, 2011.
79. Andrea Passarella, Marco Conti, Robin IM Dunbar, and Chiara Boldrini. Modelling Inter-contact Times in Social Pervasive Networks. In *ACM WSWiM*, 2011.
80. Luciana Pelusi, Andrea Passarella, and Marco Conti. Opportunistic networking: data forwarding in disconnected mobile ad hoc networks. *Communications Magazine, IEEE*, 44(11):134–141, 2006.
81. George Percivall, Carl Reed, and John Davidson. Open Geospatial Consortium Inc. Sensor Web Enablement: Overview And High Level Architecture, 2007.
82. Sam G.B. Roberts. Constraints on Social Networks. In *Social Brain, Distributed Mind (Proceedings of the British Academy)*, pages 115–134. 2010.
83. Sam G.B. Roberts, Robin I.M. Dunbar, Thomas V. Pollet, and Toon Kuppens. Exploring variation in active network size: Constraints and ego characteristics. *Social Networks*, 31(2):138–146, 2009.
84. T P Ryan. *Modern Regression Methods*, volume 39. Wiley, 1997.
85. J Saramaki, EA Leicht, and E Lopez. The persistence of social signatures in human communication. *arXiv preprint arXiv*, 2012.
86. C. P. Van Shaik. Why are diurnal primates living in groups? *Behaviour*, 87(1/2):120–144, 1983.
87. Herbert Simon. A Behavioral Model of Rational Choice. *The Quarterly Journal of Economics*, 69(1):99–118, 1955.
88. Alistair Sutcliffe, Robin Dunbar, Jens Binder, and Holly Arrow. Relationships and the social brain: Integrating psychological and evolutionary perspectives. *British journal of psychology*, 103(2):149–68, 2012.
89. Mark Sweney. Facebook sees first dip in UK users - guardian.co.uk, 2008.
90. Alain Tamayo, Carlos Granell, and J Huerta. Analysing Performance of XML Data Binding Solutions for SOS Applications. In *Proceedings of Workshop on Sensor Web Enablement*, 2011.
91. Alain Tamayo, Carlos Granell, and J Huerta. Dealing with large schema sets in mobile SOS-based applications. *arXiv preprint arXiv:1110.0209*, 2011.
92. Alain Tamayo, Carlos Granell, and Joaquín Huerta. Using SWE Standards for Ubiquitous Environmental Sensing: A Performance Analysis. *Sensors*, 12(9):12026–12051, August 2012.
93. Alessandra Toninelli, Animesh Pathak, and V Issarny. Yarta: A middleware for managing mobile social ecosystems. *Advances in Grid and Pervasive Computing*, pages 209–220, 2011.
94. Alessandra Toninelli, Animesh Pathak, Amir Seyedi, Roberto Speicys Cardoso, and Valerie Issarny. Middleware Support for Mobile Social Ecosystems. In *CSA Workshops*, pages 293–298. Ieee, July 2010.
95. Jeffrey Travers and Stanley Milgram. An Experimental Study of the Small World Problem. *Sociometry*, 32(4):425, 1969.
96. Johan Ugander, Brian Karrer, Lars Backstrom, and Cameron Marlow. The Anatomy of the Facebook Social Graph. *CoRR*, 2011.
97. Bimal Viswanath, Alan Mislove, Meeyoung Cha, and Krishna P. Gummadi. On the evolution of user interaction in Facebook. In *WOSN*, page 37, 2009.
98. Haizhou Wang and Mingzhou Song. Clustering in One Dimension by Dynamic Programming. *The R Journal*, 3(2):29–33, 2011.

99. Greg White, Jaakko Kangasharju, Don Brutzman, and Stephen Williams. Efficient XML Interchange Measurements Note, 2007.
100. Christo Wilson, Bryce Boe, Alessandra Sala, Krishna P N Puttaswamy, and Ben Y Zhao. User interactions in social networks and their implications. In *EuroSys*, pages 205–218, 2009.
101. Christo Wilson, Alessandra Sala, Krishna P. N. Puttaswamy, and Ben Y. Zhao. Beyond Social Graphs: User interactions in online social networks and their implications. *ACM Transactions on the Web*, 6(4):1–31, November 2012.
102. Ben Worthen. Bill Gates quits Facebook - Wall St. Journal Online, 2008.
103. S S Yau and J Liu. Hierarchical situation modeling and reasoning for pervasive computing. In *SEUS/WCCIA Workshops*, pages 6 pp.–, 2006.
104. X Zhao, Alessandra Sala, Christo Wilson, and Xiao Wang. Multi-scale dynamics in a massive online social network. In *IMC*, pages 171–184, 2012.
105. W-X Zhou, D Sornette, R a Hill, and R I M Dunbar. Discrete hierarchical organization of social group sizes. In *Biological sciences*, volume 272, pages 439–44, 2005.

Acknowledgements

Foremost, I would like to thank my advisors Enzo Mingozzi, Marco Conti and Andrea Passarella for their endless support and their valuable advice that helped me a lot during the Ph.D.

Besides my advisor, I would also like to express my most sincere gratitude to Elena Pagani, who has always supported me during this adventure, being always ready to lend a helping hand.

I am also grateful to Franca Delmastro. I learned a lot from her and this thesis contains most of the work we have done together.

I would warmly thank Professor Robin Ian MacDonald Dunbar, who gave me the opportunity to make a great experience as a visiting student in his Research team. Working with him helped me to diversify my knowledge and to develop critical thinking.

I would like to thank also my fellows Andrea Guazzini, Fabio Pezzoni, Massimiliano La Gala, Giovanni Minutiello, and all the other people at IIT-CNR with whom I shared many good moments during these three years.

A special thank goes to Paolo Santerini, who helped me printing this thesis and who always has good jokes to tell.

Last but not least, I would like to thank my family, my girlfriend, and my closest friends for their support.