Universidade de Aveiro Departamento de Electrónica, Telecomunicações e Informática
2015

Christopher Viana Lima

Aplicações Colaborativas Sensíveis ao Contexto

Context-Aware Framework for Collaborative Applications

Universidade de Aveiro Departamento de Electrónica, Telecomunicações e Informática
2015

Christopher Viana Lima

Aplicações Colaborativas Sensíveis ao Contexto

Context-Aware Framework for Collaborative Applications

Tese apresentada à Universidade de Aveiro para cumprimento dos requisitos necessários à obtenção do grau de Doutor em Engenharia Informática, realizada sob a orientação científica do Doutor Diogo Gomes, Professor Auxiliar do Departamento de Eletrónica, Telecomunicações e Informática da Universidade de Aveiro e sob a coorientação científica do Doutor Rui L. Aguiar, Professor Catedrático do Departamento de Eletrónica, Telecomunicações e Informática da Universidade de Aveiro.

Apoio financeiro da FCT e do FSE no âmbito do III Quadro Comunitário de Apoio.

Dedico aos meus pais e a meu filho.

o júri

presidente

Doutor Artur Manuel Soares da Silva
Professor Catedrático da Universidade de Aveiro

Doutor José Benjamim Ribeiro da Fonseca
Professor Auxiliar com Agregação da Escola de Ciência e Tecnologia da Universidade de Trás-os-Montes e Alto Douro

Doutora Helena Cristina Coutinho Duarte Rodrigues
Professora Auxiliar da Universidade do Minho

Doutor Joel José Puga Coelho Rodrigues
Professor Auxiliar da Universidade da Beira Interior

Doutor Diogo Nuno Pereira Gomes
Professor Auxiliar da Universidade de Aveiro (orientador)

Doutor João Paulo Silva Barraca
Professor Assistente Convidado da Universidade de Aveiro

agradecimentos

O autor gostaria de agradecer aos colegas de projeto SOCIETIES e PERSIST, sem os quais a investigação não teria sido possível.

Gostaria de agradecer ao grupo CESAM da Universidade de Aveiro por cooperar na participação nos testes.

Este trabalho não seria possível sem as contribuições de Mário Antunes, Andreia Rodrigues, Telma Mota, Sancho Rego, João Goncalves, Nuno Pássaro, Nuno Carapeto, André Rainho, Carla Leite, Nikos Kalatzis, Nicolas Liampotis, John McGovern, Babak Farshchian e Paulo Chainho.

Agradeço especialmente aos meus orientadores Prof. Diogo Gomes e Prof. Rui Aguiar e a todos do ATNoG (Advanced Telecommunications and NetwOrks Group) e ao IT (Instituto de Telecomunicações).

Esta pesquisa foi apoiada pela FCT - Fundação para a Ciência e a Tecnologia (SFRH/BD/69381/2010).

palavras-chave

Computação Pervasiva, Sistemas Colaborativos, Informação de Contexto, Computação Social.

resumo

O futuro dos ambientes pervasivos irá levar em consideração não só os interesses individuais dos utilizadores, mas também as relações sociais. Desta forma, a participação em comunidades pode levar o utilizador para além da experiência em espaços tradicionais pervasivos, permitindo a cooperação entre grupos e tendo em conta não só os interesses individuais, mas também o contexto coletivo e social. Aplicações sociais na área de CSCW (Computer Supported Cooperative Work) representam novos desafios e possibilidades em termos do uso da informação social de contexto para a adaptação e personalização em computação pervasiva. Esta tese explora o potencial da utilização consciente do contexto e das informações sociais em aplicações CSCW, a fim de apoiar a colaboração em ambientes pervasivos. Em particular, a investigação descreve a abordagem do design e desenvolvimento de um framework consciente do contexto para aplicações colaborativas (CAFCA), utilizando-se das informações de contexto e sociais dos utilizadores para adaptações proactivas em ambientes pervasivos.

A fim de validar o framework proposto, uma avaliação foi realizada com um grupo de utilizadores, com base num cenário empresarial . A análise permitiu verificar o impacto do framework em termos de funcionalidade e eficiência em condições do mundo real.

A principal contribuição deste trabalho foi desenvolver um framework context-aware para suportar aplicações de colaboração em ambientes pervasivos. A investigação centrou-se em uma abordagem sociotécnica inovadora de explorar a colaboração em comunidades. Os principais objetivos residem na capacidade sociais para a formação de sessões, a comunicação e coordenação de groupware para atividades colaborativas.

keywords Pervasive Computing, CSCW, Context Aware, Social Computing.

abstract

Future pervasive environments will take into consideration not only individual users' interest, but also social relationships. In this way, pervasive communities can lead the user to participate beyond traditional pervasive spaces, enabling the cooperation among groups and taking into account not only individual interests, but also the collective and social context. Social applications in CSCW (Computer Supported Cooperative Work) field represent new challenges and possibilities in terms of use of social context information for adaptability in pervasive computing. This thesis explores the potential of using context-aware and social information in CSCW application in order to support collaboration in pervasive environments. In particular, the research describes the approach in the design and development of a context-aware framework for collaborative applications (CAFCA), utilizing users' context and social information for proactive adaptations in pervasive environments.

In order to validate the proposed framework an evaluation was conducted with a group of users based on enterprise scenario. The analysis enabled to verify the impact of the framework in terms of functionality and efficiency in real-world conditions.

The main contribution of this thesis was to provide a context-aware framework to support collaborative applications in pervasive environments. The research focused on providing an innovative socio-technical approach to exploit collaboration in pervasive communities. Finally, the main results reside in social matching capabilities for session formation, communication and coordination of groupware for collaborative activities.

Contents

List of Figures

List of Tables

Acronyms

AmI	Ambient Intelligence
API	Application Programming Interface
CAA	Context-Aware Application
CAFCA	Context-Aware Framework for Collaborative Application
CAS	Complex Adaptive Systems
CB	Context Broker
CIS	Community Interaction Space
CMC	Computer-mediated Communication
CMS	Content Management System
CoI	Communities of Interests
CoP	Communities of Practice
CSS	Cooperating Smart Space
CxC	Context Consumer
CxP	Context Provider
CORBA	Common Object Request Broker Architecture
CSCW	Computer Supported Cooperative Work
CSCL	Computer Supported Collaborative Learning
DSN	Dynamic Social Networks
DSNA	Dynamic Social Networks Analysis
DSNS	Decentralised Social Networking Systems
IM	Instant Message
IRC	Internet Relay Chat
MAS	Multi-Agent System
MoSoSo	Mobile Social Software
MSN	Mobile Social Networking
OS	Operational System
OSGi	Open Services Gateway Initiative
OWL	Web Ontology Language
P2P	Peer-to-Peer
PDA	Personal Digital Assistant
RMI	Remote Method Invocation
SGACS	Small Groups as Complex Systems

SIP	Session Initiation Protocol
SNA	Social Network Analysis
SNS	Social Network Service
SOA	Service Oriented Architecture
W3C	World Wide Web Consortium
WebRTC	Web Real-Time Communication
Wi-Fi	Wireless Fidelity
XML	eXtensible Markup Language
XMPP	Extensible Messaging and Presence Protocol

1 INTRODUCTION

With the growing adoption of mobile and pervasive devices, there is an increasing demand of applications with self-adaptive behaviours using social and environment information. These characteristics can provide better end-user experience, not only in home and entertainment places, but also in collaborative work situations.

To date, Ambient Intelligence (AmI) has achieved a matured level, thanks to the advance of the technology, mainly by sensors and mobile devices carried by users (Wang et al., 2004). Typically, context information in pervasive environments can be retrieved in a variety of ways such as physical sensors, network information, user profiles and other sources. Context information refers to a set of conditions (e.g. environment features, user activities, location) that may be utilized by applications in order to adapt dynamically their behaviour. Environments such as smart spaces are aware of context such as location/proximity (e.g. home, office) and environment features (e.g. hot, cold) enabling users to access computing resources to perform their activities. By combining smart spaces with social computing, it is possible to create a pervasive environment in which social information can be used as context information. In addition, the increasing usage of portable devices can be exploited to create rich applications involving adaptability without explicit user intervention, taking into account environmental context.

Activities in smart spaces may involve multiple users collaborating on shared tasks. Computer Supported Cooperative Work (CSCW) studies the cooperative work among individuals and addresses various applications for collaboration, ranging from conference to workflow environments. The CSCW area is not new, but only recently, research in context-aware CSCW applications reached pervasive systems. Additionally, the trend is the use of collective intelligence, where interpretation of context information related to social relationships should be harnessed as input for such collaborative applications. In this manner, combining pervasive systems and groups with collaborative context-aware applications can increase the activity's quality for the end user in a wide variety of tasks.

1.1 Motivation

With the advent of Web 2.0, various social networking sites have emerged in the present scenario of the Internet like Facebook[1], Google Plus[2] and LinkedIn[3]. The information provided by these social sites can be useful for pervasive or ubiquitous systems, where the change of environment and context are important to offer powerful ways of working and communicating.

Community support is gaining more importance, especially in areas where inter-human communication is a critical success factor (Koch et al., 2014). Communities in this case refer to a social unit, both to physical and virtual, that share common objectives. According to Grundin and Poltrock (2012), when people collaborate, they do so in the context of small groups or teams, organizations, or large-scale communities. However, there are differences among these social structures to achieve common goals. For instance, while small groups rely in working together in real time, communities have higher need for coordination. In this way, the concepts of small groups and communities of practice facilitate opportunistic collaborations taking advantage of social awareness.

Until recently, social network analysis of complex and unknown relationships were hard to achieve and only performed off-line (Chung et al., 2006). This means that data collected over a period could only be analysed after its collection, especially in scenarios where nodes may appear and disappear and edges may tie or untie. Nevertheless, today's technologies make it possible to build a solution with ability to adapt to constantly changing physical and social contexts, providing users with a unique experience for collaborations. Social context refers to the information relevant to the characterization of a situation that influences the interactions of users.

Relying on social networks analysis with the concepts of social capital, small groups and communities of practice it is potential to determine temporary social groups, based on user context information, physical location, and co-presence. This way, CSCW applications, combined with social computing and context-awareness, can bring great benefits for smart spaces participants' experience. As such, pervasive and social computing are central points to the development of systems for the future collaborative applications. All these aspects present new challenges for solutions in context-aware CSCW applications.

[1] https://www.facebook.com/

[2] https://plus.google.com/

[3] https://www.linkedin.com

1.2 Contribution

The goal of this PhD thesis is to demonstrate that CSCW applications can benefit from using context information merged with social computing in pervasive environments. In order to provide better and efficient collaboration among users, a Context-Aware Framework for Collaborative Applications (CAFCA) was designed and implemented. This is approached within pervasive communities, where each community shares information that can be combined in context-aware applications. This involves the use of context information and social context to provide context-awareness for collaborative applications, enhancing the experience of the participants in cooperative tasks.

The author participated in European projects PERSIST (Personal Self Improving Smart Spaces) and SOCIETIES (Self-Orchestrating CommunIty ambiEnT Intelligence Spaces), which enabled him the possibility to be involved with pervasive computing and context-awareness. The code developed under the Thesis was released to the public under Open Source License for public usage.

The contributions of this thesis are:

- Design and implementation of a context-aware framework for collaborative work aiming pervasive environments, enabling dynamic use of context and social information facilitating group formations (see Chapter 5). Published in Lima et al. (2014);
- Provide pro-active behaviour for communication, coordination and relationships in pervasive community scenarios for CSCW (see Chapter 5). Published in Lima et al. (2012).
- Verify the synergy among the intersecting research areas of pervasive computing, CSCW and social computing. Published in Lima et al. (2012).
- Provide a social matching mechanism capable to find participants based on the context information provided from the environment and user social context (see Chapter 6). Published in Lima et al. (2013a);
- Take advantage of context information available of social network sites such as Facebook, LinkedIn and real-world sensing sources to aggregate the context information to the user (see Chapter 6);
- Support the increment of existing context information by Natural Language Processing and semantic analyse (see subsection 5.5.1). Published in Lima et al. (2013b);
- Provide real-time adaptation of collaborative tools for collocated and non-collocated teams in order to facilitate users' interactions in response to context information changes (see subsection 5.6.3);
- Facilitate the integration of existing collaboration tools to offer context awareness, adaptability and coordination, aiming to reduce human interactions (see subsection 5.6.2). Published in Gomes et al. (2011);

- Support the election of appropriate application for communication based on the communication channels available from the users (see Chapter 5);
- Evaluate and test the proposed implementation in user trial for proof of concept and feedback (see Chapter 6);

1.3 Publications

1.3.1 Journal Publications

- Christopher V. Lima, Mário Antunes, Diogo Gomes, Rui L. Aguiar, Telma Mota, "A Context-Aware Framework for Collaborative Activities in Pervasive Communities", International Journal of Distributed Systems and Technologies, vol. 5, no. 2, pp. 31–43, IGI Global Publishing, July 2014.
- Roberta Lima-Gomes, Guillermo de Jesus Rivera, Roberto Willrich, Christopher V. Lima, Jean-Pierre Courtiat, "A Loosely Coupled Integration Environment for Collaborative Applications", IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans, vol. 41, no. 5, pp. 905–916, September 2011.

1.3.2 Conference and Workshop Publications

- Christopher V. Lima, Mário Antunes, Diogo Gomes, Rui L. Aguiar, "A Context-Aware Framework For CSCW Applications In Enterprise Environments" in Proceedings of the IADIS International Conference Web Based Communities and Social Media 2013, Proceedings of the IADIS International Conference Collaborative Technologies 2013, Czech Republic, pp. 11–18, July 2013,
- Christopher V. Lima, Mário Antunes, Diogo Gomes, Rui L. Aguiar, "Social Awareness in Pervasive Communities for Collaborative Work" In Intelligent Environments Workshops, volume 17 of Ambient Intelligence and Smart Environments, Greece, pp. 110 – 115, July 2013.
- Christopher V. Lima, Diogo Gomes, Rui L. Aguiar, "Pervasive CSCW for Smart Spaces Communities", 9th IEEE International Workshop on Managing Ubiquitous Communications and Services, pp. 118–123, Switzerland, February 2012.

1.4 Thesis Organization

The thesis is organized in two parts. The first part reviews essential domains to the work. Chapter 2 starts with an overview of the pervasive computing, introducing context-aware and pervasive communities. Chapter 3 identifies opportunities and challenges for CSCW. Chapter 4 focus on social computing describing social capital, communities of practices, Web 2.0 and social network analysis. This chapter also present the techniques used to provide support to CAFCA framework, including general cohort analysis, data analysis applied to social networks.

The second part presents the main thesis contribution with the design, implementation and evaluation of the proposed solution. Chapter 5 introduces the CAFCA framework, presenting the design concept, implementation and technical aspects of the proposal solution. In Chapter 6, it is discussed a case study conducted with the proof-of-concept implementation to validate the framework and challenges in the evaluation.

Finally, Chapter 7 presents general conclusions and implications of this thesis. The chapter also discuss further work directions and extensions.

In the Appendix A and B are presented the questionnaires used in the validation test in Chapter 6. Appendix C includes UML diagrams of CAFCA framework, as support to the work done in Chapter 5.

2 PERVASIVE COMPUTING

This chapter presents the areas of pervasive and context-aware computing from the point of view of the thesis. The chapter also introduces solutions that underlie a majority of pervasive computing application development. A set of future scenarios is presented covering different situations in enterprise domain to enable a better comprehension. These scenarios explain with practical examples the features and capabilities identified as requirements of the thesis.

2.1 Introduction and Motivation

Pervasive environments have evolved over the years, thanks to technology advances and decreasing prices in sensors and mobile devices (Want, 2010) . Moreover, the increase of consumer portable devices such as smartphone, tablets and smart watches are being also exploited in ambient intelligence (AmI) to create rich applications without explicit user intervention. AmI is used extensively in applications such as healthcare, home care, transportation and environmental monitoring (Al-Jaroodi et al., 2009). Environments in pervasive computing are aware of context, where location/proximity (e.g. home, office), activities (e.g. walking, driving) and environment features (e.g. hot, cold) provide sources of information to produce the applications. Mark Weiser's (Weiser, 1991) definition for pervasive and ubiquitous computing is that "The most profound technologies are those that disappear. They weave themselves into the fabric of everyday life until they are indistinguishable from it".

A fundamental challenge in pervasive computing has been to create systems that scale-up smart space infrastructures to support many users. In Genco and Sorce (2010), the authors describe pervasive systems to free users in which devices, networks and applications will all be seamlessly integrated and will cooperate in support of a worldwide-shared computing paradigm. Researchers have addressed both social and human-computer interaction aspects in the field, as well as technical aspects such as communication, interoperability, architectures and hardware design.

In pervasive computing, many projects address infrastructures for smart spaces and mobile users (Taylor et al., 2011). A popular definition for smart spaces is provided by Prehofer et al. (2007) as "A smart space is a multi-user, multi-device, dynamic interaction environment that enhances a physical space by virtual services". The goal of any smart environment is to release users from the tasks they usually perform to change their environment to suit their preferences and to access the available services. This objective is achieved by making the environment able to adapt itself to user needs, providing customized interfaces to the services available at each moment.

2.2 Context-Awareness

In Chen & Kotz (2000), context-aware or context information is defines as the paradigm of pervasive computing that allows applications to discover and take advantage of contextual information. Context may vary depending on application functionality, user goals and deployment environment. Schilit et al. (1994) were one the first to discuss context-aware computing, defining three important aspects: where you are, who you are with and what resources are nearby. Most cited definition for context is given by Abowd et al. (1999): "Any information that can be used to characterize the situation of an entity. An entity is a person, place, or object that is considered relevant to the interaction between a user and an application, including the user and applications themselves." Consequently, having a common definition of context is a challenging issue. Furthermore, sociological approaches typically regard context as networks of interacting entities (people, agents and artefacts) created and continually updated by the interactions among them (Zacarias et al., 2010).

A popular way of classification for context instances is given in literature (Gustavsen, 2002; Prekop and Burnett, 2003). These proposals divided context in internal and external. The internal dimension is specified by the user or captured by monitoring user interactions such as user's goals, tasks and emotional state. While the external dimension denotes the physical context, in which includes, for instance, location, touch, temperature and sound.

Figure 2.1 identifies invariant characteristics of the context for development of context-aware systems. These characteristics are (a) context relates always to some entity, (b) is used to solve a problem (c) depends on the domain and (d) is a dynamic process (Mena et al., 2007; Sandkuhl and Borchardt, 2014).

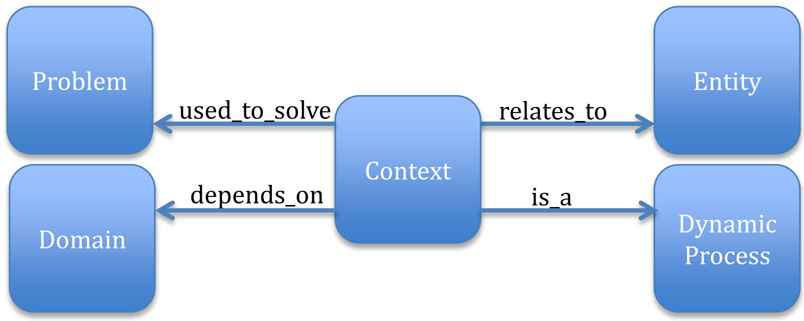

Figure 2.1: The invariant characteristics of context. Adapted from (Sandkuhl and Borchardt (2014)**.**

In order to support these characteristics, it is common the use of an integrated solution for management of context information. Next section presents in details context management systems.

2.3 Context Management Systems

A Context Management System (CMS) provides the means to receive context information and apply the information to build context-aware applications. Most of CMS rely on a centralized approach similar to Context Broker Architecture (CoBrA) (Chen et al., 2003). Context-aware systems can be designed taking into account different criteria (architecture, resource discovery, sensing, context model, context processing historical context data, security and privacy) and depend on requirements and conditions such as the location of sensors, amount of possible users and available resources (mobile phones or desktop PCs) (Baldauf et al., 2007).

A CMS is responsible for acquiring raw information from context sources and hardware sensors, modelling the collected data and maintaining the current and historic context in appropriate data repositories. It is important to stress that some CMS can slightly change or even be optional like the history management, context data aggregation and context reasoning. The components usually found in a CMS can be classified in four primary areas:

- Context Acquisition: A prerequisite for context-aware applications, it refers to a process of collecting contexts from various context sources. Contexts can be captured through physical devices (e.g. sensors, accelerometers and agents) and virtual sources (e.g., software applications and services).
- Context Storage: This component is responsible for the management of context information (i.e. retrieval, update, addition, removal).
- Context Modelling: It is in charge to construct high-level abstraction of contextual data and building relationships among them. Some models support querying not only current data, but also historical contextual information (Context History).

- Context Reasoning: It is the process of deriving high-level contexts from raw context data and deriving implicit contexts from explicit contexts information.

Projects like Mobilife (Kernchen et al., 2006) and C-CAST (Knappmeyer et al., 2009) use a broker-based approach for CMS. In these cases, the CMS was responsible for storing and retrieving context information in a distributed manner. Both projects use a Context Broker (CB), a fundamental entity that creates and manages the relationship between Context Provider (CxP) and Context Consumer (CxC) (Chen, 2004). The CB is responsible for acquire context information on behalf of resource-limited devices, reason about context information, detect and resolve inconsistent context information, and even protect user privacy by enforcing policies that the users have defined (Zafar et al., 2009). The CxP is in charge of detecting and acquiring information from context sources (e.g. temperature, position), while the CxC is where the context is used (or consumed).

In a centralized architecture, a central point handles all the requests present between the producers and consumers. In a distributed architecture, a point-to-point model is used, where each pair can be either a producer or a consumer. In some cases, the pair may play both roles (Chihani et al., 2011). Figure 2.2 illustrate a (a) centralized and a (b) distributed architecture for CMS.

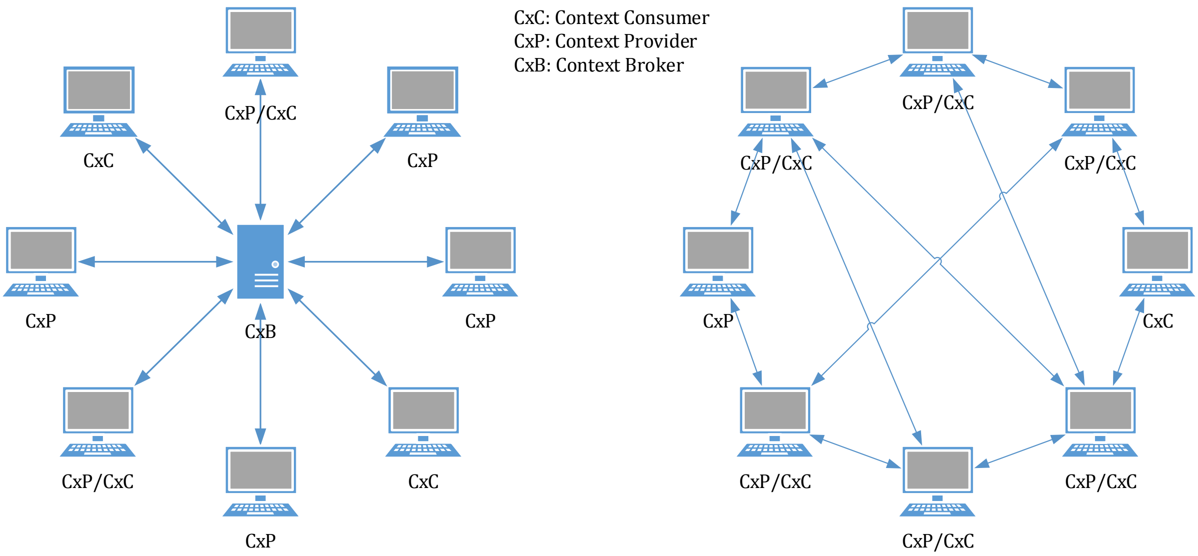

Figure 2.2: (a) Centralized and (b) distributed architecture for CMS (Chihani, 2013)**.**

Other CMS solutions also found in literature are: Gaia using Common Object Request Broker Architecture (CORBA) as a distributed middleware for communication; and Service-oriented Context Aware Middleware (SOCAM) architecture using OSGi with context events represented in Web Ontology Language (OWL) based on a predefined ontology (Tao Gu et al., 2004).

2.4 Smart Spaces and Mobility

Smart spaces play an important role in pervasive computing, providing an environment where users can interact electronically with other users and devices in the environment. However, these systems did not take into account the mobility of the user as shown in projects as Ubisec (Groppe and Mueller, 2005), MavHome (Youngblood et al., 2005), or as in smart homes scenarios (De Silva et al., 2013). More recently, projects like Daidalos (Taylor et al., 2011) and PERSIST (Venezia et al., 2009) approached this challenge proposing the use of smart spaces specially associated with personal portable devices, enabling the availability of content anytime and anywhere.

Today with mobile devices containing numerous hardware and software capabilities, it is possible to create new smart spaces systems. This way, personal smart spaces try to tackle static smart spaces, without the need of infrastructure investment, and mobility to provide pervasive functionalities to the user all the time. Personal smart space systems generally include a context management system, discovery mechanisms, event notification, personal preferences, and cooperation between the entities. Personal smart spaces are defined as “a set of services that are running or available within a dynamic space of connectable devices where the set of services and devices are owned, controlled, or administered by a single user or organisation” (Gallacher et al. 2011).

The implementation of smart spaces requires software and hardware components to establish communication among users and resources, collecting information about users’ context, discovery, services, learning (users’ intentions), overall collecting data to plan and execute actions on the users’ behalf. Most current proposals for physical and logical mobility rely on the technique of middleware, which resides between the operating system and the application layer. Middleware is a class of software design to support application development by enhancing the level of abstractions associated with the programming effort. In general, middleware provides basic services for upper layer applications by providing programming interfaces and taking charge of management functions such as caching, fault tolerance and security (Yu et al., 2013). A number of middleware technologies have been proposed to keep devices connected in a seamless way, allowing people to take advantage of pervasive systems. There are several middleware approaches applied in the design and development of smart spaces and recently the researchers have been focusing in Web 2.0[1]as a platform to provide new smart spaces services (McFaddin et al., 2008; Prehofer et al., 2010).

While different aspects and purposes exist for middleware development, some are common for pervasive systems, equally encountered in projects such as Daidalos (Taylor et al., 2011), Gaia (Ziebart et al., 2005), Aura (Kumar et al., 2003) and One.World (Grimm et al., 2004). Based on Raychoudhury (2013), Yu et al. (2013) and Hong et al. (2009) we present a reference model consisting of three layers for middleware systems

[1]Term coined by Tim O’Reilly

concerning general issues, characteristics and the design of currently solutions. Figure 2.3 presents the architecture middleware for smart spaces.

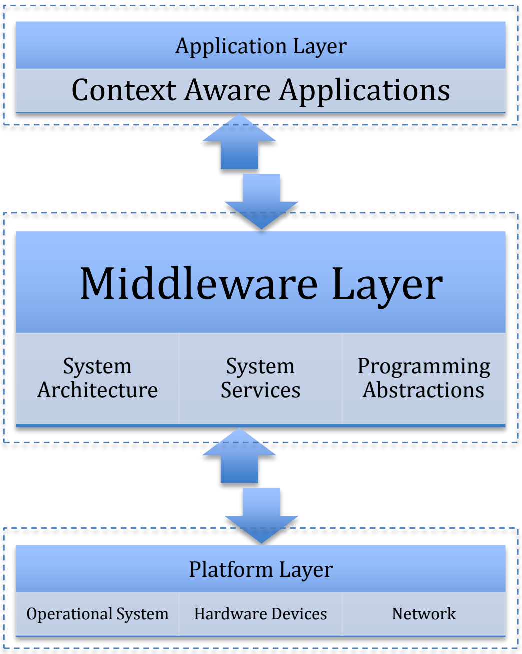

Figure 2.3: Architecture Middleware for Smart Spaces.

2.4.1 Operational systems/Platform layer

Middleware systems are typically constructed on the top of the traditional operational systems (OS). However, there are particular cases in which the middleware present an OS developed specifically for this purpose (One.World, Gaia OS) (Grimm et al., 2001; Román et al., 2002). For most of smart spaces approaches, the use of small and/or mobile devices (e.g. tablet, smartphones and netbooks) is important since they provide sensors of acceleration, sound, light and location. Popular platforms for devices development are Android[2] and iOS[3], depending on the type of the device. On the Linux family platforms, the most adopted are Tizen[4] and Android.

2.4.2 Middleware layer

The middleware layer can be subdivided in system architecture, system services and programming abstractions.

System architecture: In terms of system architecture, middleware design can adopt either top–down or bottom–up design depending on the system model and application

[2] http://www.android.com/

[3] https://www.apple.com/ios/

[4] https://www.tizen.org/

requirements. A top–down approach describes the system functionality at a very high level gradually refining the design at each step, until the detail is sufficient to allow coding (e.g. Gaia, Aura and One.World). Within a bottom–up approach, the design process begins with several small parts and gradually composes them to build the whole system (e.g. PICO). Hybrid approaches are also used in the literature (e.g. PICO)

Additionally, the system architecture concentrates on the mode of system control and interaction among system components. The mode of system control can be either centralized or decentralized. In centralized control, a core component controls the devices and makes decisions on their behalf (e.g. Gaia, Aura and One.World). Consequently, certain functions in the central entity are responsible for free resources in other entities. However, centralized systems have a single point and consequently data is stored only in a central entity. In decentralized control, no device takes a final decision; instead, multiple devices work collaboratively in order to reach a global decision. As an example, CORTEX project used this approach (Biegel and Cahill, 2004).

In terms of interaction, systems depend on mostly in two communication mechanisms: messages passing and publish/subscribe (pub/sub). The message passing is a direct interaction paradigm, in which communication is made sending messages straight to the recipients (e.g. PICO, Gaia). On the other hand, publish/subscribe is an indirectly interaction, where the subscriber register for some events and asynchronously receives messages (e.g. CORTEX, Aura).

System services: System services comprise the core functionalities of the pervasive middleware systems. The principal function is to provide support for application deployment and execution, including devices and network management. Common services include:

- Communication management: responsible for providing abstract interoperability among peers, devices and server;
- Context management: responsible for contextual data acquisition, processing, and derivation of higher-level contexts and context dissemination;
- Service management: responsible for service discovery and service composition in AmI environments;
- Security management: responsible for ensuring correct functioning of the system and ensures protection of user information.

This way, system services provide implementations to achieve the abstractions of programming abstractions.

Programming abstractions: Programming abstractions define the interface of the middleware to the application programmer. The goal is to separate the development of the applications from the operations in the basic infrastructures. Three aspects are involved in developing the programming abstractions: abstraction level, programming paradigm, and interface type.

The abstraction level refers to how the application programmer views the systems, providing facilities to the development of applications. It can be based in node or system level approach. In the node level, the pervasive environment is abstracted as a distributed system consisting of a collection of heterogeneous computing devices, and provides programming support to individual devices in their actions and cooperation facilitating the development of applications with increased flexibility (e.g. Aura, CORTEX). On the other hand, the system level abstracts the environment as a single virtual system, allowing the programmer to express a single centralized program into subprograms that can execute on local nodes facilitating the abstraction of the nodes behaviour (e.g. Gaia, One.World).

The programming paradigm refers to the model of programming the applications. They are divided into context-based, component-based and decentralized interaction based. In the context-based model, triggering events take place through context changes (e.g. One.World). For the component-based model, the application is composed of several component modules (e.g. Aura, Gaia). Finally, the decentralized interaction-based model uses multiple programmable smart entities interacting using some rules (e.g. CORTEX).

Finally, interface type, refers to the style of the Application Programming Interface (API). In fact, programming abstraction is embodied in the programming interface. Pervasive applications require different programming interfaces based on the underlying system architecture and functionalities (e.g. discovery, communication, events).

2.4.3 Application layer

In this layer is where the reactions to context changes are implemented. The client application is implemented on top of the middleware layer providing the users with appropriated solution in terms of environment such as home, hospital, classroom, etc. The crucial challenge for developers is to implement applications that continuously adapt to highly dynamic environments and continue to function even if people move through the physical world (Grimm et al., 2004). More details about context-aware applications are presented in the next section.

2.4.4 Context Aware Applications

The development of context-aware applications is supported by the use of context management systems that implement different operations. This support is provided through exposing functionalities (e.g., via API) that can be invoked by applications. In Chen (2003) they separate context-aware applications (CAA) in active and passive. Active context-awareness comprises applications that change according to the discovered context. On the other hand, passive context-awareness comprises applications that present new or updated context to the user, letting him choose to utilize this information in different ways. To be able to understand the operational aspects of CAA, it is important to analyse the existing applications on the different domains (Chihani et al., 2011). The following is a limited list of CAAs taken from different domains related to the thesis subjects:

- Location-Based Services: it is one of the most developed application type nowadays (e.g. Foursquare). The idea is assist people to locate their friends and interesting places for visiting or where meet friends.
- Context-Aware Communication: this type of applications applies knowledge of people's context to reduce communication barriers. Most scenarios are based on non-verbal and electronic communication services (e.g. SMS, chats and e-mail)
- Context-Aware Recommendation Systems: aim to recommend a service or product to a user based on his context. It is also identified as recommender systems, which combines collected data from users (about products rating, preferences, etc.) to predict the most relevant product or service to a user or a group.

Essentially, the goal is to provide applications that meet user expectations by actions/suggestions that an application should apply in a given situation.

2.5 Context Information in Groups

According to Kalatzis et al. (2013), group context information in pervasive systems is not new, as the first approaches have been discussed as early as the year 2000. Group context involves all the knowledge relating to the group, including group composition, rules, roles, goals, strategies, coordination procedures (Brézillon et al., 2004). Consequently, a step towards this integration is the employment of group or community context awareness.

Recently, Schuster et al. (2013) recognized the importance of the existence of a taxonomy, introducing a classification to pervasive social context along the dimensions of space, time, people, and information source. It is important to take into account not only the characteristics of the group, but each participant as well. In Kalatzis et al. (2014) they propose a middleware to manage user and community context throughout the entire life cycle of context data. In any case, the heterogeneity of context data combined with the introduction of group context information increases the complexity of processing data in terms of storage, modelling and reasoning.

Research on context aware systems capable of support dynamic community management in large scale is still in initial stages. While research efforts have been conducted on a group context, support for conflict resolution context, or for inheritance of context information across hierarchical communities is still a challenging task.

2.6 Pervasive Communities

According to Wenger (2002), a community is spontaneously built and legitimizes various degrees of participation of (new) people based on its internal rules. The degree of participation of a person or actor is proportional to its distance to the centre of the community, physical and/or logical (Cabitza et al., 2006). In pervasive computing realm, this assumption requires technologies able to recognize and support participation of its users in the different kinds of grouping living in its space.

Social awareness plays an important role in order to create a group of users that cooperates effectively and successfully, especially in a smart space (Wang et al., 2010). Considering this, pervasive systems and smart spaces should enable users to create static groups of interest to share common goals and tasks or even take part in dynamic groups. Dynamic groups allow membership through suggestions to the user, based on criteria specified by a given particular community (Taylor, 2013).

Several criteria can be employed to form a community of individuals or organizations, such as sharing same geographic location, having same or similar preferences, sharing some interest or share forms of social relationships. People with shared criteria may belong to a common community of smart space or group. The criterion to join a community can be given by automatic or manual fashion, depending on the user's preferences. Figure 2.4 illustrates a set of smart spaces A, B, C, D and E sharing some specific criterion, forming four new communities. For instance in an educational scenario of E-learning, each community can represent students in different undergraduate programs but attending same courses (e.g. seminar). These communities enable individuals to advertise skills to their groups based on expertise levels or research areas in common. With all the formed groups, it is possible to apply communication and interaction mechanisms with CSCW solutions.

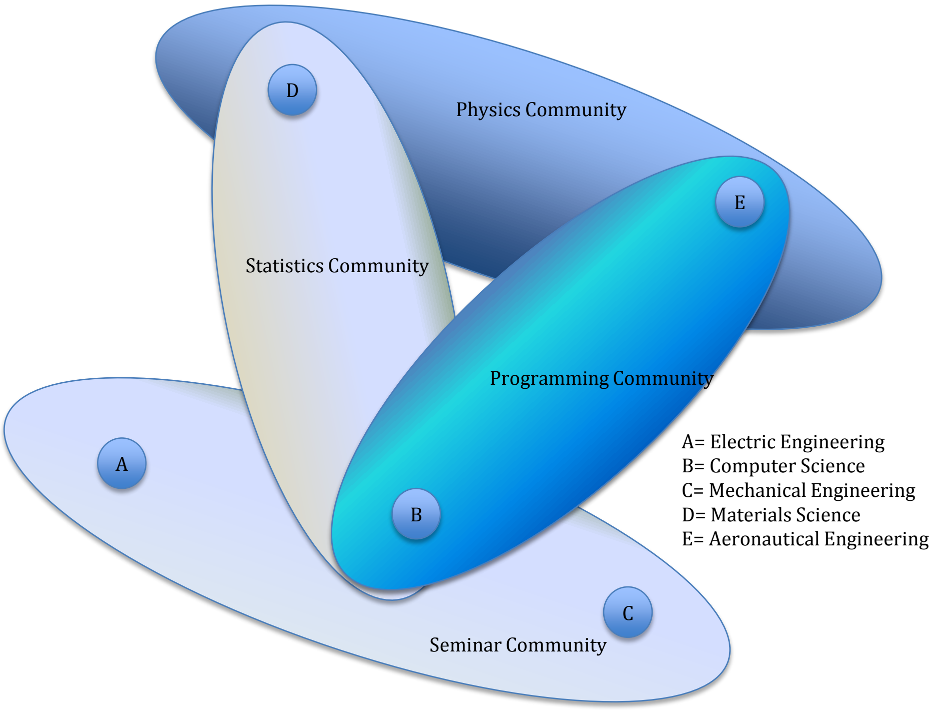

Figure 2.4: Five Communities of Smart Spaces to form four communities

Grouping individuals have always been a subject of study in CSCW (Computer Supported Cooperative Work) and Social Computing fields. In Rheingold (1993), he defines the term virtual community as "social aggregations that emerge from the Internet when enough people carry on those public discussions long enough, with sufficient human feeling, to form webs of personal relationships in cyberspace". This concept is translated to smart spaces as a group of people who share an interest, a craft, and/or a profession. People who participate in such communities learn the accepted work practices, interact with each other and can become more productive.

Most studies approaching groups in pervasive environments available in literature are based on Multi-Agent Systems (MAS). PICO (Pervasive Information Community Organization), Kumar (2003) presents a framework with the objective to create pervasive communities that can collaborate proactively in areas such as telemedicine, the military and crisis management. This framework aims to achieve a sequence of events that can lead to the creation of communities. The framework can detect unusual activity from devices (PDAs, mobile phones and computers) and individually recognizes the occurrence of an extraordinary event. In Kim (2006), it is proposed a context ontology building an organizational architecture composed by society, community and members. They identified four requirements to form cooperation systems and ubiquitous environments: context-aware, role-based interaction, organizational structure and dynamism.

Recent approaches attempt to tackle the problem of pervasive communities supporting also in mobile users. The POPEYE project approached spontaneous virtual communities that can be formed in a P2P fashion for collaborative work (Meyer et al., 2011). However, its main project focus is in transport issues, lacking in services integration and social computing. In Foell (2007) they proposed a framework to form communities according to users' interests and participation. The proposal applies recommendation and personalization algorithms to perform self-organization based on online social sites. Nonetheless, the work only encompasses the context originated from online social sites, not providing management for face-to-face communities and integration in pervasive environments.

The SOCIETIES project introduced the notion of a Cooperating Smart Space (CSS) to define the merging of social computing and pervasive computing (Jennings et al., 2014). Each CSS consisted of multiple devices, both mobile and fixed, owned by a single user, which can interact with other pervasive communities whenever possible. Each community offers several characteristics to its CSSs, such as a set of shared resources and services enhanced by additional functionalities provided by others members of the same communities (Doolin et al., 2012). Members of a pervasive community interact with a CIS via their own personal CSS. A pervasive community, once constituted, forms a Community Interaction Space (CIS) in which individuals may belong to any number of pervasive communities (CISs) simultaneously. Figure 2.5 illustrates the CSS architecture presenting an overview of the core services. The CSS platform developed within the project has a layered architecture with each layer providing various functionalities to enable pervasive behaviour.

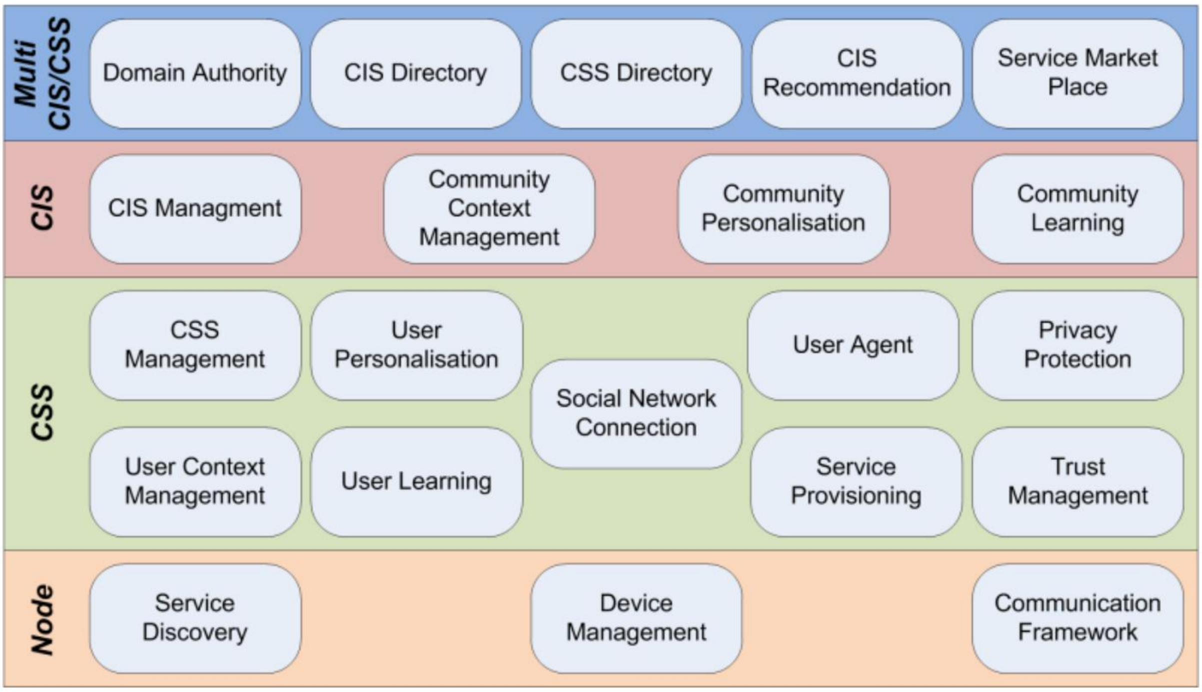

Figure 2.5: Cooperating Smart Space layered functional architecture (Doolin et al., 2014)**.**

The focus of the platform is to facilitate creation, organisation and management of communities, providing better experience for individuals and the communities. The platform architecture supports mobile and/or desktop devices. Each platform component is responsible for a specific task, providing functionalities via APIs for internal components and third party services.

The node layer of the platform manages inter/intra CSS communication, device management and service discovery. The CSS layer provides intelligence and self-improvement, privacy and security, context management, service provisioning and social network connection. The CIS layer comprises the CIS management, community personalisation, community context management and community learning. Multi CSS/CIS layer operate for a wider group of stakeholders, offering federated search and domain administration functions and require multiple CSSs or CISs. The Domain Authority manages the identities of CSS and CIS. The CIS and CSS directories enable searching for CSSs and CISs, based on specific criterion. CIS recommendation is responsible for suggesting communities (CIS). Service Market Place provides access to a repository of installable third party services.

In addition, the platform provides an administration GUI through which the CSS owner can manage the behaviour of the platform, the devices within their CSS, their communities and the sharing and consuming of services. Besides the components described, the platform provides a set of APIs that allow third party services to be built on top of it.

The notion of pervasive community of interests can provide a different impact in the CSCW field. The CSCW can take advantage of pervasive communities in several scenarios as described in the next section, in which users with the same degree of similarities can form collaborative sessions.

2.7 Motivating Scenarios in Enterprise Domain

Muller et al. (2012) highlight several critical dissimilarities in communities regarding its study domain. The authors delineate three main differences concerning context interaction. First, an enterprise domain provides a shared context in addition to the context of the community, which can contribute to a level of trust and common ground. Secondly, enterprise communities are likely to be business-focused, leading to different content and perhaps styles of discussion. Thirdly, companies that require authenticated access and use of real names eliminate anonymity and provide greater transparency. In order to understand the aspects of pervasive communities and its practical use, we developed and analysed two different enterprise scenarios.

Originally inspired by problem faced by small groups collaborating, we proposed two illustrative scenarios in which context-aware can support CSCW applications on pervasive communities (Lima et al., 2012).

2.7.1 Newspaper Scenario

In a journalistic scenario, there are different expertise profiles of employees, which often interact in different levels of tasks and hierarchy. Editors, reporters, photographers and freelancers can belong to a global pervasive community called the "Daily Times Press" that includes the whole editorial and all the subsequent sections.

The employees are also members of the internal newspaper social network, enabling them to share and follow professional content, available only for subscribed communities in the editorial. Each employee can be part of sub-communities, grouped by their work section in the newspaper (e.g. sports, finance), receiving only information relevant to their activities. These areas of interests reflect directly in the sub-communities, in which the interaction within the members is done with less privacy restriction and more information access level than the larger community of "Daily Times Press”. This distinction among sub-communities enables content and context information to be exposed differently for each smaller community. For example, in a photographer’s community who work for multiple sections of the newspaper, it is desirable that content related to photos is disseminated primarily to this group. With the formation of these communities, members can automatically share their content (wiki comments, posts) and resources (printers, projectors).

Through the interaction among members and respective communities, it is possible to acquire context information, interests and preferences of the community and each participant. All this information can be used by the system, which can predict and take decisions automatically. Based on the member’s behaviour it is possible to take into account all their professional agendas to auto-schedule meetings suiting everyone, present relevant information related to activities in mobile phones, invite members to join tasks based on profile skills.

Besides the newspaper static communities, formation of dynamic communities on the fly, especially to perform a common task in a short period is important. This type of situation is frequent in many companies, where it is setup a smaller group of individual and ungroup those after the goals are reached. In a practical example, a dynamic community of reporters and photographers can be formed to cover a given journalistic event.

Typically, in journalistic situations, the crews are often in the street, without opportunities to conduct face-to-face meetings. Through smart devices (e.g. digital cameras, tablets) it is possible to make available pictures, videos, interviews and other contents about the event in real time to the community members such as editors and others reporters. This fast availability of content allows editors to start working on the news story very quickly for publication. Pictures taken on location can be sent with geotags to the newspaper cloud. Audio and video content such as interviews can also be treated in the same way; sent and stored in the newspaper cloud services with geotags and timestamps.

The choice of a conferencing tool is important in cases where mobile collaboration is needed between the reporters, who are out covering up facts and editors who are preparing the newspaper section in the office building. Estimation of collaborative tools like chat, audio and video in these situations is very important in terms of location. In addition, the conferences can be pre-scheduled or postpone taking into account the time and priority from reporters and editors agendas.

Service schedule, media storage and content management can be done in the editorial cloud service, creating a rich history of content. In addition, each community can comment, publish articles, upload photos and record media generating a material for subsequent queries. This content also enables to find professionals through their skills and professional experiences for a given task. Information available from outside the editorial cloud such as social networking (e.g. Yammer[5]) and discussions, since it can contain professional context, may also be integrated as part of the material to enrich the newspaper searches.

The communities also can exchange information and content through inter editorial communities, respecting the privacy level of individuals and communities. Today, with the relatively easy availability of mobile, it is possible to keep the employees aware of real-time activities, by subscribing services of the newspaper's cloud. Treating each community differently is important for the group and the members as well, where the preferences of participants can personalize community behaviour.

2.7.2 Health Scenario

In the local hospital the doctors, nurses and specialists belong to the same major community, called “St. Mary's Hospital” community. Each member of the hospital staff is grouped by specialty and/or specialized centres forming sub-communities.

The doctors are equipped with a smartphone/tablet that is carried all time on the hospital. All the notifications related to current patients are received in the doctors’ device depending on the community involved in care. The notifications can be information including laboratory analyses, electronic medical records (EMR), emergencies and availability of surgery rooms/devices.

Each community of medical professionals provides information such as presence, availability and indoor location for the St. Mary's Hospital community and the ER community.

Display screens are positioned in some hospital places to present information to the community members: e.g. to find other members in the building and what activity they are conducting. Personalized screens will show information based on the member’s community. When a member stays in front of the display, the sensor detects his

[5] https://www.yammer.com/

presence and knows the communities, which he/her are inserted. All the community information exchanged between the devices and St. Mary's Hospital community is granted through security and trusted mechanisms.

Typically, EMR systems store and retrieve history of the patient, clinical analysis, x-rays, dates of appointments, etc. Each patient carries a radio frequency identification (RFID) in the wrist, which can be read by the mobile device, informing the patient's situation. Any new information from the patient sensors or clinical analysis is sent to the community in charge of the patient since his hospital check-in.

A central system orchestrates the medical ambient of the hospital. The environment, particularly surgery rooms and medical appliances (e.g. magnetic resonance imaging), is equipped with sensors that send information directly to the St. Mary's Hospital community. This capability allows for the allocation and schedule of resources as soon as available.

Daily procedures at the hospital demand temporary communities, which are to be formed dynamically. These dynamic communities can be formed for a surgery procedure, diagnosis meeting and videoconferences, aggregating professionals from different skills. In surgeries for example, communities can be formed quickly to attend emergencies, considering the availability and presence of the doctors.

Videoconferences between doctors are set up using 3D cameras that allow for the use of augmented reality in the explanation of medical facts and techniques for disease treatments and surgery. The participants' context can be used to present information on the screen (e.g. name of colleagues by face recognition, specialty). In the laboratory, medical residents can digitally tag equipment with information such as results/tips to help the next resident operate the device. These messages can be represented by air tags[6], which users can visualize via tablet or smartphones with equipped camera. The St. Mary's Hospital community enables the prioritization of resources (e.g. medical appliances) and services, for the sake of emergencies. In addition, technical problems can be addressed quickly, by alerting automatically the technical community for the repair.

2.8 Scenarios Requirements

Below the features and challenges identified from the described scenarios are presented. Some of them took place as functional requirements for CAFCA framework and it is discussed in Chapter 5. Table 2.1 presents the capabilities that the solution needs to support the features and the challenges associated with the newspaper scenario.

Table 2.2 presents the features, solutions to support them and challenges identified in the health scenario.

[6] http://www.sekaicamera.com/

Table 2.1: Features and Challenges of Newspaper Scenario

Newspaper Scenario		
Features	***Supported by***	***Challenges***
Community and sub-communities formation	Static and dynamic groups based on social interaction. Use of group context-aware	Recommendation algorithms and group context-aware sensing, decision-making
Automatic dissemination of information and resources in communities	Broadcast information in the group. Resource sharing among communities based on context-aware and preferences	Group context information management in AmI. Identification of relevant individual. Decision-making and pro-activity behaviour
Media content/history	Media content management for communities, archive of content	Content management based on group context-aware. Storage of massive content in cloud for AmI
Adaptation and prediction for collaborative tools	Context sensors, location awareness, personalization of the tools by dynamic context.	Reasoning and pro-activity behaviour for groups
Communities participation	Suggestion, follows to individuals, communication among groups and social integration	Recommendation systems based on group context-aware. Integration with SNS, pro-activity behaviour
Community management	Intra/inter community communication, merging/split of communities' capability.	Intelligent mechanisms for communities relationships

Table 2.2: Features and Challenges of Health Scenario

Health Scenario		
Feature	***Supported by***	***Challenges***
Community and sub-communities formation	Static and dynamic groups based on social interaction. Use of group context-aware formation	Recommendation algorithms, learning algorithms and group context-aware analyses
Notifications received by medical staff	Intra/inter community communication. Context-aware groups and situational context	Find relevant people and communities based on real-time context. Pro-activity behaviour
Presence and availability	Location-aware and situational context	Discovery of people through precision indoor location. Real-time state of the individual
Real-time patient information	Group communication, dissemination, information management	Group hierarchy management in AmI, decision-making algorithms and pro-activity
Medical ambient orchestration	Management of devices, services orchestration in Ami	Orchestration of people and devices for pervasive communities
Temporary communities for activities	Creation of dynamic groups on the fly, context-aware	Auto-discovery people based on context. Context management lifecycle. Reasoning and pro-activity
Videoconferences with social augmented reality	3D camera sensors, 3D microphone, face and audio recognition	Individual and group context-aware filtering. Reasoning, pro-activity for groupware
Augmented reality message tag	Mobile device, user preferences. Present context related to a community	Group context-aware, reasoning, social integration

2.9 Conclusions

This chapter presented pervasive and context-aware computing to understand the core functionalities and operations of smart spaces, mobility and context-aware application. Features such as dynamic groups, communication channels and pro-activity can be applied in all the future scenarios described.

Several collaborative applications for different purposes can be deployed in these types of environments, such as conferences (including video, audio and chat), agenda scheduling, co-browsing, etc. Some of them are appropriate for communities of smart spaces scenarios and others are not. The inclusion of mechanisms for modelling, reasoning and pro-activity using context and social context can enhance the involvement of the members in collaborative activities as presented in the motivating scenarios.

Finally, the requirements refined from the enterprise scenarios motivated the design decisions for the CAFCA framework. Nevertheless, specific challenges for real deployment of framework in the target environment moved to a conference scenario in the final evaluation.

3 Computer Supported Cooperative Work

This chapter presents the theoretical approaches from the multidisciplinary field of Computer-Supported Cooperative Work (CSCW) for collaborative applications. CSCW studies the collaboration and coordination activities on distributed computers including the development of techniques for dealing with these interactions. The subsequent sections describe the types of collaborative applications, definitions and challenges in design groupware applications. Finally, the chapter introduces the social aspects and small groups involving CSCW.

3.1 Introduction

Until the 80s, most work on supporting groups using computer technology focused on supporting decision processes for collocated groups. With the advent of network technology, the attention shifted towards supporting group and teamwork at distance (Terken et al., 2007).

The growing demands for geographically distributed work raised the need to develop a variety of collaborative environments or systems. CSCW is the generic name for computer-based systems that support groups of people in a common task or goal, providing an interface for collaborative environments (Nielsen, 1996). Users in a collaborative environment are generally a group of people with similar tasks and needs. Typical groupware applications have become increasingly widespread in all fields of research and industry, both for work and learning. By nature CSCW is a multidisciplinary area and grew out of research in fields like sociology, anthropology, informatics, artificial intelligence, cognitive science and social psychology (Stahl, 2011).

There are several classifications for collaborative applications available in literature. One of the most popular categorizations is regarding temporal and spatial features (Ellis et al., 1991):

- Space: refers to geographical distance that separates users from application. For example, members of a meeting can be at the same place or be located in a different place.

- Time dimension: characterizes the type of interaction among the users. Group members can interact at the same time, where the actions of participants are immediately transmitted to others. This type of interaction is called synchronous. Group of users interacting at different times is also possible, in which the actions occur in a period. This type of interaction is known as asynchronous. In this case, it is important the group's activities be saved permanently, enabling members to visualize actions history performed before their arrival.

The table below shows the time/space matrix.

Table 3.1: Time/Space Matrix (Johansen, 1988)

	Synchronous	***Asynchronous***
Same Place	Face-to-Face Interactions	Continuous Task
Different Places	Groupware	Communication + Coordination

- Same time/same place: Decision rooms, single display, shared table, roomware;
- Same time/different place: Video-conferencing, messaging (IM, e-mail), virtual worlds, Multi-user editors, shared screen;
- Different time/same place: Shift-work groupware, public display, physical bulletin board;
- Different time/different place: Wiki, blogs, version control, group calendar;

Nevertheless, according to Grudin (1994), there are numerous criticisms with respect to this classification and several proposals have been developed to refine this along the years. This way, some authors prefer to use a classification by application fields, which is a list of functional categories used to group collaborative applications.

The next section presents in details groupware to provide support for group collaboration.

3.2 Groupware

Groupware or collaborative software are digital technologies used for supporting cooperative interactions in formal activity context alike schools, institution and organizations (Lugano, 2010). Thus, groupware is the main artefact to provide help to people involved in a common task to achieve their goals and typically, the users use more than one application at time. Collaborative work environment (CWE) is the concept of using different collaborative applications to produce a single environment. The goal of these environments is to promote collaboration by providing mutual awareness, communication and coordination (Farshchian and Divitini, 2010). For instance, commercial suites that cover these environments are Adobe Connect[1], Cisco WebEx[2] and Oracle Beehive[3]. Nevertheless, one of the major challenges in these type of environments is to anticipate the needs of their users in different collaborative situations (Gomes et al., 2011).

Below are presented examples of groupware that compose most collaborative work environments:

- Instant Messaging applications: Widely utilized by certain groups of users, instant messaging (IM) enables debate in real-time via text messaging, in any subject of interest of their members. Particularly cases in collaborative scenarios are present in dedicated games servers (Second Life[4], World of Warcraft[5]) and for internal communication in companies and institutions. Examples of such applications tools are Skype[6], Jabber[7] and Google Hangouts[8].

- Chat Rooms applications: Just like in IM, exchanged information is sent via text messaging. Most IM applications allow the user to create chat groups to discuss about a specifically subject. In addition to the already familiar IM tools, there are Facebook Messenger[9], Internet Relay Chat (IRC), and Tox[10] amongst others.

---

1 http://www.adobe.com/products/adobeconnect.html

2 http://www.webex.com/

3 http://www.oracle.com/technetwork/middleware/beehive/

4 http://secondlife.com/

5 http://www.warcraft.com

6 http://www.skype.com/

7 http://www.jabber.org

8 http://www.google.com/hangouts/

9 https://www.facebook.com/about/messenger

10 http://tox.im/

- Conference applications: Conference applications can use both video and audio in real-time for discussions about a subject in common. Some applications of this type include Microsoft Lync Server[11], Citrix GoToMeeting[12] and Apache Openmeetings[13].

- Co-browsing applications: applications for co-browsing enable a group of users navigate the Web content together, each one on their own computer. Examples are LiCoB (Santos et al., 2011), CoLab (Jesus Hoyos-Rivera et al., 2006), and Collaborative Web Browsing (Franke and Cheng, 2013).

The evolution of hardware technologies, software combined with the continued maturation of the Internet and ubiquitous computing, provided intensive use of advanced applications and services, creating serious implications for the CSCW matrix (Schneider et al., 2012). CSCW needs to deal with new paradigms as social web, ubiquity and crowds to pursue a 'CSCW 2.0' model.

Through hardware improvements, this achievement was possible due to wireless networks (Wi-Fi, LTE) and positioning technologies (GPS, RFID), while the evolution of software came through the Web 2.0. New technologies such as HTML5 and WebRTC are also enabling substantial advances in developing complex web applications for video, chat and P2P communication. The HTML5 is a new specification for WWW and is expected to be a strong candidate for cross-platform mobile applications. WebRTC[14] (Web Real-time communication) is a framework that lets Internet users communicate in real time via video/audio by using a compatible browser without installing additional plugins. For each device (computers, smart phones) with an installed browser, it is possible to perform peer-to-peer real-time communications natively, for instance, video and voice calls, chatting or instant messaging, file sharing and screen sharing (Amaral et al., 2014). WebRTC uses an API definition enabling real-time communications capabilities via native JavaScript APIs. Thus, initiatives such as Matrix[15] provide an open standard that enable the creation of services providing VoIP and IM using WebRTC and HTML5 as an alternative to XMPP (Extensible Messaging and Presence Protocol). XMPP is a communication protocol based on XML (Extensible Markup Language) and used widely over the Internet.

[11] http://products.office.com/en-us/lync

[12] http://www.citrix.com/products/gotomeeting

[13] http://openmeetings.apache.org

[14] http://www.webrtc.org/

[15] http://matrix.org/

3.2.1 Asynchronous and Synchronous Communication

Groupware applications with support for asynchronous communication and are usually denoted as messaging systems. These systems are utilized for the asynchronous exchange (i.e. do not occur at the same time) of messages with a relatively structured form and/or contents (Hinssen, 1998). Messaging is considered the most mature and diversified of groupware applications. This is exemplified by email and bulletin board systems.

On the other hand, conferencing systems belong to the class of groupware applications that are intended to support synchronous (real-time) communication. Each member of a group, which can be geographically distributed, has access to a shared information space. This is exemplified by audioconference and videoconference.

3.2.2 Coordination

Coordination refers to managing activities that occur synchronously and asynchronously on groupware shared among collaborators (Cabitza and Simone, 2013). In Grudin & Poltrock (2012) they argue that small groups or teams have interest to work together in real time, while communities prefer to communicate and share information asynchronously. To support real-time collaboration across distance, features such as floor control and session management are desirable to facilitate coordination. While for asynchronous coordination, tools as workflow management, case tools and calendar/scheduling (Poltrock and Grudin, 1999).

In a collaborative application, floors may be associated with shared resources to provide exclusive access. The user holding the floor is the one with the right to control the associated resource. Floor refers to the need for coordinating activities occurred in synchronously cooperating applications among actors. For example, in audio conferences, the floor control mechanism can avoid too many persons from speaking simultaneously. Consequently, only authorized users can talk-and-listen in a conference while the others can only-listen.

Several floor control mechanisms have been proposed for conferencing, including the use of a centralized chair to control the floor, by election, by user's role (e.g. lecture speaker), and by some simple policies such as First-Come-First-Served, Least-Recently-Served to prioritize floor requests (Joung and Chien, 2008).

In addition, another possible behaviour is to combine floor control strategy for integrated collaborative applications among two or more applications, to assure that the users have the same permissions granted in all applications.

In order to provide organization among multiples users, objectives and group work relationships; a collaborative session must support formal or informal meetings for work cooperatively. Edwards (1994) defines a session management as the process of starting, stopping, joining, leaving and browsing collaborative situations across the network. The author also states that session management takes two practises: explicit and implicit. In an explicit session, the participants in collaboration are required to take some action when joining a session. The explicit sessions can follow two approaches:

- Initiator-based: consists in a sequence of dialogs that an initiating user performs to invite other users to the collaborative session, with the invited users accepting or rejecting the invitation
- Joiner-based: comprise in the initiating user creating a new session, making other users that want join him to browse the list of current active sessions.

Moreover, the author discusses the practicality in situations where a high degree of formality exists or a natural name for the activity exists.

On the other hand, an implicit session is based on activity information, where the system must be aware of what are the current activities running across the network, users and the applications that they are currently engaged. In this way, the system would simply detect the potential for collaboration when multiple users are working on the same object, establishing a session without the need for session creation, naming, and browsing lists of sessions. This practice is indicated for serendipitous and transient collaborations as opposed to planned and long-term formal collaborative sessions.

3.3 Social Aspects in CSCW

The collaboration support is exclusive to providing technologies and tools, but is also about shaping socio-technical systems according to Koch (2008). More recently Stahl (2013), approached CSCW and CSCL (Computer Supported Collaborative Learning) studies taking into consideration the social aspects and categorised it in three levels: individual, small-group and community units. Albeit the author does not approach pervasive computing, the analysis is also interesting for both fields as exemplified below:

- **Individual**: It is based on the principle that the individual constructs his/her own understanding of the reality. The individual uses resource and he/her experiences;
- **Small groups and group teams**: In small groups, the members usually know each other. A group collaborates to achieve a common goal and it is highly focused. E.g. software development team, conference program committees, research project teams.
  Group teams can be considered part of small groups however they have some particularly differences. Generally, they are geographically distributed and have hierarchical management structure. Both small groups and group teams have a strong need for communication. E.g. companies, governments, non-profit organizations;
- **Community**: In terms of characteristics, the members usually do not know each other. Nonetheless, they have common interests or preferences with loose interactions. E.g. citizens, newsgroups, auction participants;

The figure 3.1 represents our point of view of interactions among the parts for CSCW and pervasive computing.

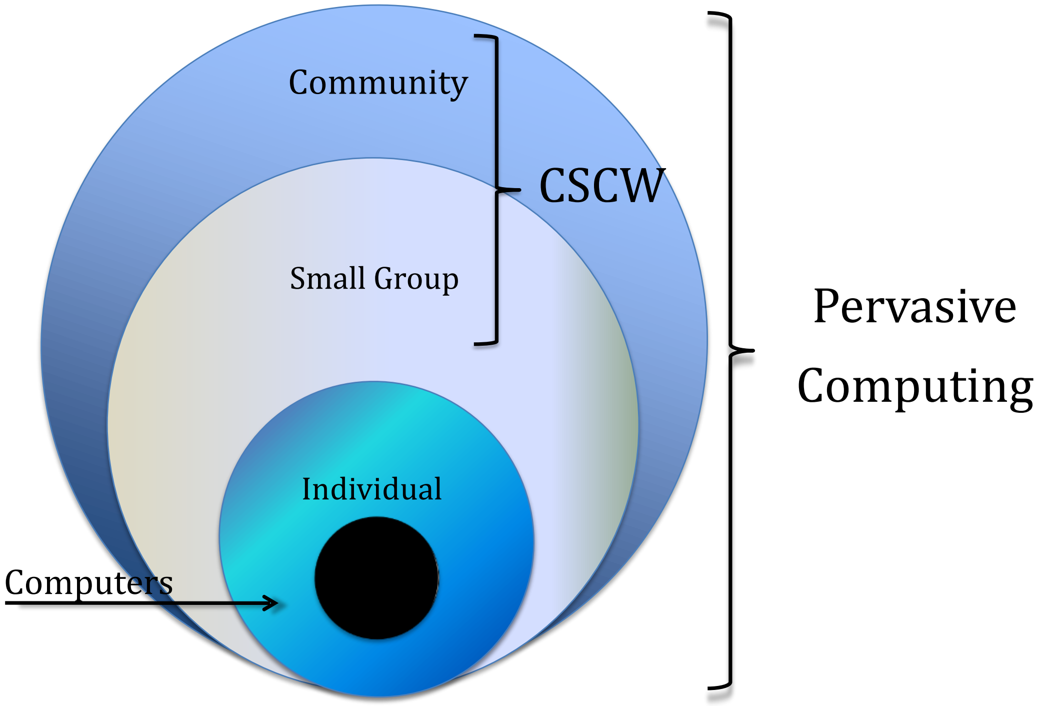

Figure 3.1: Interactions among the parts for CSCW and pervasive computing

The author highlights that although cooperative work may be coordinated across large communities of practice, tasks are typically accomplished by small groups. Using a similar approach, we adapted some of the terminologies and present table 3.2.

Table 3.2: Terminology for phenomena at the individual, small group and community levels of description. Partially adapted from (Stahl, 2011)

Level of description	Individual	Small group	Community
Role	Person/worker/ student	Group participant	Community member
Adjective	Personal	Collaborative	Social
Object of analysis	Mind	Discourse	Culture
Unit of analysis	Mental representation	Utterance response pair	Socio-technical activity system, mediating artefacts
Form of knowledge	Subjective	Intersubjective	Cultural
Form of meaning	Interpretation	Shared understanding, joint meaning making, common ground	Domain vocabulary, artefacts, institutions, norms, rules
Learning activity	Learn	Build knowledge	Science
Communication	Thought	Interaction	Membership
Mode of construction	Constructed	Co-constructed	Socially constructed
Temporal structure	Subjective experiential internal time	Co-constructed shared temporality	Measurable objective time
Tacit knowledge	Background knowledge	Common ground	Culture
Action	Action	Inter-Action	Social praxis

3.4 Awareness in Collaboration

Awareness and awareness support systems in collaboration has been subject of studies since its origins. Various awareness classifications have been proposed in the past for CSCW (Gross et al., 2005) and social studies (Markopoulos, 2009) such as: situation awareness, task awareness, organizational awareness, objective self-awareness and so on . Almost all definitions focus on the awareness of user, rather than on systems or their environment. This can be contrasted to the concept of context-awareness that has also been studied in CSCW (Kulyk et al., 2008).

While classifications by Gross et al. (2005), Gutwin and Greenberg (2002) are on the subject of CSCW, the definitions from Markopoulos (2009), Kulyk (2010) recently present concerns for a more social practice rather than only cooperative work.

In Gutwin et al. (1996) they propose a classification of awareness presenting a set of basic ideas considering the design of awareness support. The classification is represented by four types of awareness: workspace, informal, group-structural, and social awareness. In terms of collaborative awareness design, this approach is still relevant for several areas that can differ from medical to distributed software development as reviewed in Steinmacher et al. (2012).

Following the characteristics of awareness introduced in the conceptual framework by Gutwin et al. (1996), it is present the concerns applied to the framework in this thesis illustrated by figure 3.2.

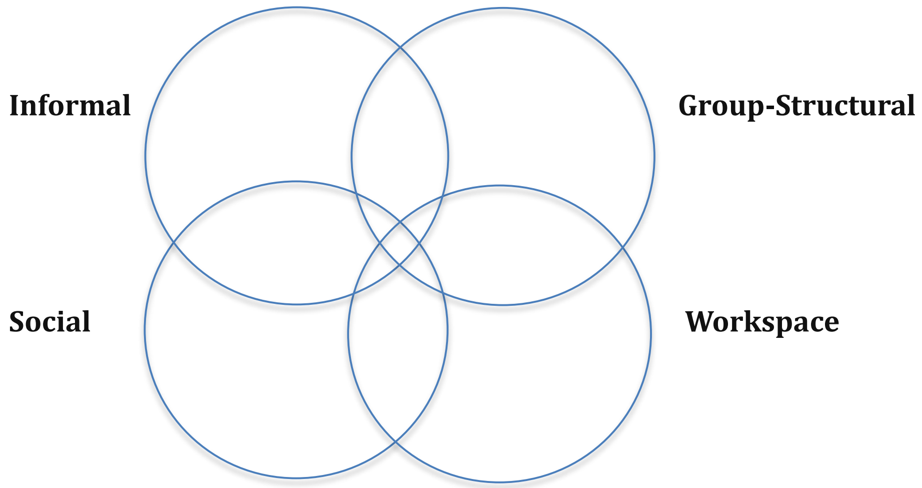

Figure 3.2: Types of awareness according to Guwin et al. (1996).

Below the four types are described in details.

- **Workspace Awareness**: The workspace awareness is the up-to-the-moment understanding of another person's interaction with the workspace. It helps people move between individual and shared activities, provides a context in which it is possible to interpret other's utterances, allows anticipation of others' actions, and reduces the effort needed to coordinate tasks and resources. This includes real time changes that occur on distributed workspaces, presence, location and identity of the participants.
- **Informal Awareness**: Refers to the general sense of who is around and what they are doing. This kind of information is typically related to presence awareness, since it is associated with technologies, such as IM and VoIP, which enable colleagues at distributed locations to be aware of who is online, offline, busy, etc.
- **Group-Structural Awareness**: Group-structural awareness is related to the expertise of the collaborators based on the roles and responsibilities they assume.

This knowledge is important in choosing with whom to start an interaction. Thus, roles and responsibilities awareness elements focus on finding out experts on a given subject.

- **Social Awareness**: The social dimension creates opportunities to publish personal information helping to establish a shared context among users, with little or no chances to meet physically. This way, social awareness relies on knowing the work context of a person. This includes availability, social/personal information. Availability in this case differs from presence since it offers specific information about the possibility of being interrupted or not.

Importantly, the definition at the time did not consider context-awareness research on sensor-based smart environments, but people awareness. Consequently, the term "awareness" in collaborative work has many disagreements as exposed in Karat et al. (2012). Oulasvirta (2009) quotes for example: "The boundaries among technology, human, and action have become blurred. Omitting social inference from the analysis may have led to the untenable conclusion that awareness can be almost anything". Additionally, the author states: "The quest for an answer is fuelled by the on-going ubiquitous computing revolution, which will soon warrant more imaginative and pervasive forms of awareness."

3.4.1 Awareness in different domains

There are different domains for collaborative work, which implies that new domains mean new awareness behaviour and new requirements for awareness support. The domains can vary from home living, healthcare in homes and in hospitals, education, workplaces and many others (Karat et al., 2009).

Some important issues related to almost all areas are privacy and interruption. In the case of privacy, depending on the domain it may be desired. For example, the domestic domain is very different from workplaces, where sharing personal information and intimate relationships are not disturbing. In Rittenbruch et al. (2007) they expose a concept of passive and active awareness which is very correlated to privacy. In active awareness, the users intentionally provide information to others and the system, while the passive system needs to consider only the information exposed by the user and the application to take decisions.

Interruptions are other significant requirement in collaborative work. Sometimes the participants do not want to be distracted. Again, the domain will determine this behaviour. In cases such as crisis (Wood et al., 2012) and hospital/healthcare domains (Fitzpatrick and Ellingsen, 2012), this is not an option. In organizations and offices most conversations are unplanned, and hence can be potentially interrupted (Bardram and Hansen, 2004).

3.5 3C Classification of Supported Functions

The 3C model created by Ellis and Wainer (1994) classifies groupware systems into three aspects: communication, coordination and collaboration. The model pursues to characterize cooperation in three different levels, depending on the intensity of the relationships between individuals and tasks performed. Communication or conversation comprises in the exchange of messages and in negotiations among people; coordination consists in the management of people, their activities and their resources; and cooperation is the production that takes place in the shared workspace (Fuks et al., 2007). The factor is defined as the relation of distance between the placement of the groupware and the corners of the triangle illustrated in figure 3.3. This definition implies, for example, that a chat can be seen as a communication tool that requires communication (exchange of messages), coordination (access policies) and cooperation (registration and sharing) (Cheaib et al., 2011).

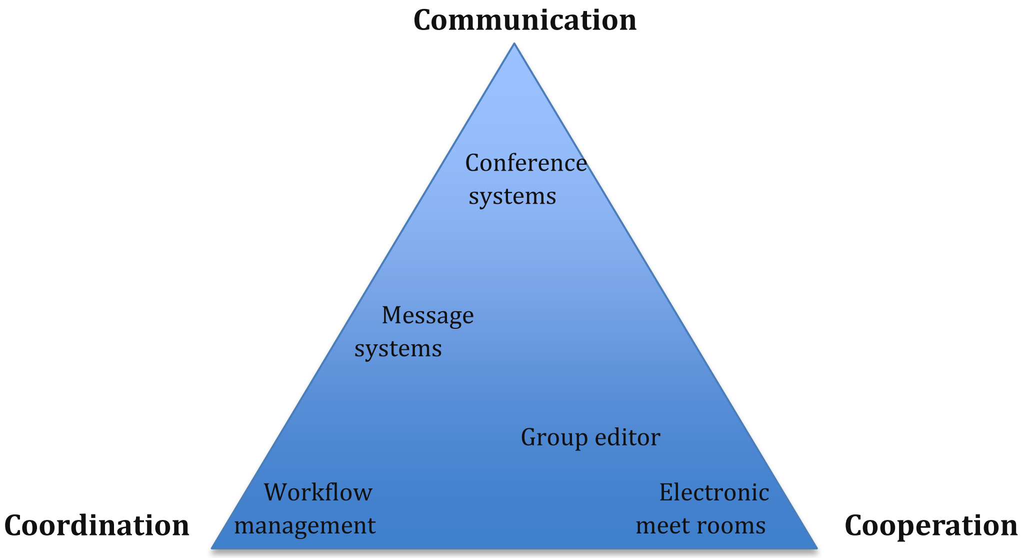

Figure 3.3: Classification of collaborative applications based on 3C model (Borghoff and Schlichter, 2000)**.**

The 3C model is also used as an analysis tool for different purposes, design, evaluation and implementation of collaborative systems (Fuks et al., 2008). Recently, a literature review done by Steinmacher et al. (2012) indicated that coordination is by far the most supported dimension of the 3C model, while communication is the less explored. According to Cheaib et al. (2011), the use of the 3C model it is not mandatory, but helpful as a guidance to analyse a groupware application domain. The authors also sustain that an optimal collaboration pattern is achieved when users initiate by a communication phase, then a coordination phase and eventually cooperation.

3.6 Context-awareness in CSCW

Applying key concepts of coordination, communication and cooperation to pervasive environments is particularly challenging. It implies users that may move in the physical location and change their interdependences in terms of interaction (physical space) and cooperative application (logical space) (Cabitza et al., 2006). In this manner, a pervasive environment in collaborative activity is the composition of a physical space including devices and persons, with a logical space where persons perform some activity.

Context and context-awareness becomes an essential subject in design and realisation of many systems and a crucial factor for their success (Schmidt et al., 2004). Despite the combination between CSCW and context-aware systems is reasonably recent in literature, successful cases can also be found on mobile environments (Bardram and Hansen, 2010), web (Kulkarni et al., 2012; Sakurai et al., 2008) and others fields. Lately, CSCW context-aware applications have been extended beyond the desktop to different work settings and into the domain of mobile devices. Mobile devices are by default close to its users by its social nature. Johnson (Johnson, 2013), highlight the opportunities in research that mobile devices can bring into the CSCW area supporting “anyone, anytime, anyplace” concept.

In order to enable applications to adapt themselves to discovered context, context-aware applications should be designed to discover and present contextual information that serves as resources for cooperative work (Schjørring, 2004). In Tripathi et al. (2005) for instance, the authors proposed a framework, classifying context information of CSCW applications into two categories: internal and external. The first is related to the execution of operations and reactions in a task. The latter is defined as different attributes that are related to the environment such physical location, Internet domain or devices interacting with user’s environment.

However, according to Haake et al. (2009), several proposals have been made lately to consider a more general interpretation of context to include non-physical aspects such as the user’s interests, tasks, or interaction behaviour. The authors also highlight that a generalized view of context and methods for use in adaptive interactive systems are still missing.

In this way, the approach is more significant for groupware, where the context can be seen not only from the perspective of the individual user, but be incorporated in the group as a whole. Michael Koch (2008) highlight the emerge of Web 2.0 and Social Software fields towards support for collaboration activities. More recently, he complements with community attention, concerning that implementing a CSCW system means to design a complete socio-technical system, including organizational and social aspects (Koch et al., 2014).

3.7 Collaboration Environments

There are several works oriented to improve integration flexibility of collaborative environments as discussed in Gomes et al. (2011). Four main approaches are identified for this: user-tailorable solutions, CSCW toolkits, middleware based solutions, and platforms for integration of collaborative systems. The follow concerns the aspects of context-awareness in groupware.

Different definitions of tailorability can be found in literature. Most of them focus on user tailorability defining that tailorable applications can be adapted and modified by each user in order to meet their different requirements. In CSCW, tailorability must concentrate on the requirements of the group task and the organization wherein the CSCW system is used. According to Mørch (1997), tailoring can be supported in three different levels: *customization*, selecting among a set of predefined configuration options; *integration*, linking together predefined components within or between applications; *extension*, improving the implementation by adding new program code. However, most of user-tailorable groupware tools support only the *customization* or *integration* level (Cheaib et al., 2008). As an example, the InContext project uses web services and semantic web technologies and provides a platform that captures diverse dynamic aspects of team collaborations. To date, many solutions are now focusing in web-based collaborative systems converging in Software as a Service (SaaS) concept. In a SaaS system, vendors host applications on the web and deliver them via browsers to users, who perform and store their work online (Li et al., 2012). Examples include Cyn.in[16] and Zimbra[17].

CSCW toolkits facilitate the implementation of CSCW systems by providing reusable components and behaviours designed to be applicable in a range of circumstances (Dourish et al., 2000). As an example, Intermezzo is a collaboration support environment supporting the coordination information sharing, offering fluid interactions, user awareness, session management and policy control (Edwards, 2005). It addresses dynamic flexibility by allowing applications to adapt not just their own behaviour, but also the behaviour of the toolkit reacting to the changing dynamics of the world they run into. Toolkits may represent an interesting solution for helping the development of CSCW systems, as they promote the reuse of components. In general, to reuse components of the toolkit, developers often need to implement very specific details of the toolkit in order to adapt it to the user needs (Pichiliani and Hirata, 2009).

Middleware-based solutions enable the integration of heterogeneous applications, mainly in distributed systems area. General integration solutions based on middleware, like CORBA, .NET and Enterprise JavaBeans. The emergence of web service has led to the development of general solutions for integration of distributed applications, due

[16] http://cynapse.com/cyn-in/

[17] http://www.zimbra.com/

mainly to the use of open standards. As an example, the ECOSPACE project proposes an environment that relies on semantic web service technologies to support semantic description of collaborative services. Besides a semantic description of each service, a semantic description of the composition of services is required to coordinate their orchestration. However, this part of the project remains at a design level.

Integrated Collaboration Environments (ICEs) purpose is the integration of different collaborative applications into a single and easy-to-use operational environment (Fox, 2005). A platform for integrating collaborative applications aims to improve the combination and flexibility of the groupware. They focus on the integration of collaborative functionalities provided by these applications while trying to define any semantics behind integration. In order to avoid considering application internals during the integration process (facilitating the integration of existing applications) some solutions propose a loosely coupled approach. This approach presents two main features: once integrated, collaborative applications preserve their autonomy, i.e., they can still be used as standalone application; the integration environment remains independent of integrated applications, and accordingly, applications can be integrated and detached from the environment without compromising its behaviour (Gomes et al., 2005). This last feature is particularly important considering the integration flexibility aspect. An example is the framework XML-based General Session Protocol (XGSP) (Kim and Fox, 2011). The XGSP proposes the integration of conferencing tools based on SIP, H.323, and Access Grid18 standards. XGSP manager servers are in charge of controlling collaborative sessions. A different gateway is defined for each application type. Using a signalling protocol based on web service, these gateways are employed to mediate the communication between applications and XGSP servers. An important disadvantage is the fact that it only allows the integration of three standards cited.

3.8 Conclusions

Since the 80s until the turn of the millennium, groupware had been the exclusively form of social computing, while social software only recently was introduced by business-oriented communities aiming at revitalizing the Internet (Lugano, 2012). Some authors argue that groupware and social software are incompatible, citing that the first is academic oriented while the last is business-oriented. According to Fuchs et al. (2010), groupware are traditionally developed to meet the needs and profiles of the collective group, instead social software typically provides an self-centred view of the individual social world. However, the synergies between areas can bring innovative aspects, where social relations can help the interaction among users. While social software seems to be contained in the CSCW area and is characterized by techno-centric definitions, social computing is more comprehensive, involving long date studies in theory and analysis of

[18] http://www.accessgrid.org/

social network. The next chapter present the social computing taking into account the other areas of the thesis.

4 Social Computing

This chapter presents the social computing and its techniques for social network analysis. Social computing studies technologies that consider social context, human interactions analysis in digital realm and the design of social computational systems. Subsequently the concept of communities of practices and small groups are described. Finally, this chapter discusses theory of social network that can be applied in real-world applications.

4.1 Introduction

Social networks have been studied for years and are a research field in disciplines such as epidemiology, sociology, economics, and others (Narayanan and Shmatikov, 2009). The area characterize social context by "Information relevant to the characterization of a situation that influences the interactions of one user with one or more other users" (Wang et al., 2010).

The growth of online social networks such as Facebook, Google Plus, Twitter, and so on has attracted the attention of computer science for social studies (Kossinets et al., 2008). Social computing has become also very popular field in the current IT industry, mainly due to the emergence of the Web 2.0. Through Web 2.0 it is possible to enable provision of content, goods and services in areas of life and business by anybody and everybody (Taylor, 2008). However, the term is controversial and some authors just consider it a reuse of old concepts, while others a new philosophy (Koltay, 2010).

Social computing is defined as the use of systems to support online social interaction (Erickson, 2013). Wang et al. (2007) define it as "the computational facilitation of social studies and human social dynamics as well as the design and use of computing technologies that consider social context". Thus, social computing is interdisciplinary by nature and includes other research domains of computer science such as groupware, social software and mobile social software.

4.2 Social Capital

Social capital[1] represents the value a person gains from social ties that exist between individuals and groups. According to Bourdieu and Wacquant (1992), social capital is the "sum of the resources, actual or virtual, that accrue to an individual or group by virtue of possessing a durable network of more or less institutionalized relationships of mutual acquaintance and recognition". Social capital is different from human capital, while human capital represents individual characteristics (attributes), the social capital benefits from relationships from his social networks (Monge and Contractor, 2003).

In literature, the human capital refers to the resources that individuals bring to the table such as their education and experience. Robert Putnam defines social capital as features of a society that help facilitate and coordinate actions within that society (Putnam, 2001). These features include social networks, norms of reciprocity, and levels of trust (Prell, 2006).

By this definition, a more interlinked community has a stronger social capital and is more resilient to external pressures and challenges. Recent approaches are taken into account the social capital for computer science studies addressing: web based communities, virtual communities and social media (Forte et al., 2014; Koch et al., 2014; Schreurs and de Laat, 2014). Karat et al. (2012) highlight Putnam statement, contemplating a collaborative work perspective. He states that: "construction of social capital is an important feature for social organization and systems should support social relations, including norms, networks and trust that facilitate cooperation and coordination for mutual benefit".

4.3 Communities of Practice and Interests

Individual context and social context of a person take into consideration situations determinate by time and space activities. While the individual context focuses on the person himself, the social context emphasis relies on social interaction distance. Consequently, a group of people with a common interest or practice who share information and/or network can be considered a community.

In line with anthropologist Wenger landmark work (Lave and Wenger, 1991; Wenger, 1998) a community of practices (CoP) is "groups of people who share a con-

[1]Coined by French philosopher Pierre Bourdieu

cern, a set of problems, or a passion about a topic, and who deepen their knowledge and expertise in this area by interacting on an on-going basis". There are three structural elements of CoP proposed by Wenger in further studies (Wenger et al., 2002):

- Domain: represents common ground where participants share their knowledge.
- Community: is a group of people who interact together, facilitating building relationships related to that domain.
- Practice: is the specific knowledge that the community develops, shares, and maintains. In addition Gunawardena et al. (2010) state, that the nature of the tool that mediates communication impacts and alters their perceptions of the communication process as well, and how they perceive their social roles.

A CoP develops among people who practise the same trade or share the same working conditions. The CoP emerges from collective activity and is the result of the involvement of individuals in the actions of a practice. For individuals, a virtual CoP represents a means of investing themselves in the social or professional definition of trade, to reinforce their identity, to enrich practice while contributing to the practice of the community (Henri and Pudelko, 2003).

Similarly, a community of interest (CoI) is a community of people oriented around a topic of common interest. The members take part in a CoI to exchange information, obtain answers to personal questions, improve their understanding of a subject, and share common passions.

Both communities are characterized not only by the interaction in a virtual level. From point of view of Fischer (2001), a CoP is considered an homogeneous community, which comprises practitioners with similar work. On the other hand, CoI are considered heterogonous. This means that CoI are formed from different CoPs members for a common concern.

Communities of interest and practices have a variable lifetime. Some appear and disappear soon after their formation, while others can succeed for long times. Usually CoIs are more temporally than CoPs, since they are considered a community of representatives of CoPs and takes place based on specialised topics of interest.

4.4 Social Computing Classification

Lately, the impact of social computing has been intensifying due to social software and social networks sites (SNS). The interest of groupware research in CSCL and CSCW for informal social interactions only recently has been subject of investigations. Following this basis, Lugano (Lugano, 2012) presented a classification of the existing social computing with scope not only in groupware but also in mobile social software (MoSoSo). MoSoSo refers to social software designed for interactions in mobile contexts. A classification, based on the work of Shirky (2005), took into consideration static/mobile mediated interactions and formal/informal social relations. Table 4.1 presents the classification that considers two dimensions: the style of social interaction and the nature of interaction.

Table 4.1: Suggested classification of social computing (Lugano, 2012)

	Formal Social Interaction	*Informal Social Interaction*
Static Interaction Context	Groupware	Social software
Mobile Interaction Context	Groupware	Mobile Social Software

4.5 Social Media and Social Software

Social media is a term associated to social practices of participants to create, share and comment among a variety of Internet applications. According to Lerman (2007), social media sites share four characteristics: users create or contribute content in a variety of media types; users annotate content with tags; users evaluate content, voting or passively by using content; users create social networks by designating other users with similar interests as contacts or friends.

Some representative examples are:

- Authoring tools: Blogs, Twitter
- Collaboration tools: Wikis, Wikipedia,
- Tagging systems: delicious, Flickr, Reddit
- Social Networking Sites: Facebook, Yammer, LinkedIn
- Collaborative Filtering: Amazon, Yahoo! Answers

Social media sites are constantly growing. Figure below presents a scheme called "The Conversation Prism", which tries to illustrate all the present types of social media.

Figure 4.1: The conversation prism V4.0 by Brian Solis & JESS3[2].

The term social media is predominantly used by business and marketing, while social software invokes a scientific association. Social software application is a category of software that allow user to collaborate and communicate facilitating interactions. In Shirky (2005), the author also considers groupware as a social software since it support group interactions. Examples of social software include instant messaging, chat rooms, forums, social network services, etc.

In terms of possibilities of using social software, Koch (2008) tries to classify social software similarly to the 3C model presented for CSCW (presented in figure

[2] https://conversationprism.com/

3.3). The figure 4.2 presents a software class's organization. The approach has three core concepts: information management, identity and network management, and communication.

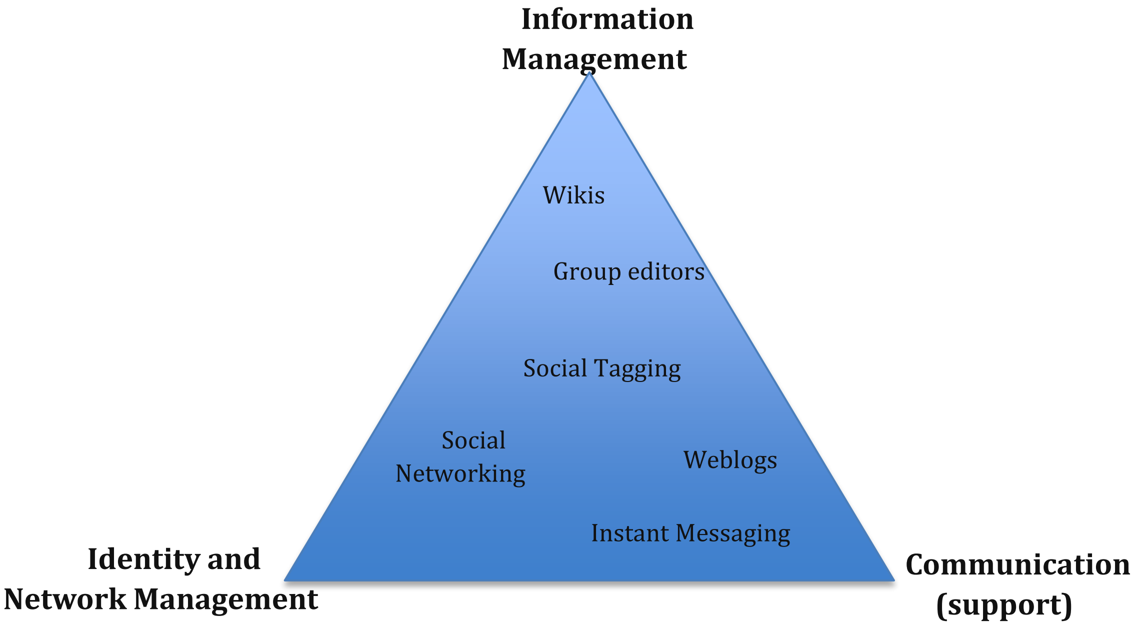

Figure 4.2 Social Software Triangle (Koch, 2008)**.**

The model was originally planned for social software in enterprise 2.0 realms. Nevertheless, the model can also offer an overview to support collaborative work in a general manner.

Many companies following the Web 2.0 trend started to adopt the same platform, but instead focusing in the enterprise field. The enterprise 2.0 refers to the use and dissemination of the social software sharing philosophy among employees of the same enterprise, as well as beyond enterprise boundaries. Thus, an enterprise social software is seen as a component of enterprise 2.0, consisting of the use of social software in a business context.

4.6 Social Network Sites

Online social networks platforms have gained much popularity due to Web 2.0. The SNS are the most known from all social medias. This major trend is in part due to the development of applications, which led to differences in available services among existing social networks. These applications allow access to various types of information, such as profile, activities occurred, friend list, etc.

Although, the existing SNS basic services and network model are very similar to each other, they not offer user privacy, user control and interoperability. To tackle these technical issues, the web community has developed two solutions: API based solutions and Decentralised Social Networking Systems (DSNS) (Yeung et al., 2009). There are initiatives such as OneSocialWeb[3] and Diaspora[4] to create open standards for communication among social networks, embracing the DSNS philosophy.

Another particular use of the APIs is to create mash-up applications. The mash-ups objective is to combine data by one or many SNS to produce new client applications or web sites. This way, social media presents an opportunity to explore user's data to assist a variety of scenarios.

4.7 Mobile Social Software

Mobile social software (MoSoSo) is a category of mobile applications to support social interaction among interconnected mobile individuals. The term has its origins in social software and groupware (Lugano, 2007) . The author claims three important differences between desktop and mobile environments: physical context, social context and distinction concerns.

- The physical context moves from the desktop, where the user is static, to a more dynamic mobile context, which presents higher constraints to human attention. Subsequently it provides an opportunity for information or communication at anytime and anywhere.
- The social context becomes broader, shifting from the group to the network concept, where boundaries cannot be easily identified. A mobile social network is the social space determined by the MoSoSo application.
- Distinction concerns are fundamental for MoSoSo applications. While MoSoSo applications are designed for usage in everyday life situations, collaborative applications aim at increasing productivity and teamwork. From this perspective, MoSoSo is more similar to social software than to Groupware applications.

Nevertheless, these notions overlap the smart space interactions concepts, which take into consideration user-to-user, user-to-resource and resource-to-resource interaction (Gilman et al., 2013). In this manner, although the popularity of mobile groupware applications has recently been growing, most of these are merely a miniaturized version of their desktop ancestors (Farshchian and Divitini, 2010). The social interaction should not only be based on user-centred services such as SNS, but also consider physical context as an active participant of interaction.

[3] http://onesocialweb.org/

[4] https://joindiaspora.com/

The SNS and location-based social networks are also present in mobile devices. Mobile Social Networking (MSN) comprises the use of contextually aware, pervasive, distributed computing, and sensor networks to bridge the gap between the physical and online worlds (Chin and Zhang, 2014). MSN enables to enhance the capabilities of more traditional SNS. The concept is not restricted only to SNS, but for applications that take into consideration mobility-related context, such as physical location and co-presence (Bellavista et al., 2013).

4.8 Socially Aware Computing

Recently the integration of pervasive computing and social computing resulted in a new emerging research field called Socially Aware Computing. While the concept of social awareness has been developed in the field of CSCW for years, the concept of Socially Aware Computation and Communication has only recently been raised by Alex Pentland (2005). This new paradigm promises to bring new focus on the design of new software methodologies, infrastructure, data analysis, and applications (Yu et al., 2013). The field aims to leverage the large-scale and diverse sensing devices that can be deployed in human daily lives to recognize individual behaviours, discover group interaction, and support communication and collaboration (Yu and Zhou, 2014).

4.9 Social Network Analysis

Social network analysis (SNA) emerged as an important research topic in sociology, with first studies focused on the adoption of medical and agricultural innovations (Coleman et al., 1957; Valente, 1995). It is an interdisciplinary topic and has attracted researchers from psychology, anthropology, economics, geography, biology, and many others (Bonchi et al., 2011).

The SNA sees social relationships in terms of network theory, consisting of nodes and ties (also called edges or connections). A social network comprises of a set of actors (nodes) and the relations (ties or edges) between these actors (Wasserman & Faust, 1994). The nodes are the individual actors within the networks, while ties are the relationships between the actors.

This way, a social network is a map of specified ties between the nodes being studied. The nodes to which a person is connected are the social relationships of that individual. The network can be used to measure social capital of actors and consequently the value that an individual develops from it. These concepts are often represented as a graph, where nodes are points and ties are lines.

The SNA uses a specific terminology to describe the components in graph:

- Actor or node: is the social linked together according to some relation.

- Edge: represent a binary social relation.
- Alters: are the actors to whom the ego is tied.
- Ego: is the focal actor of interest.

SNA studies are divided in ‘whole” and “ego” network approaches. Whole network studies the relationship patterns between actors within a defined or bounded group. On the other hand, an ego network studies an individual or several individuals, egos, and all their direct connections (alters).

Graph theory provides concepts and methods to analyse the structure of a network. The graph term for conceptualizing network is mathematical object and not a diagram (Harary, 1969).

Depending on the type of relationship, edges may be directed or undirected. In directed graphs, edges have direction and often referred to as arcs. Directed graphs are used to represent relational occurrences that have a sense of direction e.g., “is parent of”, “gives advice to”. The directed relations can be reciprocated. For example, in a certain group of people, every time someone gives advice to someone else, they receive advice from that person as well (Borgatti et al., 2013). In undirected graphs, the relations are not reciprocated. The edges do not have direction e.g. acquaintance ties.

In SNA, it is useful to distinguish the investigation in three levels: dyad, node, and the network. Dyad level is the fundamental unit of a network data collection and studies the relations between a pair of actors. In table 4.2, the notation O (n^2) indicates that the number of dyads in a network is of order n^2, where n is the number of nodes in the network.

At node level of analysis, most of network-level node properties are aggregations of dyad level measurements. The number of nodes in the network is of order n.

At the network level, the analysis is focused in terms of connected groups. The number of objects in this level of analysis is n^0, which means one. This means that a variable at this level of analysis will consist of a single number that characterizes a network as a whole (e.g. how densely connected it is).

Table 4.2: Types of network studies (Borgatti et al., 2013)

	Network variables as independent/explanatory	***Network variables as dependent/outcomes***
Dyad level O (n2)	Friendship between pairs of farmers to predict which pairs of farmers make the same decision about going organic	Similarity of interests (e.g., sky diving) to predict who becomes friends with each other
Node level O (n)	Centrality in organizational trust network to predict who is chosen for promotion	Extraversion to predict who becomes central in friendship network
Network level O (n0)	Shortness of paths in a group's communication network to predict group's ability to solve problems	Type of organizational culture (emphasizing either cooperation or competition) to predict structure of the trust network

4.9.1 Relations and Tie Strength

Relationship is a specific set of ties among a set of actors. In most cases, there are values associated with edges. These values represent the strength of the tie, frequency of interaction or a probability. This applies to both directed and undirected network data. When considering tie strength, ties are not exclusively measured in binary terms as one for presence and zero for absence, but rather as ordinal valued data in which the value of the tie indicates how strong the tie is. Some examples of relations are:

- Friendship (likes, friend-of)
- Professional relations (boss-of, same team member)
- Commercial relations (buys from, delivers to)

In a conventional social data set, each actor is described by attributes. Regularly there is interest in various actors' attributes and in multiple types of ties that connect actors in a network. Actors may be tied closely in one relational network, but be quite distant from one another in a different relational network. The positions of actors in multi-relational networks and the structure of networks composed of multiple relations are some of the most interesting areas of social network analysis (Hanneman and Riddle, 2005).

Methodologies for working with multi-relational data are different from those working with single relations. Many interesting areas of work such as network correlation and multi-dimensional scaling and clustering have been developed to work with multi-relational data (Hanneman and Riddle, 2005). For example in the work of Wu et al. (2010), the authors examined two dimensions of relationships between colleagues: professional vs. personal closeness. The intention is to demonstrate how professional and personal closeness can differ along strong and weak ties.

Mark Granovetter introduces the seminal concept of tie strength in his study "The Strength of Weak Ties" (Granovetter, 1973). He characterizes ties as strong and weak. Strong ties are the individuals whose social circles tightly overlap with a certain node, while weak ties are the other way around. Nevertheless, weak ties have an important role in a diversity of social activities, from spreading a disease to finding a job.

4.9.2 Clustering and Similarities

Algorithms that analysis ties strength in social networks are usually implemented through programs such as UCINET (Borgatti et al., 2002), Gephi (Bastian et al., 2009) or PAJEK[5]. While these software packages are capable of extracting complex clusters that are otherwise not apparent, they are highly dependent on the composition of the data matrices. Clustering is the notion that data can be classified grounded on their similarities. Data matrices can be gathered either directly from the data or inferred by sociologist. Multimodal matrices require complex inferences and implicit knowledge of the nature of the relations. Due the complexity in generating the necessary matrices, performing the decomposition and determining similarities in traditional social network analysis is both a temporally and computationally expensive task (Chung, 2006).

Another research area that makes extensive use of user similarity data is recommender systems. Collaborative filtering, one of the most common approaches, is based on similarity among users (Goldberg et al., 1992). Generally, user similarity is calculated based on input of users by rating a set of items in the system (Guy et al., 2010).

Opportunistic social matching is a class of systems that matches users independently of a user request. The matching is typically done based on shared interests and similarity found among individuals, where people are considered good matches with high affinity in common.

Differently from group recommender systems, social matching systems focus on suggest individuals instead of items.

4.9.3 Groups and Small Groups Analysis

According to Katz et al. (2004), group and small groups can be divided by two characteristics: structural feature of a network and determined category or boundary in set of people (e.g., a corporation, students). In the first, the study of group formation requires a set of criteria for classifying a given set of relations as a group. The authors state that although the definition of a clique (a fully connected or almost fully con-

[5] http://mrvar.fdv.uni-lj.si/pajek/

nected set of relations) for group formation is straightforward for binary choices, the issue of choosing threshold values becomes more complex for rankings. However choosing threshold values is also more complex when it is desired to relax the balance theory (Katz et al., 2004). The second characterisation takes the context of network analysis, typically used to compare patterns of intra-group versus inter-group category communication. The author also makes an analogy comparing group and small group in which small group would be collections with clearly defined boundaries and membership. This way, members are viewed as belonging to one particular group (belonging to a particular social class or category), not as belonging to multiple overlapping groups.

Additionally, they highlight the challenges in applying network perspective to the domain of small groups mainly arguing that SNA to date has been static in nature, typically consisting of a one-time snapshot of the network ties and lacking of longitudinal analysis.

In Sutcliffe (2005), based on the work of Arrow et al. (2000), makes a linkage between the theory of Small Groups as Complex Systems (SGACS) and CSCW for designing and modelling collaborative systems. He argues how technology can support social aspects of collaboration, suggesting that SGACS theory can contribute to CSCW design. The scope of the concept is limited to small groups, which usually work with less than twenty members (Velozo and Sagara, 2010). Complex systems represent the science that studies how dynamics of real systems emerge from the interaction of individuals and their environment. The systems term is not related exclusively to computer science, but also others sciences ranging from ecology to engineering.

The SGACS theory presents taxonomies of groups for intra-group modelling (local dynamics) and whole group modelling (global dynamics). The local dynamics provides an internal view of the group composed of actors, objectives, tasks, tools and communication channels. While global dynamics view to developing properties of whole groups such as social cohesion, motivation and effectiveness in achieving tasks. In this sense, the approach is focused social interaction in groupware systems.

When complex systems are utilized to adapt their behaviour in response to changes in the environment, they are classified as Complex Adaptive Systems (CAS).

4.10 Dynamic Social Network Analysis

Typically social networks analysis addresses the structure as a static graph, where it is either derived from a aggregation of data over the time or a snapshot of data at a particular point of time (Lin et al., 2009). Dynamic Social Network Analysis (DSNA) instead, studies temporal sequence, rate of communications, and the level of communication both locally and of the whole network. In other words, the nodes may appear and disappear and edges may tie or untie.

Hence, social network analysis of complex and unknown relationships is difficult and can only be performed off-line. Consequently, the data is collected over a period and then analysed using various graph theoretic techniques (Chung et al., 2006).

Research in social networks can employ two observational designs. Data can be collected at a single point of time (cross-sectional) or at multiple points in time (longitudinal). Social network tend to exhibit patterns of temporal evolution which result in characteristic spatial and social temporal signatures (Chung, 2006).

Most of the solutions for identifying communities in DSNA make explicit use of temporal changes where communities tend to evolve gradually over time and not spontaneously (Berger-Wolf and Saia, 2006; Lin et al., 2008; Tantipathananandh et al., 2007). Differently for CSCW and small groups, this approach is not suitable since the collaborative nature, primarily in pervasive environments can occur spontaneously,

Techniques often applied in static analysis remove all of the dynamic attributes of the relationships and often only find the dominant and explicit activity. In addition, Chung argues that incorporating a real-time approach to DSNA can complement traditional analysis techniques.

4.11 Conclusions

The attention in social networking analysis has been emergent over the last years. The key research issues involving SNA for computational analysis includes modelling of social networks, identification of techniques and if required handle its dynamic nature. The context information derived from social networks was not properly exploited in most context aware applications currently in use. Context aware applications underlying on the user similarity from different sources is motivating rather than depending exclusively on single individual information.

The SNS provide opportunity to access detailed contextual information that is hard to achieve in other ways to combine both online and offline data. Furthermore, this level of information is becoming increasingly available, but analytical and computational tools are still missing in terms of collective concerns.

Social software and SNS focused on social-technical systems while the social network theory on social ties. By exploiting the homophily principle (inclination of persons to associate with similar others) it is possible to provide an opportunity to connect with other individuals (Lazarsfeld and Merton, 1954).

The next chapter presents the CAFCA framework that was developed to fulfil all requirements discussed in the scenarios of Chapter 2. It extends the existing approaches for supporting synchronous collaboration in pervasive computing environments taken into account the CSCW, social computing and pervasive communities.

5 CAFCA: DESIGN AND IMPLEMENTATION

This chapter introduces the Context-Aware Framework for CSCW Applications (CAFCA). The proposed framework focuses mainly on social matching capabilities to session formation and application adaptability for collaborative activities. Furthermore, the framework architecture is based on loosely coupled components to support modularity. This modularity facilitates the integration in complex systems and the use of third party services where changes in one layer will not compromise the others. Each layer addresses a well-defined part and the details are described in the following subsections. This chapter describes the architecture and framework in a bottom-up fashion.

5.1 Introduction

The CAFCA main objective is to support informal and social communication by improving the means of interaction. The framework was primarily developed to be integrated with pervasive systems or to be used as a subcomponent for third party applications. In pervasive systems, CAFCA can perform social matching of users and groups to manage pro-active behaviour on integrated collaborative applications. As a third party tool, the framework can perform social matching techniques to suggest small group users for other platforms.

A number of requirements were identified to design CAFCA, partially to guide the development process of the architecture. The next section presents the architecture requirements to provide support for collecting, context modelling, organising and reasoning of context information.

5.2 Requirements Analysis

The framework analysis consisted on functional, non-functional requirements and domain requirement. The scenarios described in Chapter 2 were the starting point for gathering the functional and non-functional requirements. These requirements are a set of prerequisites working as guidelines for development of software systems.

The framework also took into consideration aspects from the analysis made in SOCIETIES project (Jennings et al., 2014) and is described in the next chapter. The project followed a user centric design approach employing field studies, surveys, scenarios and participatory methods. CAFCA framework concentrated on the enterprise group study and focused on collaborative work as a domain requirement.

The relevant user requirements concerning the framework from the project study were:

- Provide different communication channels for different tasks and groups.
- Users should be able to define the level of networking they are interested in, e.g. only interested in new contacts that are highly relevant to their own work focus.
- Support a variety of devices, including mobile.
- Devices should support as many services as possible.

Privacy requirements were a point of concern when it comes to allow exchanging resources and using others application as stated by the users. However, most of them stated wiliness to connect to people who have similar interests, since there exists a separation between personal and work use of the context information. In this way, user data is secured and only exposed to authorised services and users. Privacy is generally approached by the pervasive platform and the framework does not cover this requirement.

According to Roussaki (2011), scenarios and case studies have proven to be the most efficient ways to extract the technical and non-technical requirements within software-oriented projects. CAFCA adopted a four-stage process for scenario development and requirement analysis: scenario generation; scenario decomposition into scenario patterns; scenario evaluation, classification and filtering; analysis of scenarios that resulted in requirements specification (Lima et al., 2012).

The following subsections detail the functional and non-functional requirements for CAFCA framework.

5.2.1 Functional Requirements

The functional requirements describe what the system must do; how to react to particular inputs and how to behave in some given situations (Gava et al., 2012). In the framework case, the inputs are associated to the context information, while the outputs are to the adaptation behaviour in a collaborative application. Below, the essential functional requirements of the solution are presented:

- Support collaboration for co-located and non-collocated teams: The meetings can occur in a variety of places and situations. Therefore, it is fundamental that the solution facilitates the communication in both cases.
- Support for mobile and desktop devices: With the emergence of the smartphone and tablets, this implies to need support mobile and desktop devices. This takes into account the framework solution and collaborative applications. As a sub requirement, the framework should provide a GUI for the administrator and for collaborative applications; the users should be allowed to choose any application adequate for the device and situation.
- Find relevant users through use of context information: The approach should be capable to find participants based on the context information provided from the environment and the user social context.
- Support of social context: The increasing popularity of mobile devices makes users to be connected physically and virtually with others. In sum, the solution should concern diverse possible forms of relationships and interactions of the participants.
- Context enrichment: Be able to extend the context information collected. This implies the enrichment of terms semantically and consequently increasing/decreasing the similarity value of the terms.
- Support integration with existing collaboration tools: This translates to support a variety of cooperative applications. The framework should support synchronous and asynchronous tools for communication.
- Provide context awareness for collaborative tools: The system should have a pro-active behaviour to changes in context information. The framework must deal with the dynamic aspects of context information.
- Establishment of appropriate communication channel: if a user joins a group and needs to collaborate with audio communication support, the system should be able to choose the appropriate media for communication.
- Enable user to be notified about adaptations: When end-users match a certain criterion, the potential participant should receive a notification explaining the reason for the recommendation. Since users may be confused when facing some adaptation (López-Jaquero et al., 2008).

5.2.2 Non-Functional Requirements

While the functional requirements describe what the system should do, the non-functional requirements focus on what the system should be. The non-functional requirements for CAFCA framework are:

- Extensible and modifiable: The system should be flexible to be able to increase and extend functionalities. For example, the CMS and NLP enrichment used by the framework may change. Facilitate adding and changing services are important.
- Support scalability of integrated tools: the framework should support the integration with collaborative tools. Besides, these tools should run in parallel in order to provide flexibility to the user.
- Reasoning and adaptation behaviour performances time should be acceptable by the users.
- Reusability of the components: The framework should be designed aiming a variety of collaborative applications.

5.3 Framework Architecture

This section presents the context-aware framework for CSCW applications that was developed in order to enable context awareness in collaborative activities. In order to support separation of concerns the framework design is divided into three layers: modelling of context information, the context interpretation and the runtime actions. The structure of the CAFCA framework architecture is depicted in Figure 5.1. A box represents each component with its subcomponents inside. The arrows represent the flow of the data from the bottom (context sources) until the top (the devices).

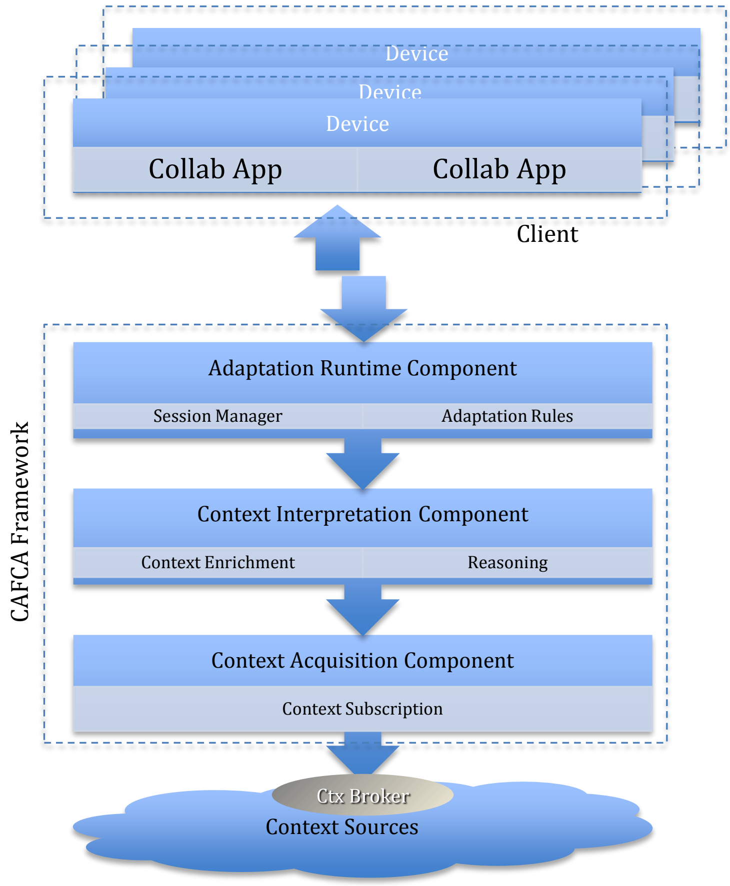

Figure 5.1: CAFCA Framework high-level architecture

The architecture was modelled in UML language in Enterprise Architect 8 and programmed in Java 6 (J2EE) language using Eclipse IDE. As a database solution, CAFCA uses a graph database and section 5.5 describes in details. As external tool for verifying the graph results, Neoclipse[33] and Gephi[34] were used for visualization .The framework was developed as an OSGi[35] bundle and a standalone application. The OSGi (Open Service Gateway initiative) is a set of specifications that describes a service platform and dynamic component for Java runtime services. Bundles can be locally or remotely installed, started, stopped, updated, and uninstalled without requiring a reboot of the system. These specifications reduce software complexity by providing a modular architecture, life-cycle management model and a service registry mechanism to facilitate the service-oriented platform integration.

5.4 Context Acquisition Component

The context acquisition is the first layer of the framework and responsible for collecting relevant context information and prepare it to the subsequent layer. As discussed in Chapter 2, there are some relevant context categories depending on the scenario. Some of the categories that might be used include geographic location, current availability (in the communication tools), current computational (mobile/desktop) resources and professional information.

5.4.1 Context Subscription

The context information is acquired via an external CB, which communicates with the context sources. The CB is a remote entity and responsible for gathering context information from the users and the environment through heterogeneous sources that can range from mobile sensors to SNS. The mobile devices, for example, can provide location, availability and resources. On the other hand, SNSs provide information related to user profile such as interests, professional position and expertise.

Technically, the CB provides the context consumers with a query interface for retrieving, adding, removing, and updating context data. The information retrieved from the CB is classified as long-term or short-term, depending on temporal characteristics. The long-term information is comprised by data that does not change often such as job position, areas of interests, skills. Alternatively, short-term context comprises data that changes frequently such as location and availability (e.g. user busy, away). This classification needs to be made by the administrator in charge of the framework, which choses the suitable context information for applied environment and the temporal characteristics.

---

[33] https://github.com/neo4j-contrib/neoclipse

[34] http://gephi.github.io/

[35] http://www.osgi.org/

Generally, pervasive systems have by default a specific component to manage context, i.e. CMS, allowing it to capture and retrieve context information. The CMS acts as an intermediate layer between platform/3P context-aware services and the sources of context information. Thus, the focus of the CAFCA framework can reside in the analysis and adaptation, abstracting the concern to deal with context sources providers.

All the context information can be checked automatically when an update event happens or periodically via awareness monitor of the framework. This procedure will depend on CMS capabilities and CAFCA framework will cover one or other. If the CMS connected to the framework provides subscription of event changes, then the framework will react in real-time to these notifications. In case the CMS does not provide such functionality, the framework will periodically fetch from the CB for information changes. The CB manages the interaction among components that gather contextual data and components or services that request the retrieval of context information from the context database. Figure 5.2 illustrates the scenario of CAFCA connected to an external CMS.

For instance, the CMS in SOCIETIES consists of two bundles, namely, Context Management and Location Management. The Location Management component exploits IBM's Presence Zone Server (PZS) functionality in order to determine the location of the user.

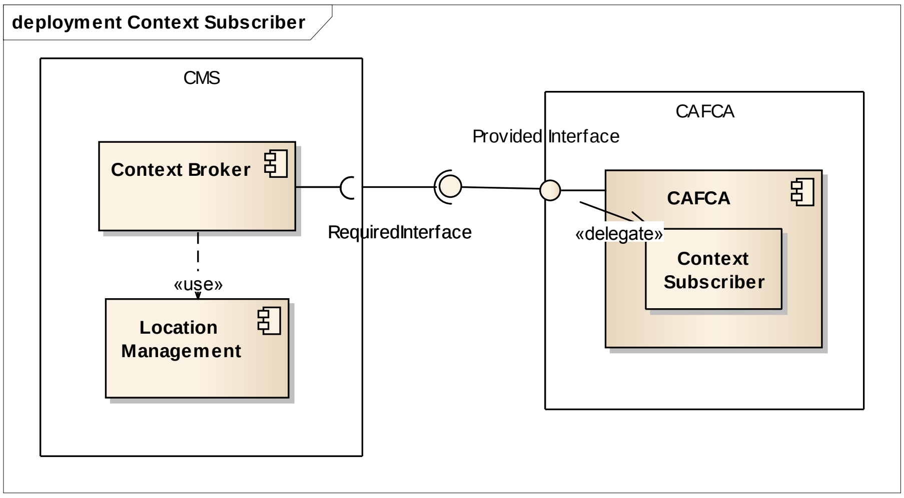

Figure 5.2: Context subscriber component diagram.

In relation to privacy issues, it is expected that user provides only the information that he thinks appropriate during the communication. On pervasive systems such as SOCIETIES, the architecture provides a set of privacy policies associated to the sensitive user data. When this information is retrieved for the first time, the platform requests the user if it grants permission to fetch and receive update about each specific information. The permission can also be granted to a specific community or a third party service. The user also has the opportunity to agree/disagree with the privacy policy rules before being inserted in a community. In this case, the members of the community and the third party services that belong to that community do not have access to the data or request in runtime. An important point is that personal information is only available in the user's client device. The platform only maintains information related to the communities in which the users are included.

5.5 Context Interpretation Component

The Context Interpretation component is responsible for modelling the knowledge and deriving the user data aiming to provide the adaptation layer with derivative knowledge for decision-making.

An important concern for context model is to sufficiently capture the targeted context characteristics, support efficient query, retrieval and maintenance (Beamon, 2011). Moreover, the author argues that the popular OWL and others description languages are useful when the context expected involves inference on relationships between concepts. The benefits found in using these conceptual representations is when the knowledge base is necessarily large, where consistency maintenance and inference prove to be unmanageable using other techniques. In this way, small knowledge base and those involving medium relationship inference justifies other representation and reasoning techniques, which are most appropriate.

In order to translate the data to the framework, this information is stored in a graph database. Although the context management systems have the purpose to store context information, the framework has to model it into graphs in order to perform the social network analysis. A graph database is a NoSQL (non-SQL or non-relational) that uses graph theory to store, map and query relationships. Differently from relational databases, NoSQL databases have a very efficient query time.

Each node in the graph designates an individual. The context information of each individual is used to assign edges on other nodes, representing relations existing among the users. These relations are based on the long and short-term context information.

The long-term information is associated as property of the person node as illustrated by figure 5.3. The short-term is associated to a new node that is created according to demand. Each new node of short-term information stores the timestamp and the new information using the person node as root. This behaviour is very similar to an activity stream, typically used by SNSs to maintain a list of recent activities performed by the users. In addition, the component keeps a history of short-term information. On the other hand, the long-term is not stored since the information change in most cases is not significant. This separation allows the layer to make decisions on how to link individuals from the same community and assign weights by the context information available.

To create the graphs it was chosen the Neo4j[36] graph database engine, which offers a graph-oriented model for data representation. The graph structures in Neo4j consist of nodes, relationships and properties. In the framework, the nodes represent per-

[36] http://www.neo4j.org/

sons and the short-term context information for each individual. The short-term scheme is illustrated in figure 5.3.

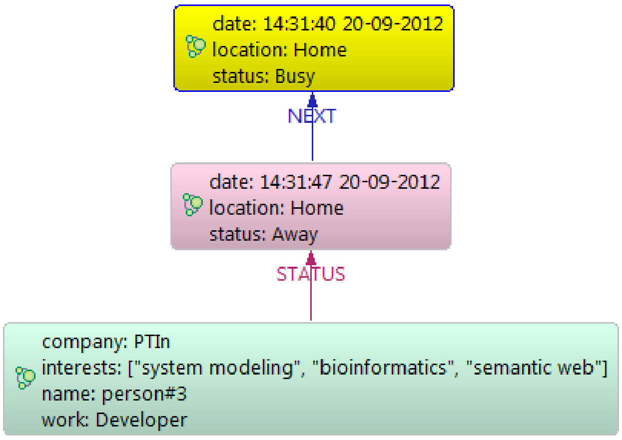

Figure 5.3: Short-term context information in the graph database

The relationships among persons are associated with the degree of similarity for given long-term information and are expressed as weights. On the other hand, the short-term information is represented with a timestamp and has LIFO (Last-In-First-Out) behaviour, where the last node is more recent update. The properties express the node information. Figure 5.4 demonstrates long-term context information.

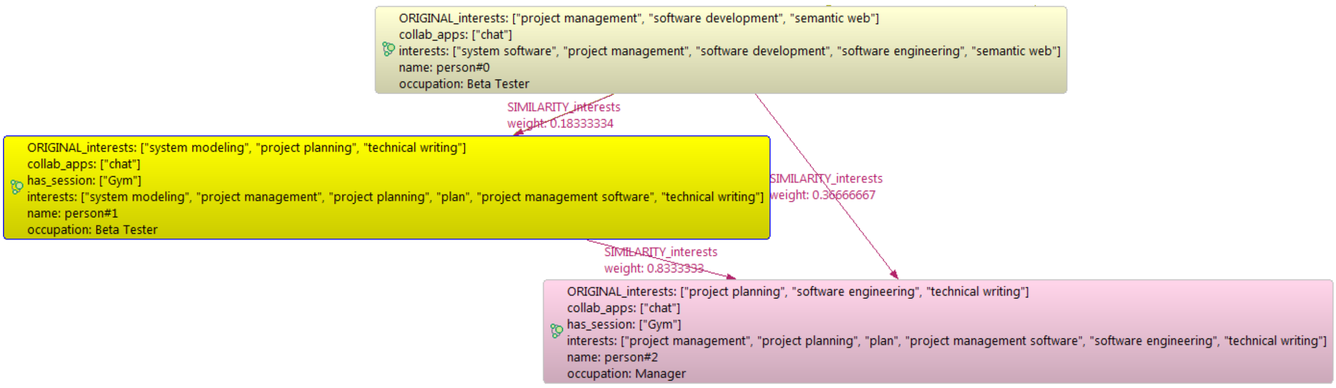

Figure 5.4: Long-term context information in the graph database

5.5.1 Context Enrichment

In order to extend the context information collected, the Context Interpretation component performs a semantic enrichment, depending on the data nature. The data can be expressed in either numeric or text values. The texts values can optionally be submitted to externals sources of Natural Language Processing (NLP) for semantic analyse, returning new keywords that are aggregated with the existing information. The enrichment addition allows the similarity metric to take into account the semantic value of the relevant terms that can be used later by the adaptation layer.

The enrichment of context information is an optional step in the framework analysis, although the enrichment plays a significant role to calculate the similarity of the individuals.

In the implementation for CAFCA evaluation, the framework integrated a NLP provided by Alchemy API[37]. The content to be analysed is submitted to the respective website API, which return the results in JSON (JavaScript Object Notation) or XML format. Nowadays, there are several possibilities of extracting semantic meta-data such as: concept, sentiment, text categorization, etc. For each keyword, the NLP is capable of returning a confidence level; a metric that indicates to what extent the extracted keyword could be a potentially relevant item. The framework adopted confidence levels greater than 90% for the enrichment. Other similar NLP services are possible to be integrated in the framework such as Synonym[38], Coginov[39] and MLTagger[40]. The connection with these external sources uses an API and requires an interface implementation by the developer in order to communicate with the services. Figure 5.5 illustrate and the Context Interpretation component and the NLP services.

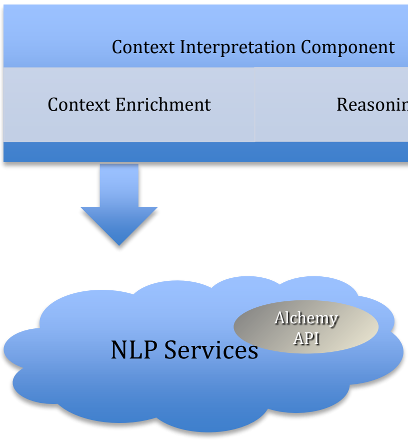

Figure 5.5: Context enrichment subcomponent integrated with an external NLP sources

5.5.2 Reasoning

Reasoning aims to generate new information of a relevance to the adaptation layer. With the context enrichment performed, it is possible to assign similarities among persons' node in the graph. This similarity enables to associate weights among person nodes. The similarity is calculated by dividing the matched context information between the nodes by the total context information of that particular node. This is applied to

[37] http://www.alchemyapi.com/

[38] http://www.synonym.com/

[39] http://www.coginov.com/

[40] http://parsie.deri.ie/EEYORETTT

both nodes and the similarity result is achieved summing both and dividing by two as represented in the formula below:

$$W = \frac{(matched\ ctx/total\ ctx\ node_1\ ) + (matched\ ctx/total\ ctx\ node_2\ )}{2}$$

Additionally, the Context Interpretation component can be prompted to report a threshold calculation in order to select relevant persons according to a similarity score.

This technique is called automatic threshold and it is regularly used in image processing for segmentation (Ridler and Calvard, 1978). The threshold value is calculated based on the similarity scores assigned for the long-term information. The values used by the algorithm are illustrated as properties of the edges among the individuals in figure 5.4. The automatic threshold equation is specified below.

$$Threshold = \frac{Average\ low\ similarity + Average\ high\ similarity}{2}$$

The technique is an iterative procedure based on the *isodata* algorithm from Ridler & Calvard. The method in the context of CAFCA divides the individual's similarity weight into low similarity and high similarity by taking an initial threshold. The averages of the similarity weights "*at or below*" and "*above*" are calculated and the threshold is incremented. The process is repeated until the threshold is larger than the composite average. Figure 5.6 presents the algorithm of the automatic threshold.

```
procedure GETAUTOTHRESHOLD(ArrayofElements)
    initialize: (initialThreshold, finalThreshold) ← (0)
    for all Element ∈ ArrayofElements do
        initialThreshold ← initialThreshold + Element
    end for
    initialThreshold ← initialThreshold + NumberofArrayofElements
    initialize: (done) ← (false)
    repeat
        initialize: (averageElements1, averageElements2, nElements1, nElements2) ← (0)
        for all Element ∈ ArrayofElements do
            if Element > initialThreshold then
                averageElements1 ← Element
                nElements1 + +
            else
                averageElements2 ← Element
                nElements2 + +
            end if
        end for
        averageElements1 ← averageElements1 + nElements1
        averageElements2 ← averageElements2 + nElements2
        finalThreshold ← (averageElements1 + averageElements2) / 2
        if initialThreshold = finalThreshold then
            done ← true
        else
            initialThreshold ← finalThreshold
        end if
    until done ≠ false            ▷ It is done if initialThreshold is equal to finalThreshold
    return finalThreshold
end procedure
```

Figure 5.6: Auto-threshold algorithm.

5.6 Adaptation Runtime Component

This component is responsible for managing the collaborative sessions and performing actions on the collaboration tools through use of rules. The session manager coordinates the collaborative sessions and a rule engine is in charge to provide adaptations to the applications integrated in the framework.

5.6.1 Session Manager

The session manager coordinates the collaborative sessions, session members, collaborative applications and floor control. The formed sessions are available in real-time for the administrator in a web page, as represented in figure 5.7.

- Session: the name of a session is given by the context information of the first rule. In the example illustrated below, the symbolic location of the participants is used as session name.
- Members: This field presents the members who actually accepted the invitation to join a session.

- Floor control: Indicates the commands available for that specific session.
- Language: Represents the common language used by the participants for the session. This feature was added to demonstrate the use of group context information.

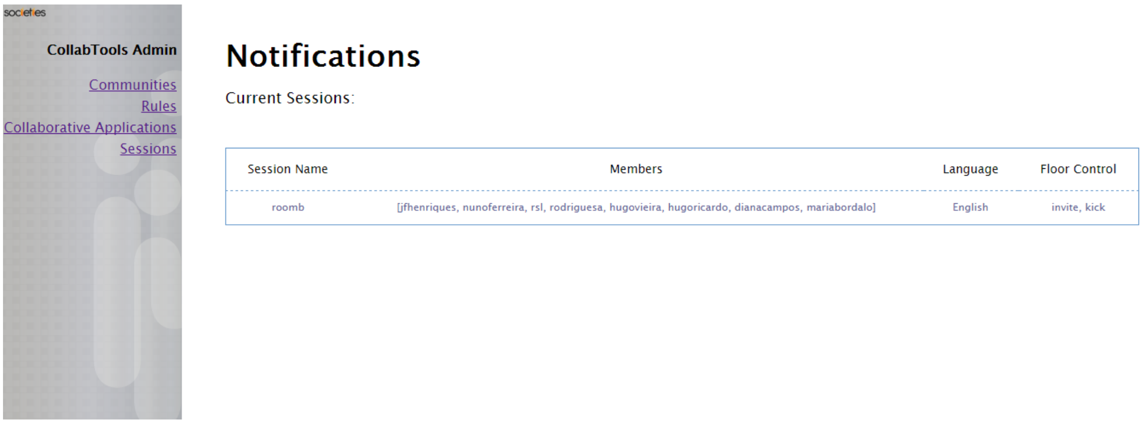

Figure 5.7: Screenshot of collaborative sessions on administrator's page.

In addition, the web front-end was built separately from the framework as bundle following a model–view–controller (MVC) approach. The web front was developed as browser solution allowing to be visualised in desktop and mobile phones.

Each collaborative session is stored in a graph structure and is composed by participating members; context information matched by the rules and invited members that not effectively accepted the invitation. Past sessions are also stored in nodes, which act exactly as the user short-term information approach.

The session's history follows the same short-term approach used for the individuals, storing new nodes with a timestamp with the matched context information, invitations and participants. Figure 5.8 exemplifies a session associated to a location and members in real-time.

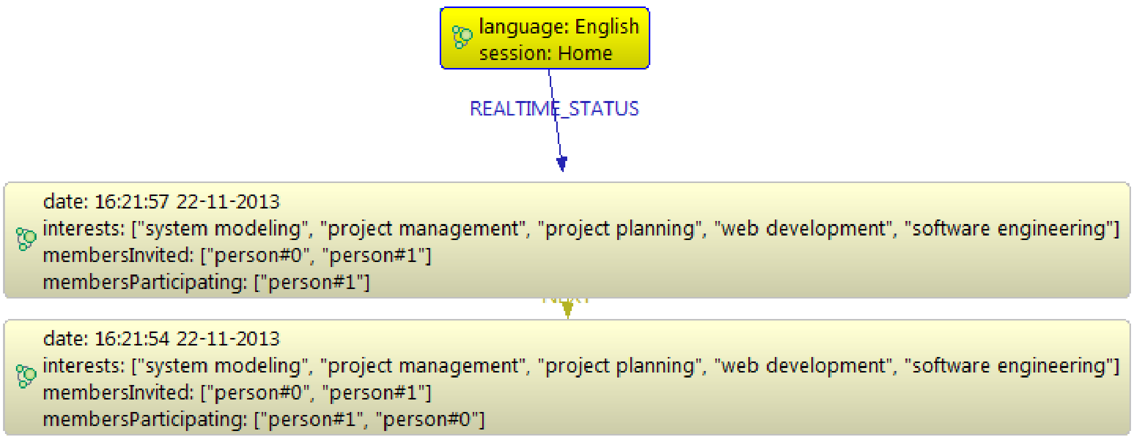

Figure 5.8: Long-term context information in the graph database

Following the 3C Cooperation Model concept described in Chapter 4, it is present the relation among communication, coordination and cooperation model to highlight the features of this layer:

- Communication: The session manager is in charge to choose which communication channel is common to all parties. If a person is idle for a long time then the user can be removed from the session or if busy, not invited to join the session. Furthermore, as a communication awareness feature, the user is notified with context information of which data matched with other users. One feature added to the experiments performed in the evaluation chapter was an automatic choice of a language for sessions. The language chosen is based on the language information provided by the group context information. Thus depending on the majority of the users, the invitation is made on that specific language.
- Coordination: To support coordination for the team members, the session manager provided a mechanism to identify probable members for a session. In a real case scenario, it is possible to find experts for a specific task making them aware of how experienced the candidates are. Furthermore, the session manager is supplied with floor control, which can be managed by the framework administrator. The floor control is responsible for controlling the role of the session members. A member role defines a set of operations that can be performed by its collaborators, for instance to speak and to write. The floor control strategy is that the users have the same permissions granted in all integrated applications. The administrator has full control to the floor and can add or remove a role to a session member
- Cooperation: Cooperation is related to workspace in which people are able to collaborate and share information related to their collaborative work (Steinmacher et al., 2012). For usual CSCW activities, this translates in collaborative document editing, use of annotation to present changes and avoid conflicts during cooperative activities. In pervasive scenarios, these cooperation reflects on the use of smart environment and intelligence sensors such as in multi displays environments (Kulyk, 2010) and immersive environments (Papadopoulou et al., 2014). The framework supports synchronous and asynchronous cooperation. In asynchronous work, the framework can provide the social matching capabilities e.g. calendar sharing. While for synchronous communication, it can offer the social matching and the adaptation rules e.g. chat, videoconference.

5.6.2 Integration of Collaborative Applications

To ensure adaptation coordination across multiple devices, the framework provides an approach for integration of applications running on various platforms. Support a variety of devices is important to provide flexibility to the users. The framework supports this integration with existing third party collaborative tools or implemented by developers a posteriori. This approach is based on the work of Gomes et al. (2011) that propose a loosely-coupled integration environment enabling collaborative applications to interact with each other without losing their autonomy. In order to use the collaborative tools with the framework, the user can integrate the server application or the application itself with the adaptation layer. The adaptation layer is in charge of receiving and performing actions on one or more collaborative applications at same time. In this manner, there is no direct access to the framework internal subcomponents, avoiding the coupling among applications.

The applications are integrated depending on its nature. Synchronous applications require implementing an abstract class, which informs how to join and leave a conference and the events related to the user such as join, leave and availability status. As a result, the adaptation layer can execute actions directly in the application server. Some examples of server applications are Openfire[41] for chat and Asterisk[42] for VoIP.

For asynchronous applications, it requires to register for bundles events to receive updates about social matches calculated by the rule engine. Differently from synchronous approach, the applications use the suggestions from the framework, but do not receive any action to perform. Some examples in this area are calendar sharing services, CoBrowser and car-pooling systems.

The applications integrated in the framework are not responsible for managing the collaborative sessions and are not aware of the behaviour of others integrated applications. This way, the framework deals with the orchestration of each tool without compromising them.

5.6.3 Rule Engine

The adaptation in the collaborative tools occurs with the use of a rule engine. A rule engine was designed and developed especially for the framework. The engine allows a real-time evaluation of the rules in a dynamic social network.

The rules engine focus on two characteristics for analysis: co-located teams and non-co-located teams.

- Co-located teams/Priority mode: In this mode, the engine takes into consideration the priority of the rules. The rules are analysed one by one in priority ranking. This mode is indicated when the intention is to achieve a precise result (E.g. using location).
- Non co-located teams/Relevance: This mode uses social network analysis (SNA) techniques. The relevance mode considers all the individuals present in the graph database. This technique uses weights in each rule to achieve the results.

[41] http://www.igniterealtime.org/projects/openfire/

[42] http://www.asterisk.org/

This mode is indicated when the intention is relaxed results. E.g. using location and other personal attributes.

In terms of performance, Neo4j in the first access uses the disk storage to search for relationships and nodes. The following accesses will be presented in memory cache. Neo4j also supports indexing of the information available in the nodes and relationships for fast searches. The index is supported by Apache Lucene[43], an external open source library for information retrieval. Lucene is known for its capabilities for indexing and searching from within large collections of documents.

In priority mode, the framework has a time complexity of O (|T |.|D|) for each rule, where T is set of terms and D is the set of all documents. This concerns the time complexity of the Lucene search. The documents refer to the nodes and relationships properties found in long-term and short-term information. For evaluate long-term values, the automatic threshold is applied as described in the context interpretation component.

The time complexity for relevance mode includes the complexity of the indexes search plus the complexity of a matrix aggregation. For each context information is constructed an adjacency matrix. The objective is to put the data into an aggregated distance matrix. To be able to aggregate the multivariate data in matrices, it is necessary to normalise each similarity matrix separately into values ranging from zero until one. Data value as long-term information does not require any conversion, since the data is already calculated in the precise form, i.e. interests. For short-term information, the binary data needs to be normalised using a Simple Matching Coefficient (Teknomo, 2014). This technique is used for example with location. The formula below represents the concept where p is the number of positive variables (1) and q the negatives (0). The result is divided by the number of variables.

$$S_{ij} = \frac{p+q}{t}$$

Moreover, each matrix has a weight associated with the rule. This weight is multiplied with each element of the matrix. Subsequently, the time complexity for matrices addition is O (n^2) for each rule, where n is the number of elements to be added. Since any number of matrices required for addition is two. Again, the amount of time taken will depend on the number of individuals and rules required to evaluate.

The similarity result from the aggregated normalised matrices is the Gower's General Similarity Coefficient (Gower, 1971):

[43] http://lucene.apache.org/

$$S_{ij} = \frac{\sum_{k=1}^{n} W_{ij}.S_{ij}}{\sum_{k=1}^{n} W_{ij}}$$

The index K represents the number of rules; W represents the weight and S the similarity of that context information. The denominator specifies to divide by the sum of the variables weight. As described in section 5.5, the threshold is calculated automatically based on all the individuals and used with the similarity result of the aggregated matrix.

Consequently, the time complexity of priority mode analysis is smaller than the relevance mode, since the priority mode does not require calculation for all nodes for each rule. Two factors influencing the performance of the algorithm: number of individuals and number of context information. Thus, the time complexity is acceptable for both modes and for the scenarios described in Chapter 2. There are libraries developed to improve speed and performance for matrices calculations such as Efficient Java Matrix Library (EJML)[44] and ojAlgo[45]. However, those were tested and not presented advantage in terms of performance related to the standard math libraries available in Java 6. Next chapter approaches the evaluation and demonstrate in details the complexity involved.

The rule engine was developed for the administrator to be able to create, modify and delete rules without any previous programming knowledge. Figure 5.9 exemplifies two rules in the web front-end: location and interests.

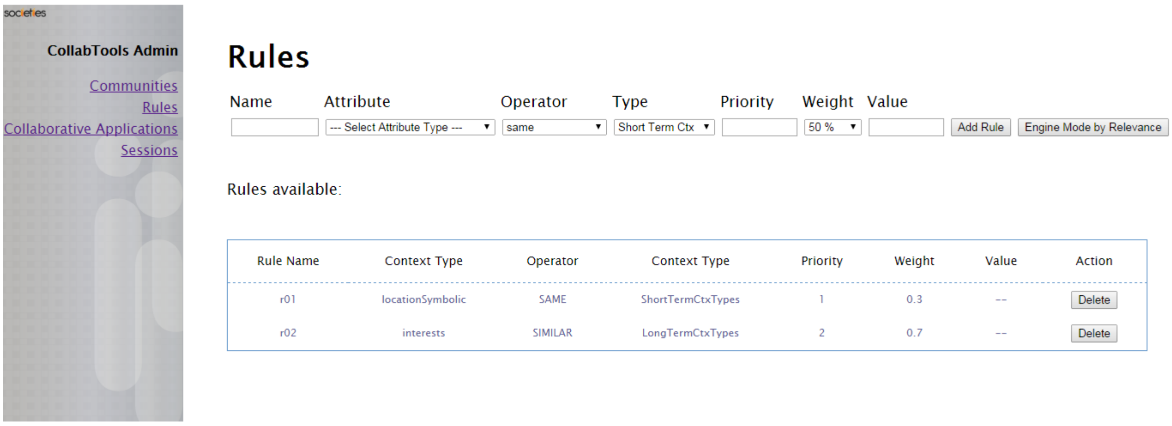

Figure 5.9: Screenshot of rules on the administrator page.

The administrator needs indicate the fields for the rules that are presented below:

[44] http://ejml.org/

[45] http://ojalgo.org/

- Name: Expresses the name of the rule. The name is only used as a reference.
- Attribute: Context attribute to analyse.
- Operator: Rule operators compare the context attributes of the members with the value provided.
- Type: Long-term or short-term context type.
- Priority: Priority of the rule in case of the rule engine is set to priority mode.
- Weight: Represents the weight of the rule. Used with the rule engine set in relevance mode.
- Value: Used within the operator to compare a value, which can be textual or numerical.

As illustrated in figure 5.9, an engine mode button has the function to switch between relevance and priority modes. The administrator can change this behaviour anytime. In the next subsection, the adaptation rules are explained in details.

5.6.4 Adaptation Rules

The adaptation rules are in charge to perform actions on the applications integrated in the framework via stipulated conditions. The rules are expressed by IF-THEN clauses and are defined as if "*condition*" then "*action*", e.g.: "*If at least two individuals are in the same location and work at the same department and have the similar interests THEN invite to a session*".

Some operators are available with text and numeric values, while others only with numeric values (see Appendix C). The operators available for the rules are detailed below:

- SAME: Represents the same context information. This operator is normally used when no information is known a priori. E.g. a location.
- SIMILAR: Represents the relational operator for values higher than the auto-threshold.
- EQUAL: Represents the relational equality operator.
- NOT_EQUAL: Represents the relational inequality operator.
- GREATER: Analyse if the value is greater to a given value. Used only with numeric values.
- GREATER_OR_EQUAL: Analyse if the value is greater or equal to a given value. Used only with numeric values.
- LESS: Analyse if the value is below to a given value. Used only with numeric values.

- LESS_OR_EQUAL: Analyse if the value is below or equal to a given value. Used only with numeric values.

Apart from the operators commonly found in rule engines, the SAME and SIMILAR operators are also introduced. The SAME operator is very useful in situations where context information is not known a priori, but a relational equal behaviour is desirable. The adaptations rules can also check the availability status of the user to decide which communication channel best suited for collaboration or which is common to all parties. For example: "*if a person is idle for a long time then the user can be removed from the session*", "*if a person is busy, not invited the user to join the session*".

For each stipulated rule it is possible to assign a weight or/and a priority, which will be verified by the rule engine. The weight is used to assign a score to each rule on a percentage format. The rules and weights are formally represented below:

$$P = W_1 . Rule_1 + W_2 . Rule_2 + \cdots W_n . Rule_n$$

E.g., if the first rule is stipulated in 30% for location and 70% for interests then a user may be invited for a session, even if it is not present in the same location.

In priority format, the significance is sorted according to the rules. The rules and priorities are formally represented as:

$$P = Rule_1(P_i) + \ Rule_2\ (P_{i+1}) + \cdots\ Rule_n(P_{i+n})$$

Following the previous example, if location is the first rule and the interests is the second rule, the engine should first check who is at the same location and then verify the interests of the users filtered from the first rule results.

5.7 Comparison with Related Work

This section compares the framework proposed with existing research on platforms and frameworks for CSCW that take into consideration social computing in pervasive environments. Although the effects from the different research fields influenced the study, the results share concerns in social computing, pervasive computing and CSCW.

In Divitini et al. (2004), Farshchian and Divitini (2010) the authors propose a service platform called UbiCollab. UbiCollab[46] is an experimental platform for the development of ubiquitous applications to support cooperation among distributed users. UbiCollab also allow the integration of the specifically implemented applications on the

[46] http://www.ubicollab.org/

top of platform and support different levels of channel communication. Recently the authors enhanced the platform to support other features such as presence awareness and peer-to-peer communication.

In Nino et al. (2009), the authors present MaPS (Matching People to Share). MaPS is a framework designed to offer a model to search people in ubiquitous environment with the goal of collaborating. The proposal uses context information and user profiles in order to search for users and communication channels selection. However, the framework uses context information but the solution does not mention the use of pervasive environments sensors besides the user context information.

In Hussain (2010) the author extends CONTact (Context-based Adaptation Collaboration Technology) framework. CONTact provides runtime system allowing context-based adaptive tools/applications to register (by specifying supported adaptations) to it, and use context information to reconfigure their behaviour according to changing context (Haake et al., 2010). The modification applied by the author provides generation of explanations based on context information to help users understand the adaptation behaviour of the framework.

In CASMAS (Community-Aware Situated Multi-Agent Systems), a conceptual model to support coordinated activities of community members is introduced (Cabitza et al., 2006; Locatelli et al., 2008). The model allows the design of systems focusing more on collaborative tasks. This solution approaches multi-agents to model and develop environments to support group collaboration among persons in ubiquitous computing.

For proper understanding the contributions and limitations of each proposal, table 5.1 presents a comparison table illustrating the most sensitive functional requirements.

Table 5.1: Comparison table of the frameworks

	UbiCollab	MaPS	CONTact	CASMAS	CAFCA
Social Matching		x	x	x	x
Real-time analysis of DSN				x	x
Integration with existing Collaborative applications			x		x
Collocated and non-collocated teams support	x			x	x
Context and Social awareness			x	x	x
Different communication channels selection	x	x	x		x
Provide context awareness for collaborative tools			x		x
Context enrichment		x			x
Support a variety of devices	x			x	x
Pro-active behaviour	x		x	x	x

Most of the models and frameworks presented here offer support for collaboration applications via APIs for specific application development (UbiCollab, CONTact and CASMAS) and multiple-computer devices (UbiCollab, CASMAS).

The main drawback from the solutions thought, is the lack of synergy among pervasive computing, CSCW and social computing. By correlating context information from user (e.g. user's profile, interests) and environment (location and environmental characteristics) with social information (relevance, reliability), it is possible to define new ways of communication and new physical social interactions among users.

Differently from the solutions proposed, mobile social networking has recently increasing attention of the benefit of context and social information (Conti et al., 2012). Mobile social networking studies opportunistic networks for the reason of the dynamic and unpredictable nature of users' encounters in open deployment environments, currently referred to opportunistic computing (Boldrini et al., 2010). The main challenges opened in this field are under the perspective of both adopted protocols and design/deployment choices (Bellavista et al., 2013). The solutions take into consideration relevant aspects such as users and places, the social relationships (e.g. event participation or user's current activities in a place). For example, recently in CAMEO framework (Arnaboldi et al., 2014) proposal, the authors use context-awareness and social-awareness for mobile devices to support the development of real-time mobile social network applications. The aim is in provide opportunistic communications among users' mobile devices where Internet is not always present. Consequently, the interest of the study is mostly focused on networking protocols issues and resource sharing and not in CSCW/CSCL domains.

5.8 Design for Communities Recommendations

The CAFCA framework was developed during the period of the SOCIETIES project. However, CAFCA was separately investigated and implemented as a standalone framework aiming at CSCW. Further, in the middle of the SOCIETIES platform development, the project decided to adopt core features available from CAFCA for a platform component entitled Community Recommendation Manager.

The Community Recommendation Manager (CRM) was responsible for providing a list of communities' suggestions, based on filters provided by the user or a third party applications. The component used the CIS Management (see figure 2.5) to retrieve the remote communities, which the user still not participates, presenting the results according to the selected filters. Figure 5.10 illustrates the CRM component diagram and presents its internal structures.

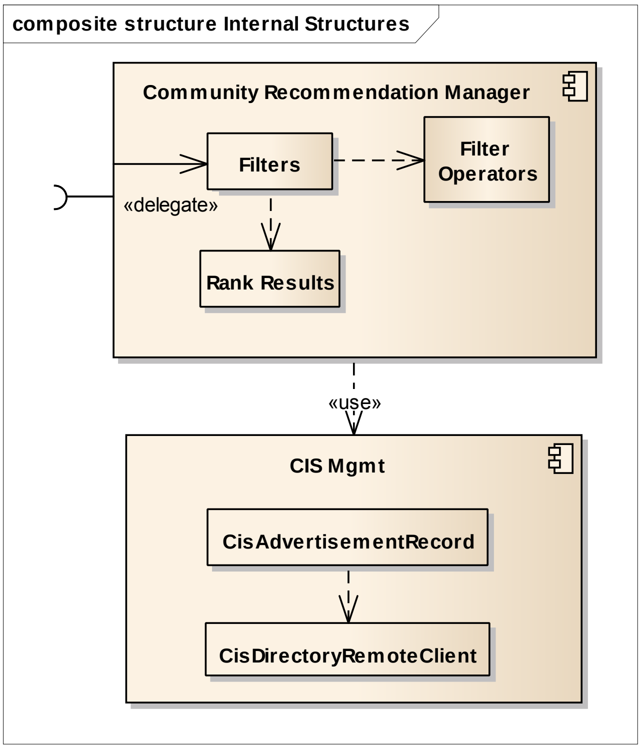

Figure 5.10: Community Recommendation Manager Component.

The CRM provide, via an API, searches for communities by defining a set of primary or secondary filter. The primary filter allows querying accurate results, consequently only results that exactly match the filter will be presented. On the other hand, the secondary filter allows it to submit queries that can provide accurate or close results (see Appendix C).

The communities have by default a number of membership criteria associated, which expose the purpose of their creation. The membership criteria present their information using context information available in the CMS component. The context information can be either numeric or string values and are employed subsequently for comparisons by the CRM.

Each filter uses a set of operators and is similar to the approach used in CAFCA rule engine. The filters compare the context information in the membership criteria with the data provided by a user or a service. The defined filter operators are equal, not equal, greater, greater or equal, less, less or equal and in. For instance, a filter can be defined as "Age GREATER or EQUAL to 25". While a second filter may be defined as: "Location IN Aveiro". As a result, the outcomes would present communities located in Aveiro or nearby with people above or equal to 25 years. In addition, the CRM results are shown ranked from most to least relevant, based on the filters defined for search.

This subcomponent was part of the Intelligence Community Orchestration, which was in charge to manage the pervasive communities and is detailed in Papadopoulou (2014).

5.9 Conclusions

This chapter has introduced the design and architecture of the CAFCA framework, covering aspects of context-awareness, social context and adaptation. The approach aims to facilitate the interaction of pervasive user groups in collaboration environments by proactive behaviour in the applications for the participants. In addition, the proposed framework architecture facilitates the integration of existing collaborative applications enhancing them with context-awareness features with minimum effort.

Another important contribution is the proposed representation for modelling social and context information considering a social network. This approach aims to benefits collaborative work considering social aspects and mobility of the users. Additionally, the framework was implemented to run as a standalone or third party service OSGi bundle[47].

To prove the applicability of CAFCA framework, the subsequent chapter presents the evaluation and a user trial. The framework evaluated empirical users in a real environment based on the motivating scenarios presented in Chapter 2.

[47] https://github.com/societies/SOCIETIES-SCE-Services/

6 Testing and Evaluation

This chapter presents the test and evaluation of the proposed framework. The first analysis was conducted using an enterprise scenario as part of the SOCIETIES project to cover user requirements for a pervasive community platform and third party applications involved. The second analysis focused exclusively on the CAFCA framework operation and intrinsic user evaluation. In addition to the CAFCA analysis, it was conducted a trial with eleven participants in a conference use-case scenario, which enabled an assessment of the framework behaviour in real-world conditions.

6.1 Introduction

In order to test and evaluate CAFCA, this chapter presents first the requirements collected by the SOCIETIES project, which involved the platform and third party applications design and creation. This study conducted a comprehensive investigation in an enterprise scenario (Doolin et al., 2014), from which it was possible to characterize the necessary functionalities for CAFCA evaluation.

For the second part, it was carried out an evaluation that demonstrates the capabilities of CAFCA integrated with a collaborative application. This integration considers the interoperability of context and social networks mechanisms to verify the impact in terms of functionality and efficiency of the framework.

The primary emphasis of the framework targeted the enterprise users group and, as suggested by Muller et al. (2012), this type of communities and groups presents potential advantages. Firstly, business activities provide shared context in addition to the context of the community, which can contribute to a level of trust and common ground. Secondly, enterprise communities tend to be business-focused, leading to different content and perhaps styles of discussion. Thirdly, company environments, which require authenticated access and use of real names, eliminate anonymity and provide greater transparency.

In this way, the evaluation of CAFCA followed a twofold analysis. The first objective was the qualitative data collection and analysis using pre and post questionnaires (see Appendix A and B). Second objective was to collate information about the users via analysis of log data and databases for the quantitative logging. This data collection and analysis was intended to examine the influence of CAFCA behaviour.

6.2 Enterprise Study Case

As a part of user trials conducted by SOCIETIES project, three specific scenarios were developed to cover different standpoints of pervasive communities (Jennings, 2011). The target scenarios defined for user groups were disaster management, student and enterprise. In this manner, each scenario was created aiming to cover all the functionalities of the SOCIETIES platform as much as possible and features exclusively in third party services.

Many requirements investigated in this analysis stage were included in the platform development. However, specific requirements identified in only one of the user groups, as collaborative work and small groups, were not considered for inclusion in the platform. Although each developed scenario has been analysed by the same methodology, the platform tried to tackle common requirements among the three user groups. The CAFCA framework benefited from the enterprise group study, which concerned a conference scenario (Roddy, 2014; Roussaki, 2011).

The specific research methodologies applied by the project in each user group were ethnographic observations, questionnaire and participatory design sessions. Qualitative methods were used for ethnographic observation and participatory design sessions, while quantitative methods for the questionnaire to confirm user behaviour and trends.

The enterprise user group was composed of Intel Corporation employees that included researchers, managers, support staff and office staff. These employees were chosen because they recently attended conferences or may do so in the future. They are all part of multiple communities within the larger enterprise organisation located in 48 countries worldwide and with over 79,000 employees across several geographical locations.

In ethnographic observations, the objective is to gather information about the user, his attitudes, behaviour and activities. The main findings for enterprise user group were:

- Users need easy access to the agenda information for each physical location where sessions are taking place.

- Users would like to be able to access information from other sessions during the event and some users intend to share information with colleagues who are not in the same session or at the conference.
- Users are only interested in new contacts that are highly relevant to their own work focus.
- Users have a variety of devices and techniques to capture notes and details about the conference. Interoperability and the use of standards is an abstracted requirement.

The participatory design sessions were arranged both physical and online with the enterprise user group. The methods applied in the sessions were influenced by CSCW philosophy and experiences in living labs. In the case of the enterprise group, the storyboard scenario comprised in a conference, where each participant was able to install an application in his/her smartphone or tablet, operating as a welcome pack registration kit. The application automatically registers the participants in the conference sessions chosen by the users. Additionally the application indicates the indoor location of each session, workshop and coffee break following the participant agenda (Jennings et al., 2014).

Last, a questionnaire was developed to gather statistical information about users' demographics and behaviours in relation to social media usage, community group communications, technology usage and domain specific issues.

6.2.1 Enterprise Group Questionnaire

In a sample of 20 participants that were interviewed, 86% were male and 14% were female. Typically, these users spend working days in the office and attend conferences occasionally to learn about new technologies or research areas, disseminate their own work to other colleagues or third parties and for professional networking to meet other attendees sharing common interests.

In order to analyse devices, sensors and social media the participants were asked to state the activities for which they are using each particular device. Table 6.1 shows how each particular usage activity is distributed over different types of devices.

Table 6.1: Participants that use specific devices to perform the listed activities

	Mobile Phone	PDA/ Smartphone	Laptop	Desktop PC
VoIP	20%	45%	70%	20%
Web Surfing	30%	75%	90%	55%
E-mail	30%	90%	95%	45%
IM/Chat	5%	25%	80%	30%
SMS/MMS	40%	80%	20%	20%
SNS	5%	40%	60%	35%
Blogging	5%	10%	25%	15%

Next, participants were asked to identify different types of sensors (ID sensors, Biosensors, etc.) that they used on a daily basis. Only 10% of participants do not use any features based on sensors. The other 90% of participants use at least one of the sensors on a daily basis. Location sensors are used the most and were identified by 75% of participants. Figure 6.1 shows the use of the participating sensors on a daily basis.

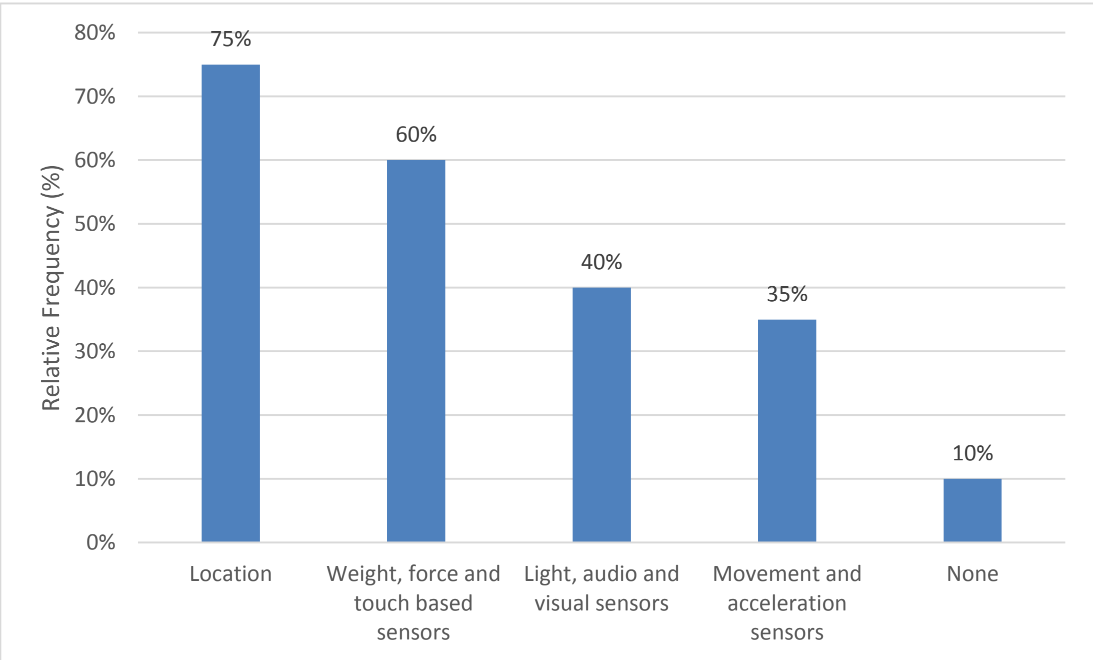

Figure 6.1: Daily use of features and sensors integrated in devices.

The participants were then asked to specify what social media they use. Figure 6.2 illustrates which social media the enterprise group uses. The most popular social media indicated by the group are IM tools, Facebook, and LinkedIn. Although the social networks have a large part of the interests of the participants, instant message tools still have a lot of attention in this type of user group.

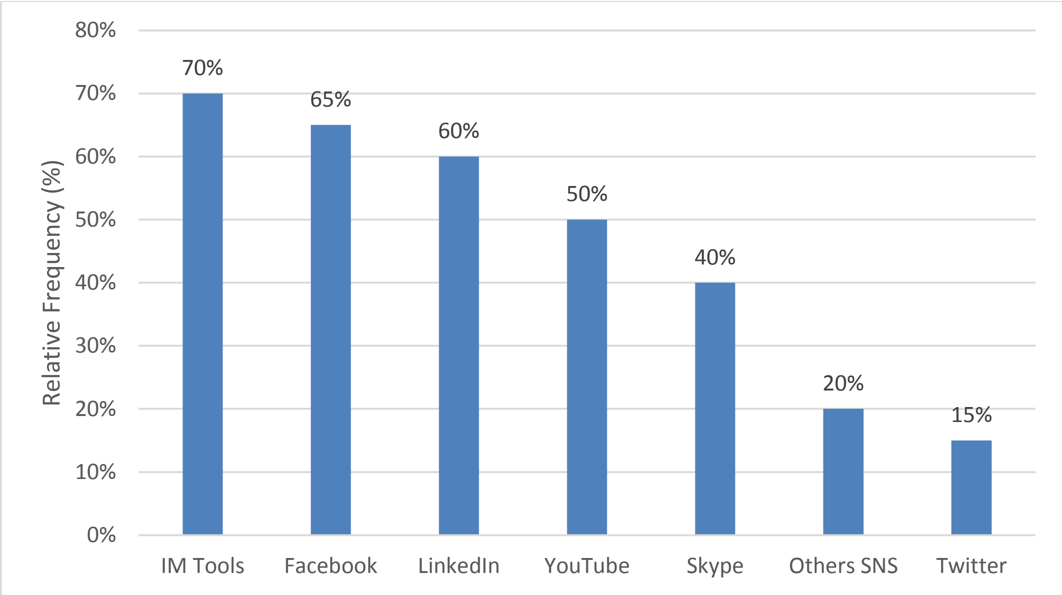

Figure 6.2: Social media usage.

Next, the participants were asked how often they use social media compared to the following social media categories: social networking sites, instant messaging and blogging. The frequency of social media usage regarding social media categories is shown in figure 6.3.

Again, IM is used most often with 80% of participants using it most of the day. Social networking sites come in second place with 44% of participants using them either daily or most of the day. On the other hand, blogging is the least popular social media category. Social networks integration proved to be very important again in the question with most participants picking Facebook and LinkedIn. Blogging in the other hand proved to be irrelevant for the enterprise group.

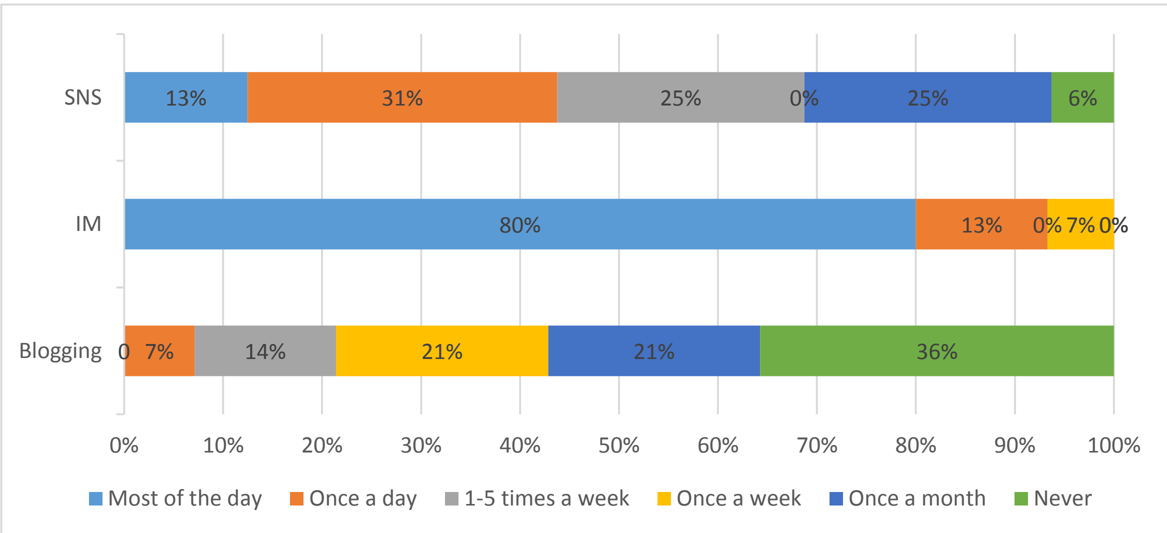

Figure 6.3: Usage of Social Networking Sites, Blogging sites and Instant Messaging applications

As demonstrated in figure 6.4 activities such as social media are mostly used are for keeping in touch with friends (60%), emailing and messaging (50%), chat and instant messaging (50%), as well as sharing of videos and photos (50%). Moderately popular activities include professional networking (40%), sharing information (35%), research (25%), commenting on photos and links (25%), organising events (20%) and groups (20%). Moderately business activities demonstrated to be relevant in a certain degree such as professional networking and sharing information. IM tools again, are highly relevance for the participants.

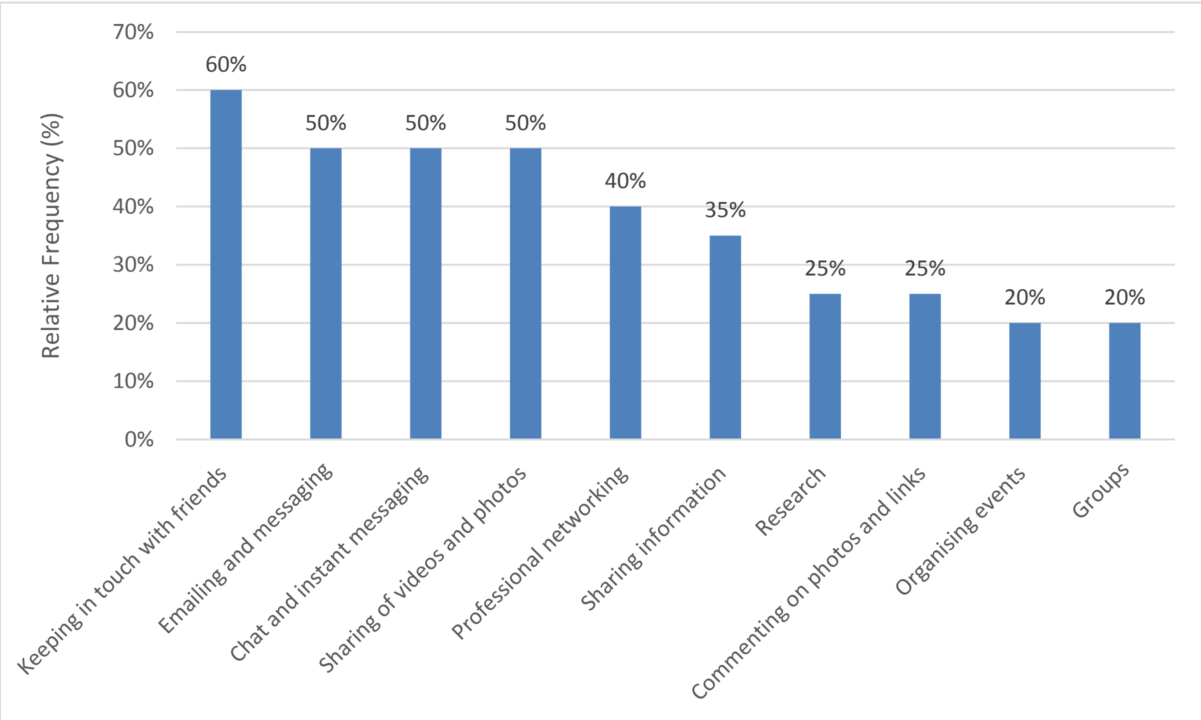

Figure 6.4: Type of activities that social media are used

Regarding community involvement, the survey asked participants about the most significant criterion for joining a new group. The approach followed a scale, where the most important criteria are ranked by one and least important by four. The question was answered by half of the participants (10 out of 20) and presented in figure 6.5.

The most important criterion for joining a new group is members to be friends. The second criterion is other group members being friends of friends. The third criterion is people having the same interests. The least important criterion is spending significant time in close proximity.

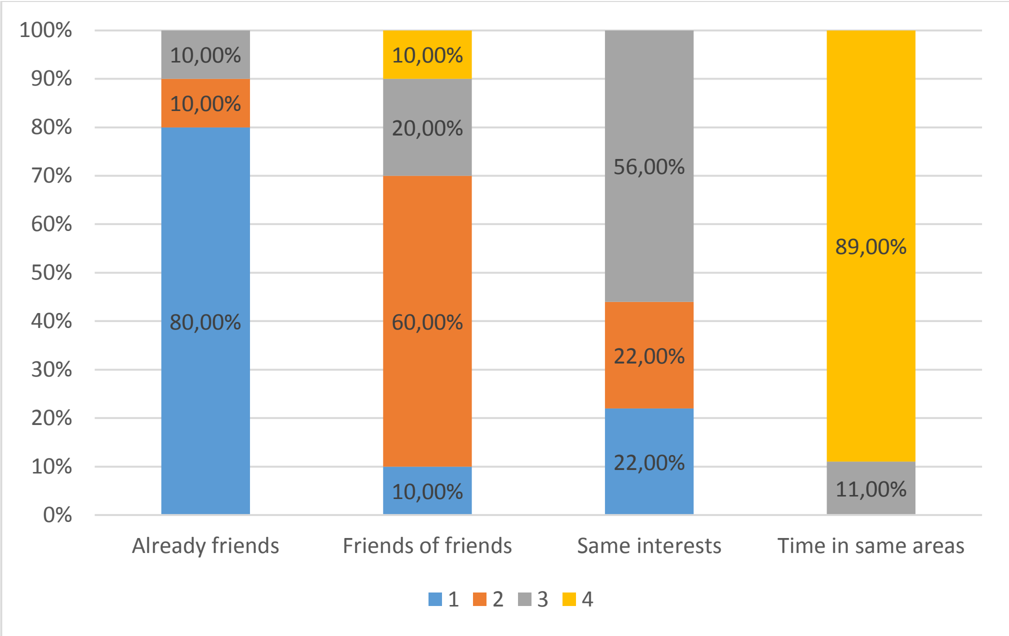

Figure 6.5: Significant criteria in order to join a new group

Finally, the participants were asked to rank their objectives when they attend conferences. The results are visualized on figure 6.6, ranking with one the most important and by four the least important.

Education is the most important goal for enterprise users with 13 participants out of 20 answering the question with 69%. Networking is the second most important goal, with 14 participants answering this question and with 50%. Dissemination and showcase are the least important goal with 11 participants answering this question with 55%.

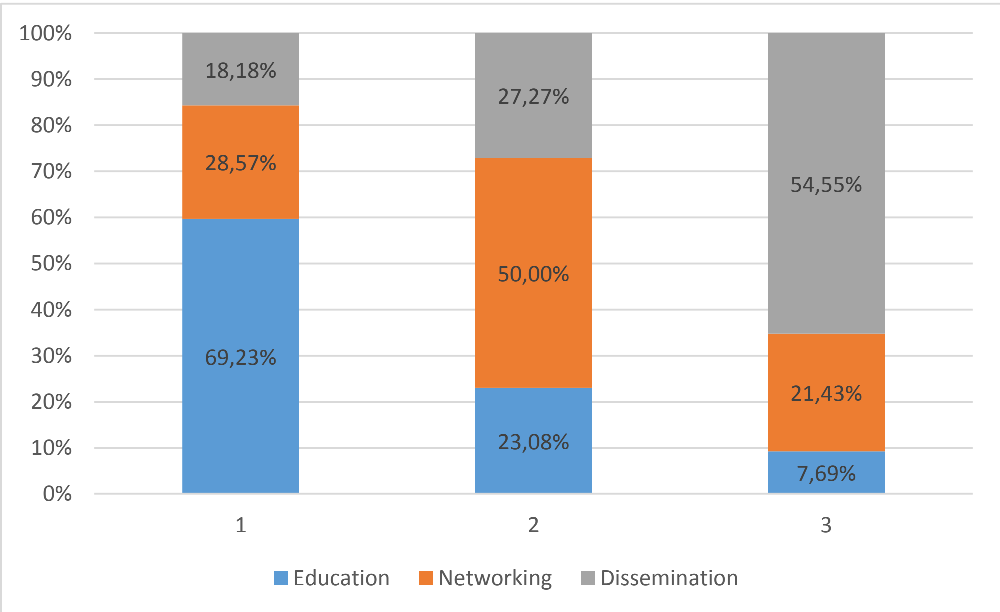

Figure 6.6: Ranked conference objectives

The outcomes of the user groups study, in particular the surveys, were employed in the investigation of the disaster management (Floch and Angermann, 2012), student (Papadopoulou et al., 2014) and enterprise scenarios (Lima et al., 2013). The role-play and participatory approaches were considered for prototyping the platform and third party services were also investigated as whole by Jennings et al. (2014).

6.3 CAFCA Standalone Trial

Based on the results of the qualitative and quantitative analysis conducted by the project, it was carried out an evaluation with CAFCA framework. Originally, the objective of the CAFCA evaluation was to be held in enterprise user group within the SOCIETIES trials. However, given the nature of the changes within Intel, this was deemed not possible (Roddy, 2014).

In order to assess CAFCA, an evaluation was structured into two parts, one by a pre and post questionnaires and the second part with a user trial. The first part consisted of an online pre-questionnaire, benefiting from collecting the information about each volunteer and his/her relationship with rest of the group. Moreover, it was designed a post questionnaire to be applied to the participants after the user trial. The second part involved a user trial, organized with the same participants in a conference scenario. The scenario consisted in identify experts, who have similar expertise or that work in related areas, to suggest them to join a chat room.

The user trial was carried out twice, with a pilot test and a final experiment based on the experiences of the pilot.

As a final point, CAFCA evaluation process followed a quasi-experimental approach. The quasi-experimental design involves selecting groups, upon which variables are tested, without any random pre-selection processes. The data collection of the first part was cross-sectional (specific point in time). In the second part, the framework collected the information in a longitudinal approach (several observations over a period).

6.3.1 Participants

In terms of demographic analysis, the user group consisted of eleven participants, six males and five females, ranging 25-31 years. The participants include students, technical staff and researchers that work in a research group at the University of Aveiro. The volunteers belong to CESAM[1] (Centre for Environmental and Marine Studies), a large group that is part of the Department of Biology. Although all participants have the same multidisciplinary backgrounds, the group works in different areas. Furthermore, the participants had familiarity with smartphones and chat applications, but are not expert in the area

[1] http://www.cesam.ua.pt/

6.3.2 Pre-Questionnaire

The online pre-questionnaire was the first part of the evaluation that included six questions ranging from multiple-choice questions to Likert scale questions (see Appendix A). The pre-questionnaire was twofold: background information of participants' area and questions about their relationship with rest of the group. The respondents answered individually the questions that followed a close-ended format, i.e. which provide the participants with a list of predefined names to choose from.

The data presented here is grouped into professional interests; daily work activities; skills and expertise; degree of similarity with other participants; analysis of privacy aspects and social media. This pre-questionnaire also gathered information about their professional interests and consequently used as context information for part two of the study. The questions were answered by 100% of the survey participants.

Q.1 Asked to indicate at least three skills and expertise, based on their daily work. The interests chosen by the participants in this question were used subsequently in the second part for the live user trial. Figure 6.7 presents theirs interests.

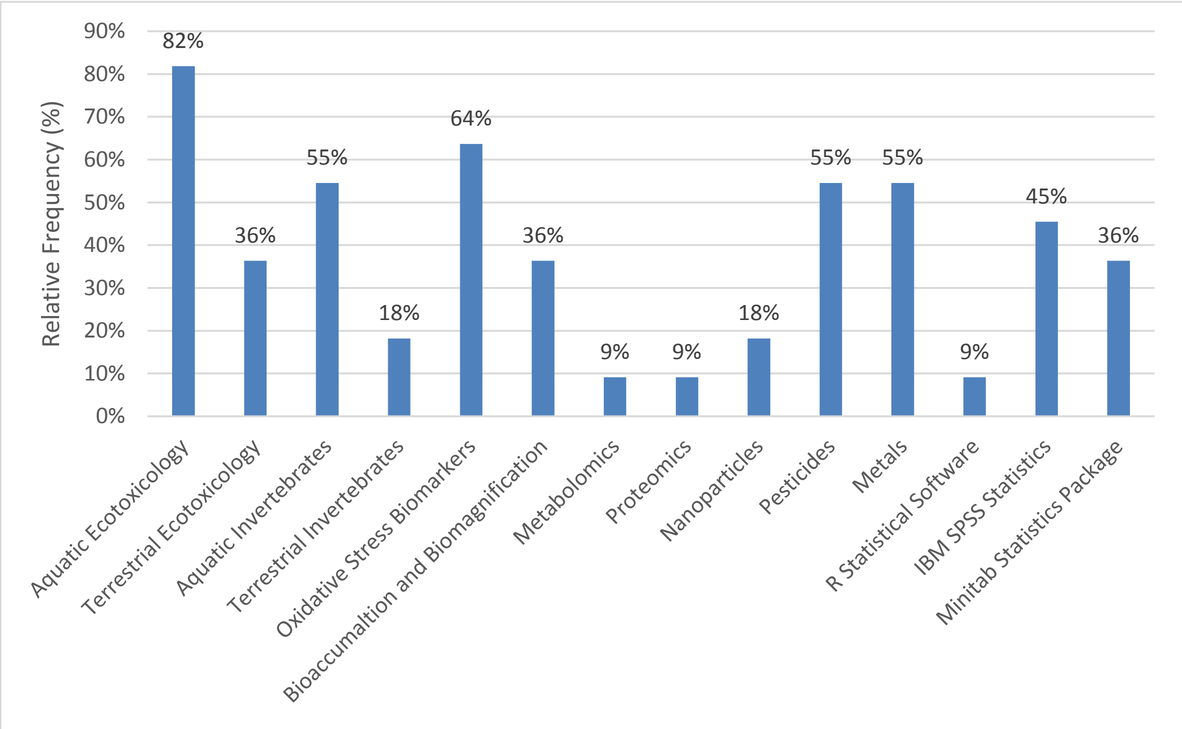

Figure 6.7: Professional interests of participants

Q.2 The participants were asked to specify those people that they probably have same work routines in terms of compounds class, organisms and methodology. It was instructed to them that, if they work with the same organisms, the same type of methodology and same compound to assign a higher value. On the other hand, if they work with same organism, same methodology but different compounds, assign a lower value. Table 6.2 present the results of the three working routines in an adjacent matrix.

This question was important to understand the perception of the respondent towards the others individuals of the group. The point of view of two people in some cases is in most cases similar. As a result, it is possible to verify their affinity among the participants comparing them with other questions in the survey.

Table 6.2: Matrix of persons that have the same work routines from respondent point of view.

	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	P11
P1		0%	100%	0%	0%	0%	0%	0%	0%	0%	0%
P2	67%		67%	0%	100%	67%	0%	100%	33%	0%	67%
P3	100%	67%		0%	33%	33%	0%	67%	0%	0%	0%
P4	0%	33%	0%		33%	33%	33%	0%	33%	33%	33%
P5	0%	100%	67%	0%		33%	0%	33%	0%	0%	33%
P6	0%	100%	33%	0%	33%		0%	33%	0%	0%	0%
P7	0%	0%	0%	0%	0%	0%		0%	0%	0%	0%
P8	0%	67%	67%	0%	67%	0%	0%		0%	0%	0%
P9	33%	0%	33%	33%	0%	33%	67%	33%		67%	100%
P10	0%	67%	0%	0%	0%	0%	0%	0%	33%		33%
P11	0%	33%	0%	0%	0%	33%	33%	0%	100%	33%	

Q.3 The respondents were asked to indicate which of the following people they consider working together on a daily basis. This question allowed the participant to choose more than one person per line in a table. In addition, the question excludes the name of the respondent himself.

Question three tries to verify that most of respondents have a good understanding of other members when compared with question two. Table 6.3 present the adjacent matrix of the results.

Table 6.3: Matrix of people that work together from respondent point of view.

	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	P11
P1		10%	90%	0%	10%	10%	0%	10%	0%	0%	0%
P2	0%		30%	0%	90%	50%	0%	60%	0%	0%	0%
P3	90%	30%		0%	10%	10%	0%	30%	0%	0%	0%
P4	0%	0%	0%		0%	0%	0%	0%	0%	0%	0%
P5	10%	80%	10%	0%		70%	0%	80%	10%	0%	0%
P6	10%	60%	10%	0%	60%		0%	70%	0%	0%	0%
P7	0%	0%	0%	0%	0%	0%		0%	0%	0%	0%
P8	10%	50%	30%	0%	70%	40%	0%		0%	0%	0%
P9	0%	0%	0%	0%	0%	0%	0%	0%		0%	90%
P10	0%	0%	0%	0%	0%	0%	0%	0%	0%		0%
P11	0%	0%	0%	0%	0%	0%	0%	0%	90%	0%	

Q.4 The participants were asked to specify how frequently they provide information related to their profession within social networks. The graph in figure 6.8 shows that the participants are divided about sharing their professional information.

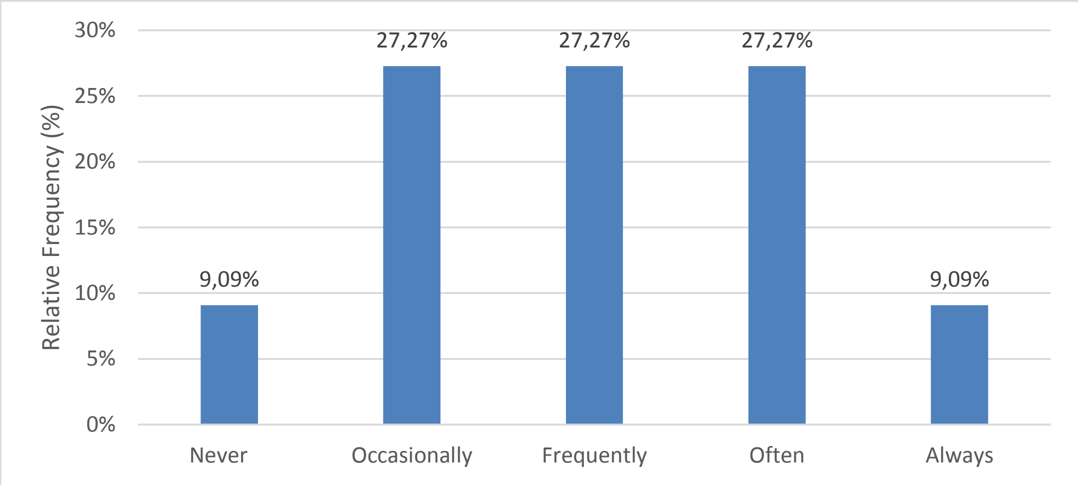

Figure 6.8: Frequency of professional information shared over SNS by the participants.

Q.5 Asked the participants to indicate their degree of similarity to the following people based on their professional interests and work practices (organisms, methodology and compounds). Table 6.4 presents the similarity from the point of view of each respondent. The table represents a matrix sum of all answers provided. As before, the question excludes the name of the respondent himself.

This question intended to verify, comparing questions two and three, the respondents' perception about their colleagues. When comparing the three tables it is possible to see some patterns regarding a few individuals, especially values very high as above 90% or even very low values as 10%.

Table 6.4: Matrix of degree of similarity based on their professional interests and work practices.

	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	P11
P1		25%	100%	0%	0%	0%	0%	0%	25%	0%	0%
P2	50%		75%	0%	100%	75%	25%	100%	50%	0%	50%
P3	100%	75%		0%	75%	75%	0%	75%	0%	0%	0%
P4	0%	0%	0%		0%	0%	0%	0%	0%	0%	0%
P5	25%	100%	75%	0%		75%	100%	75%	25%	0%	25%
P6	0%	75%	50%	0%	25%		0%	25%	0%	0%	0%
P7	0%	0%	0%	0%	0%	0%		0%	0%	0%	0%
P8	0%	75%	75%	0%	75%	75%	0%		0%	0%	0%
P9	50%	25%	25%	50%	0%	50%	50%	0%		0%	100%
P10	25%	25%	25%	25%	25%	25%	25%	25%	25%		25%
P11	0%	50%	0%	50%	50%	25%	0%	0%	100%	25%	

Q.6 Asked the participants to specify which social network sites they use in daily base. As demonstrated in figure 6.9, the most popular SNS were Facebook, LinkedIn and Research Gate. All the survey respondents indicated the use Facebook. LinkedIn and Research Gate are used by 73% and 43% of the participants respectively. Google Plus and Twitter are moderately popular with 28% of participants stating they used them, while Foursquare is used by 19%. Finally, Reddit[2] was mentioned by one respondent as other SNS.

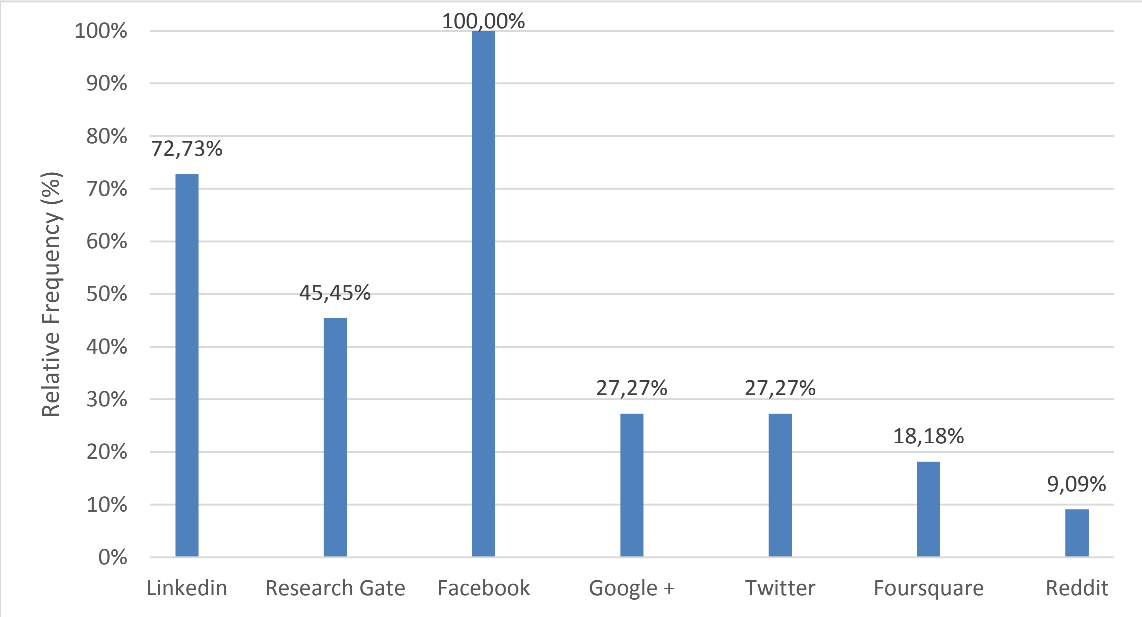

Figure 6.9: Usage of Social Network Sites.

6.3.3 User Trial Procedure

In order to test CAFCA, the SOCIETIES platform and its CMS were used by the framework as providers of context information. The test infrastructure was set up within the datacentre of the Institute of Telecommunications, Aveiro. The server was a virtual machine instance in an Intel i7 Quad Processor (3.07 GHz per core) with 16 GB RAM running VMware 5.1. This dedicated Linux machine was connected to the EDUROAM network. Aside from the server, eleven more client nodes were required for each participant. The users' platform nodes were also virtual machine instances with the same specification as the server machine with 1 GB RAM running 32-bit Ubuntu 12.04 operational system.

The server instance had installed an Admin Cloud node version in order to manage the creation, organisation and communication of the communities. For each client node was installed the User nodes version, which interact with the Admin Cloud node. All the

[2] http://www.reddit.com/

machines have in common Eclipse Virgo[3], an OSGi service container with Apache Tomcat[4] web server integrated. Finally, MySQL[5] was used as database for data storage necessary requirement for platform.

The clients were structured into rich client (desktop frontend) and light client (mobile frontend). The rich client is based on a Virgo container, and the components are implemented as OSGi bundles for this platform. In light client, these components are packaged as Android services and library for this node.

Although the user test comprised exclusively Android mobile phones, each user needed a user client node configured to be able to interact with the public administrator cloud node. Figure 6.10 illustrates the structure.

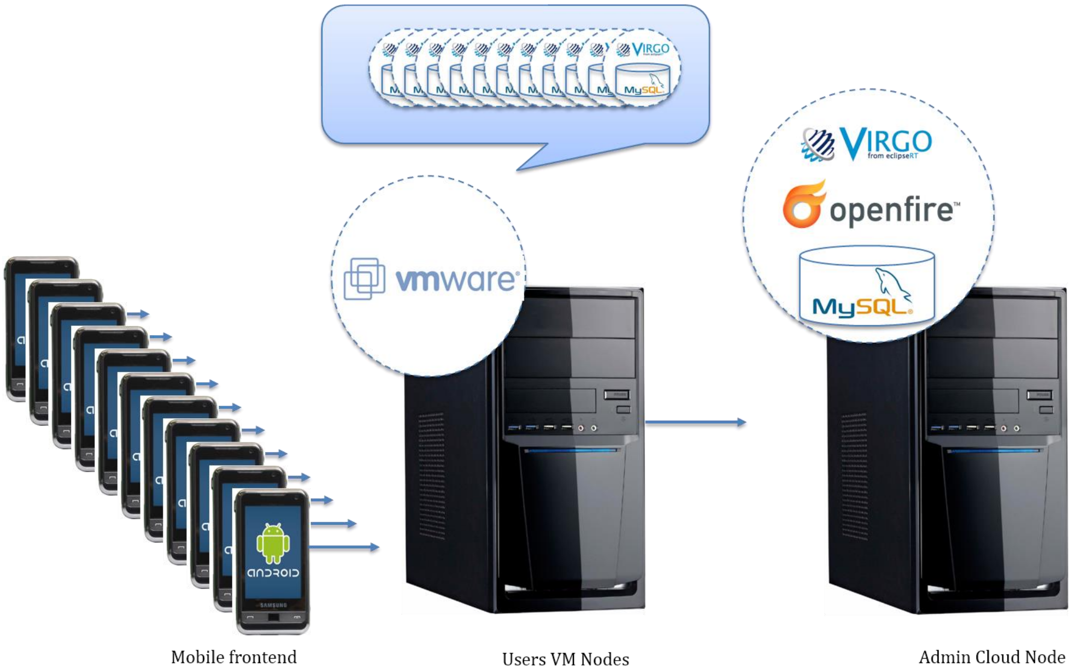

Figure 6.10: Setup used in the user trial.

For the trial, CAFCA was deployed as a third-party service on the server platform machine. The deployment is done in runtime as an OSGi bundle, supporting integration and operability with other components. In order to integrate CAFCA into the CMS component, the administrator of the sessions must choose the context information that will be mapped to the framework's graph database. Such information is presented in a predetermined list available from the CMS component. The list is not exhaustive and can be managed through the platform's API in case of addition or removal.

[3] http://www.eclipse.org/virgo/

[4] http://tomcat.apache.org/

[5] http://www.mysql.com/

Additionally, a chat client application for Android was elected for the communication among participants through invitations to join in chat rooms (see Appendix C). In order to create the client accounts for chat application, the same members' accounts available in the platform for authentication was used. Since the platform uses Openfire[6] server to create and manage XMPP accounts for clients and communication, it was not required to register each user a second time for the chat client. As seen in Chapter 3, XMPP is a message protocol based on XML and widely used over the Internet.

Lastly, the integration of the chat application into the CAFCA framework was done using an open source XMPP client library for IM and presence called Smack[7]. Smack provides an API to utilize multi-user chat functionalities in the Openfire server.

In terms of privacy issues, the SOCIETIES platform provides a privacy policy manager. This component is responsible for negotiating the terms and conditions for interact and disclose data between two parties. The privacy policies are associated to a new community or a new third party service while the agreement will be securely stored using CB. For example, when a community is created manually, the owner has to select a privacy policy among three choices: data available to nobody, only community members, or for all users. The interface illustrated in figure 6.11 allows the end-user to visualize the privacy policy of a community or a third party service.

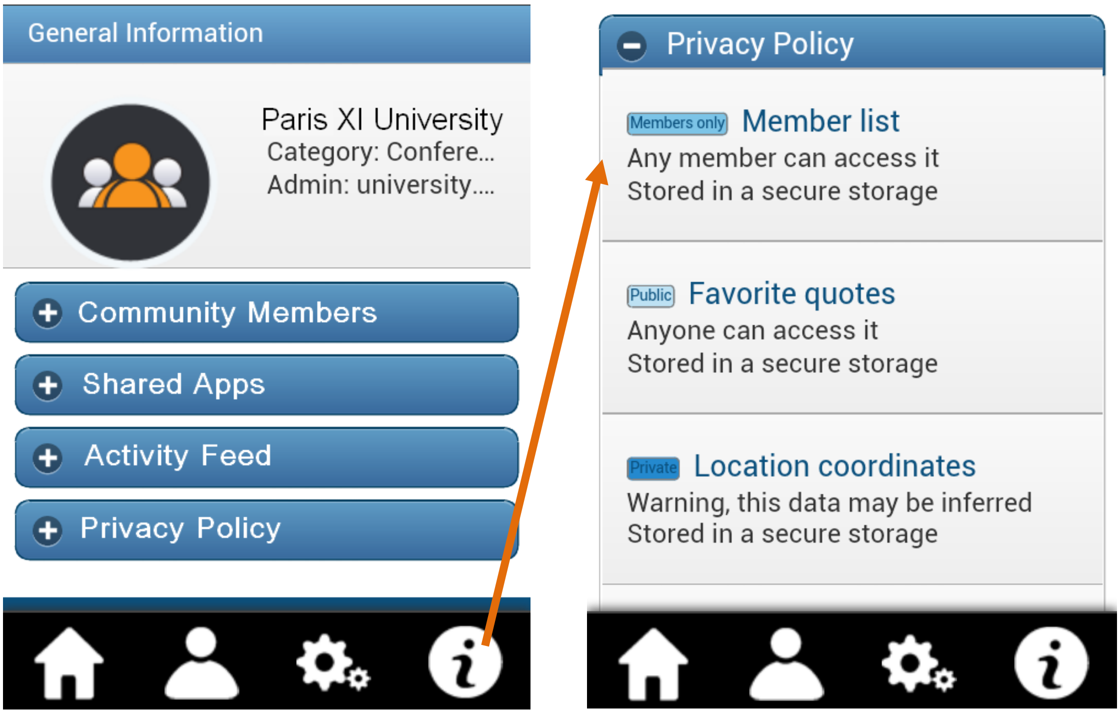

Figure 6.11: Privacy policies of a community on an Android device.

[6] http://www.igniterealtime.org/projects/openfire/

[7] http://www.igniterealtime.org/projects/smack/

6.3.4 Instructions to the group

The test environment was set up in a social room within the Department of Biology at University of Aveiro. The room was chosen because it allows access to EDUROAM[8] network and does not disrupt the routines of the participants. The social room is provided with around ten tables for notebooks and lockers on the wall. Currently the participants use the room to browse the web, check email and edit document files when they are not doing laboratory experiments.

Each participant used an Android 4.1 LG L3 II smartphone with two applications installed. One was the Android client application that allows the user to verify their user profile, personal details and his communities. The second was a chat application called Xabber[9].

To simulate the conference scenario, a modification was made in the client application allowing the users to check into different sessions in real-time. In other situations, the platform would use an external system from IBM Presence Zones[10] to detect automatically the physical location of the participants. For the sake of simplicity, this part was abstracted, with users changing the location manually. Figure 6.12 presents screenshots from the client application and chat application.

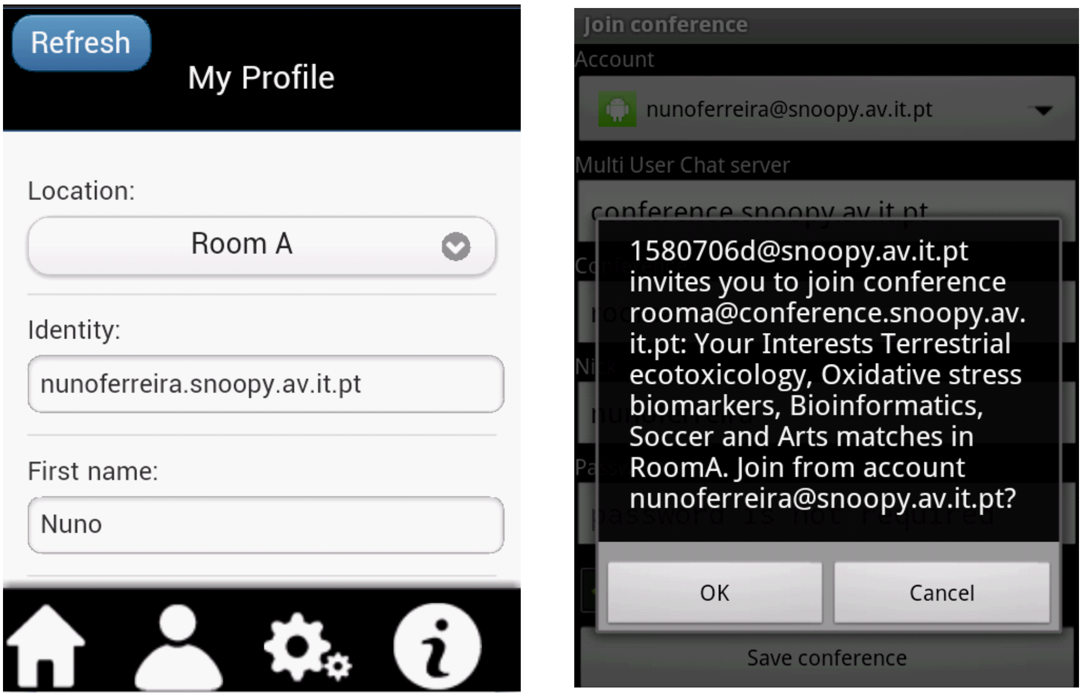

Figure 6.12: Screenshot on the left shows a user profile example. On the right, an invitation to join a chat room.

---

[8] https://www.eduroam.org/. Abbreviation for "Education Roaming". International roaming service research and higher education community.

[9] http://www.xabber.org/

[10] http://www.ibm.com/software/products/en/presence-zones

6.3.5 Pilot Trial

Before the effective experiment, a pilot session was held on November 2013 in order to test the pre-questionnaire, experiment procedure and the prototype. Based on their feedback, the necessary adjustments to the questions and the prototype were made.

The actual questionnaire was modified to incorporate a supplementary social vision, which included changes on sorting the question and the way that the questions were driven to the participants. Most changes followed propositions for SNA found in Borgatti et al. (2013) .

Some technical problems were found during the test. First, EDUROAM Wi-Fi network that connected the mobile phones to the machines in Institute of Telecommunications building were occasionally failing. Apparently, a network maintenance occurred during the trial period. Second, the context enrichment was re-evaluated after the pilot test and the increment was adjusted with a limit of two words for each context information. Consequently, each individual interests will only be increased with a maximum of two words.

We verified that more words lead to an increase (or decrease) in the similarity among participants in this scenario. Thus, CAFCA may have results that are more relevant by applying the context increment limiting to a maximum value.

6.3.6 User Trial

In order to evaluate genuinely CAFCA, the experience was scheduled with the same group of eleven volunteers of the pilot test in April 2014. The trial was developed based on a storyboard of a conference scenario, running both applications at same time. First, the conference participants were able to choose in which session room location they want to participate. Second, based on their user profile information, the framework suggests them to join a chat room. The research objectives of the trial were:

- Discovery of similar persons to initiate group interactions.
- Pervasive integration of physical and social interactions (based on people attending the conference session's room).
- Evaluate the effectiveness of CAFCA algorithms in identifying common interests and potentially appropriate collaborative sessions for the participants.

With the information available from the pre questionnaire, user accounts were created in the SOCIETIES platform and the profiles filled with the data provided. As long-term context information it was used the interests indicated by question one, while for short-term it was chosen session rooms of the conference. For privacy reasons, the social connectors available in the platform were not employed in the test. Instead, the data obtained through the survey is extremely close to the professional information provided by the participants in SNS like Research Gate and LinkedIn.

The evaluation followed two phases. The first intended to evaluate the impact of priority mode on the participants' relationship of the social graph. The second intended to evaluate the relevance mode setting, which facilitates the generation of suggestions based on all participants' relationship in a community.

Hence, the test was divided into two rounds of fifteen minutes each. In the first round, it was set CAFCA to evaluate the members with priority mode. It was stipulated two rules and assigned a sequence for location followed by interests. Consequently, depending first on location and second on interests, the runtime monitor checks whether it is appropriate to suggest a session or not. The rules were expressed as follows:

- First rule - Location: same room session location
- Second rule - Interests: The interests among individuals on the same location need to be higher than the auto-threshold.

Worth noting that in both cases, the framework adopted a longitudinal perspective for location, which means that location was changing over the time.

The figure below illustrates the sessions formed based on the three room locations. Each circle represents the number of individuals that were invited by the framework. In addition, the case below illustrates the use of semantic increment of the interests.

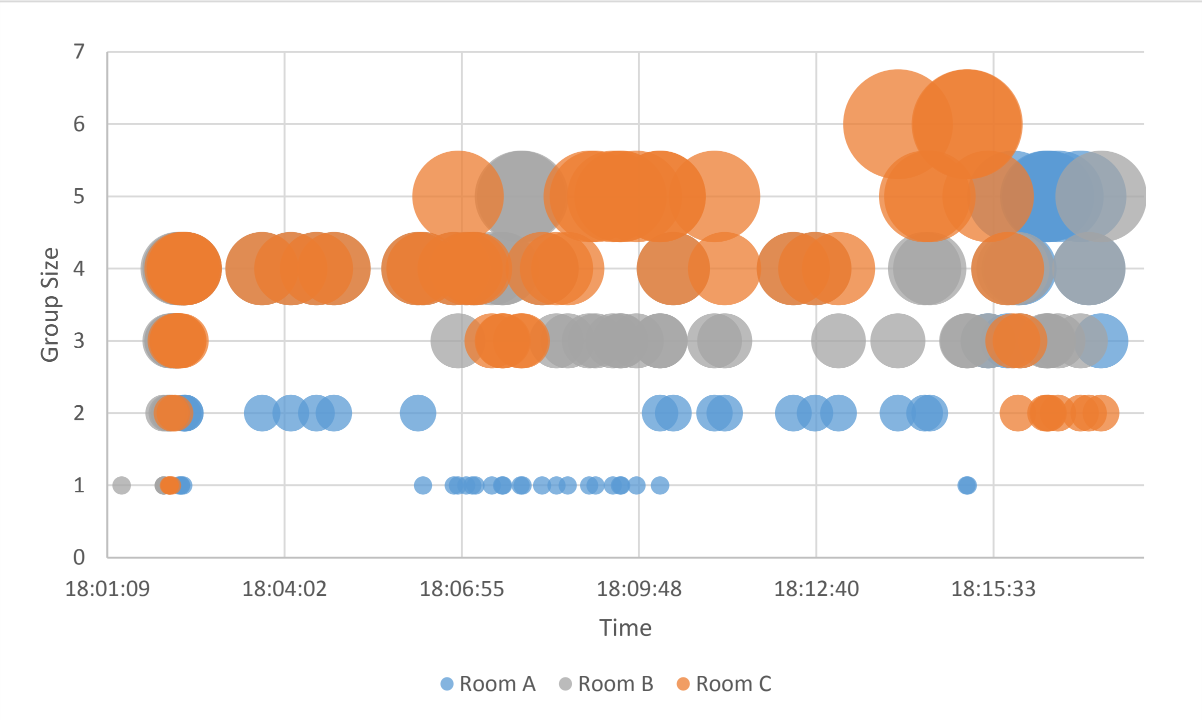

Figure 6.13: Priority groups with context enrichment.

According to figure 6.13, room C had most of the users while room A had at maximum two participants along the duration of the test.

In figure 6.14 is presented the same group of individuals, but this time without semantic enrichment in the users' interests. This test was done after the trial with the log files. Next section explains in details.

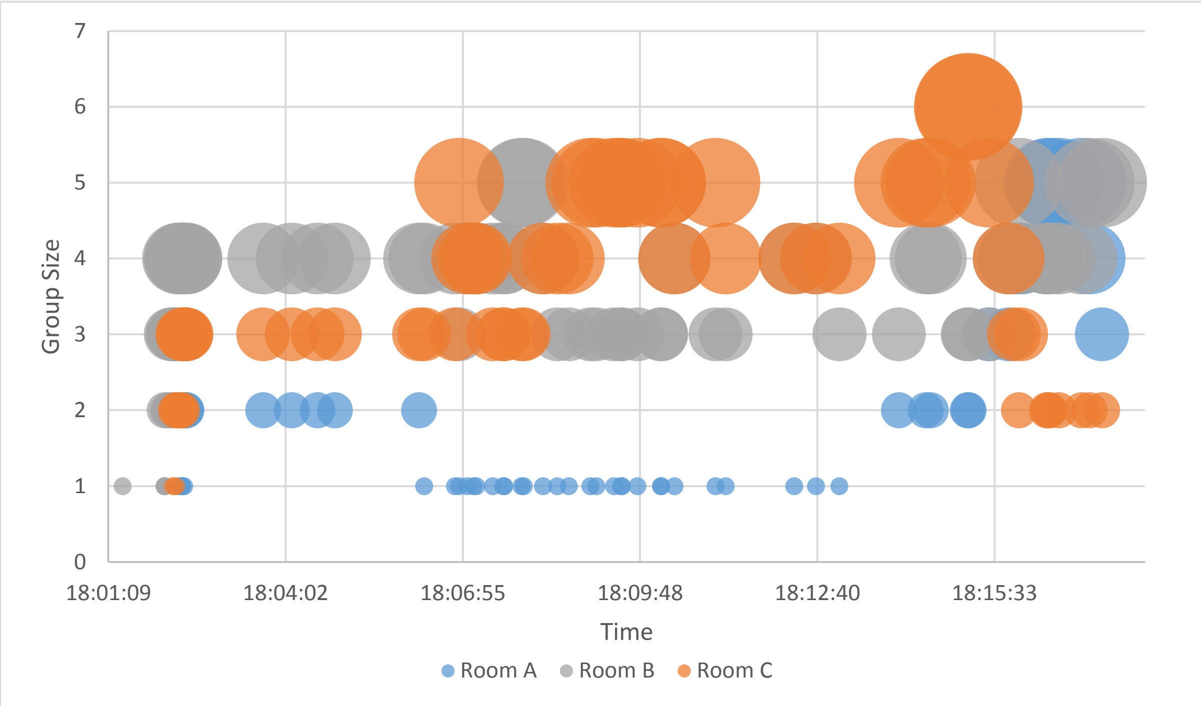

Figure 6.14: Priority groups without context enrichment.

These figures suggest that the enrichment had a subtle effect on the priority mode of evaluation for group formation. Room B was the location with most differences among the other rooms, especially at the beginning. Room C presented some slightly differences at the beginning again, comparing to figure 6.13. While room A presented other group formation in the end of the test.

Without disrupting the trial, the framework monitor was set to evaluate in relevance mode. The users at the time of the change were alerted. This step was necessary to complete a question in the post-questionnaire.

The weights assigned to the rules were the following:

- First rule - Location: same room session location with weight 30%.
- Second rule - Interests: The interests among all individuals to be higher than the auto-threshold and with a weight of 70%.

In figure 6.15, the group formation for relevance mode and with context enrichment is presented. Differently from the priority evaluation, the relevance test started with sessions formed and the members on their currently locations. The changes were more related to the time of session formation than the size of the group. Room C had most of users during the entire test ranging from 6 to 9 participants. Worth noting, that the relevance mode was more optimistic in relation to the priority mode, grouping more people.

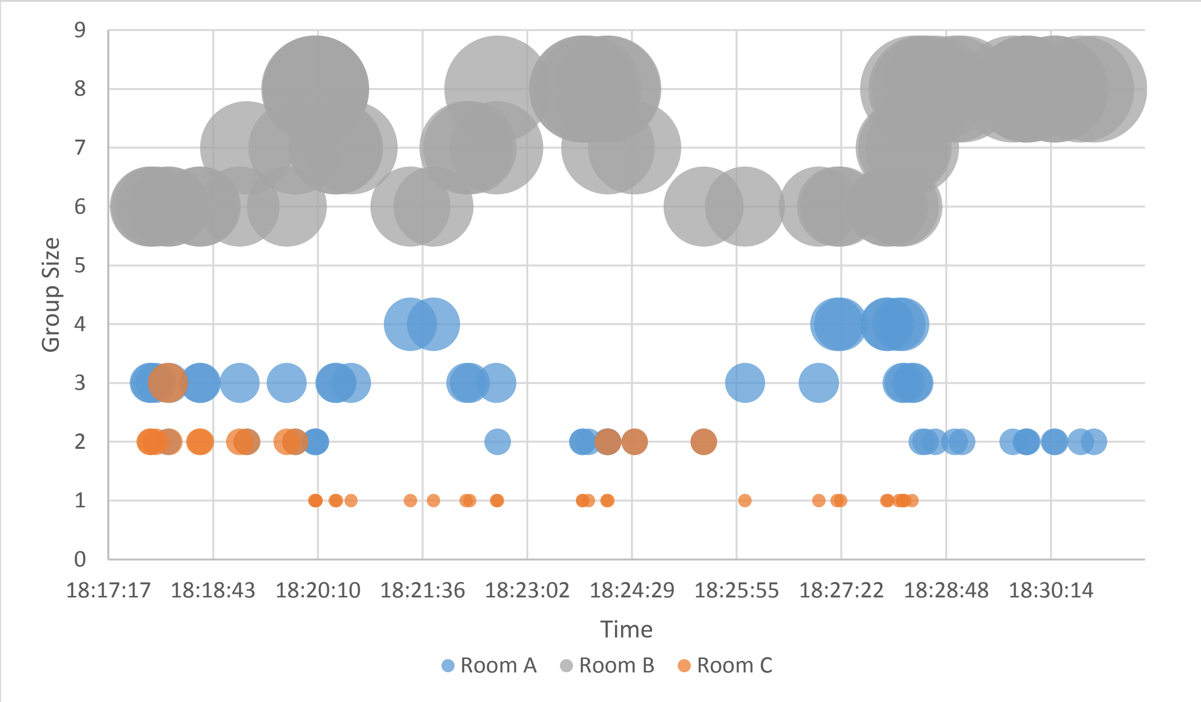

Figure 6.15: Relevance groups with context enrichment.

Figure 6.16 presents the group formation for relevance mode without context enrichment. Compared to the previous mode, the differences are subtler than the priority mode without context enrichment. Room A presented some significant changes at the end of the test. Again, these results were simulated after the trial.

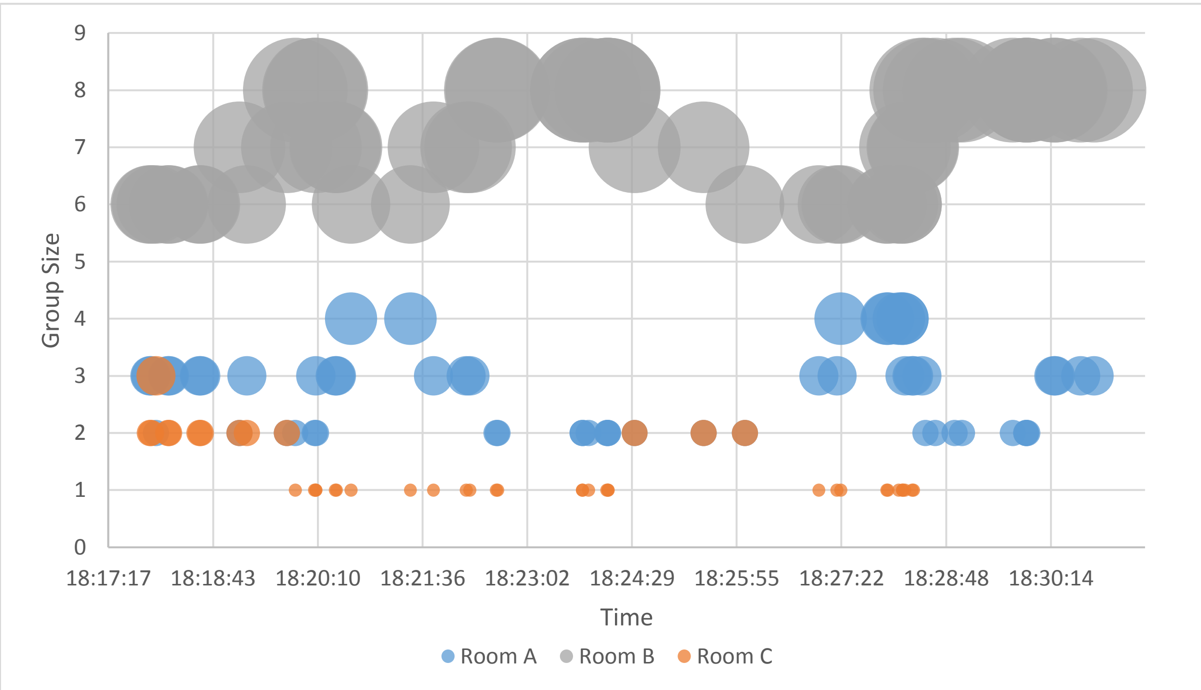

Figure 6.16: Relevance groups without context enrichment.

6.3.7 Post-Questionnaire

After the test, the participants were requested to complete a short questionnaire with four questions about the trial experience. The questions were answered by 100% of the survey participants.

First, the participants were asked to specify their acceptance about the first stage of the test, which the framework engine mode was set to priority mode. The relative distribution is illustrated in figure 6.17.

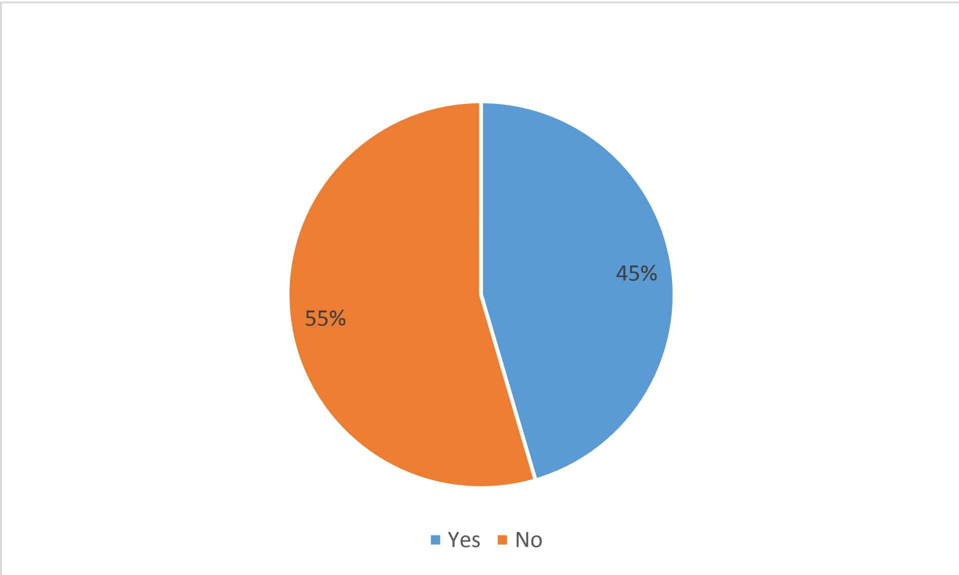

Figure 6.17: Acceptance of suggestions made by priority mode.

In priority mode, 45% of participants that answered the question do sometimes felt the framework suggested appropriated sessions, while 55% of participants, did not feel appropriate such suggestions.

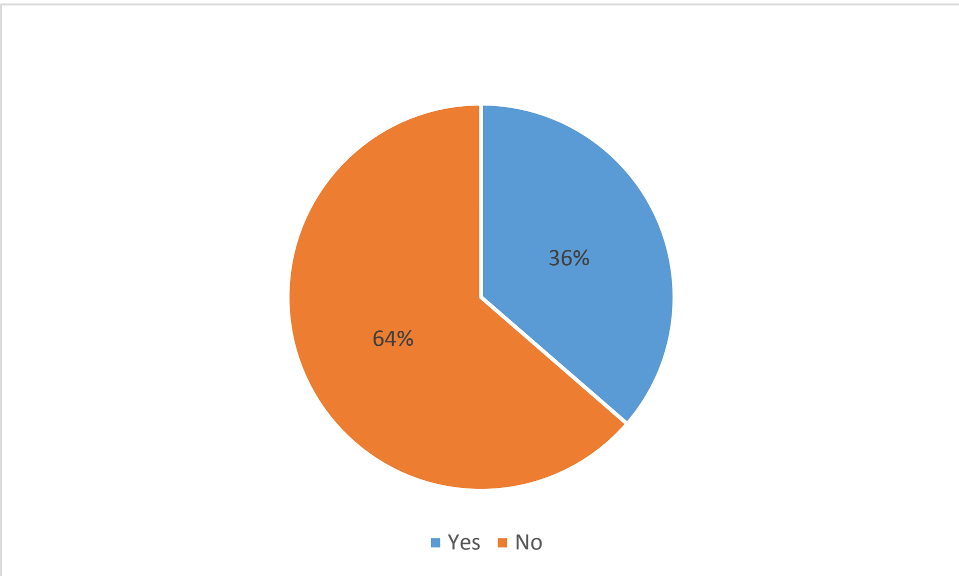

Figure 6.18: Acceptance of suggestions made by relevance mode.

Regarding the second stage that comprised the relevance mode, 64% of participants do sometimes felt the framework suggested the appropriate sessions. 36% of participants did not feel appropriate such suggestions.

Although relevance mode is more complex in terms of computation, most users believe that priority mode is more suitable. While the relevance mode is optimistic, the priority mode filtering characteristics offer best suggestions.

Concerning both stages of the trial, participants were asked to indicate the most suitable stage regarding their interests.

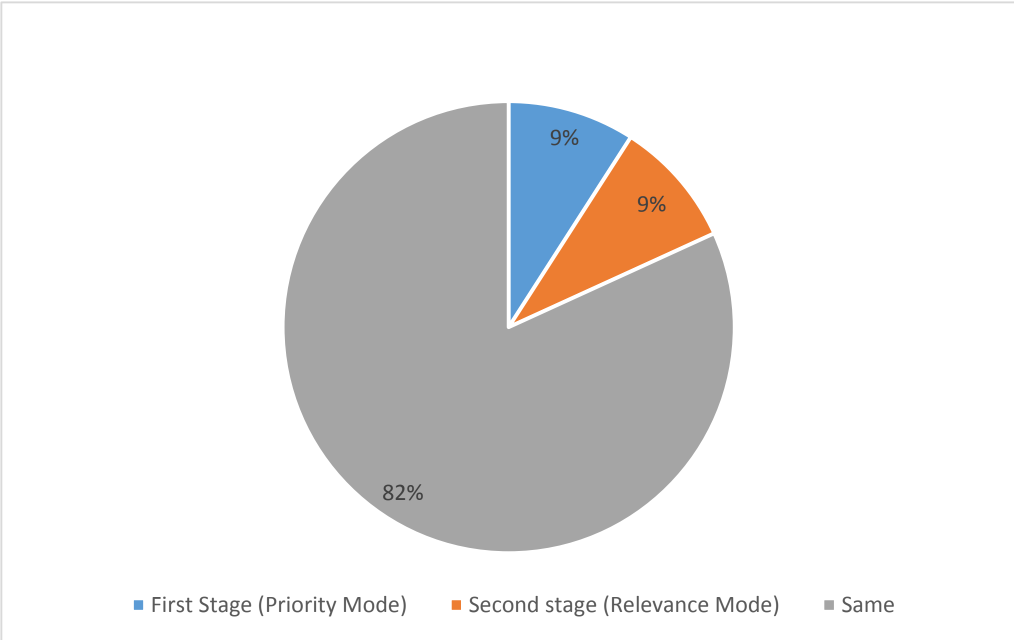

Figure 6.19: Feedback from participants about both stages of evaluation.

Most of the participants specified that both stages were the same from their perspective. Only one of the participants answered that the first stage was more suitable. While, only one participant preferred the second stage. The answers suggest no differences between priority mode and relevance mode for session's formations form the point of view of the users. The values also indicate that it is difficult for the users to notice the behaviour between engine modes.

The next question asked participants to specify the level of trust to share professional information with this type of application. Likert scale was again applied in this question in order to confront the question four from pre-questionnaire with 1 to "Don't trust my information" and 5 to "Totally trust". With 64% the participants indicating that they trust their information. With 36%, the they thing that is reasonble to share this type of information indicating number 3. Table 6.5 present the results.

Table 6.5: Feedback from participants about both stages of evaluation.

	Trust
Don't trust my information	0%
2	0%
3	36%
4	64%
Totally trust	0%

The responses showed that most participants feel safe to share professional information in this type of scenario.

Last question asked if the respondents would find useful an application that suggests participating in discussion groups with common interests. The question is measured on a ten point Likert scale, ranging from 1 “Not useful” to 10 “Very useful”. Most users think is useful to receive suggestions based on theirs interests with 36% designating 8. Table 6.6 present the results.

Table 6.6: Participants who believe it is useful suggestions to suggest participate in groups with common interests.

	Group Suggestions
1 – Not useful	0%
2	0%
3	9%
4	0%
5	0%
6	0%
7	18%
8	36%
9	27%
10 – Very useful	9%

The values represent good acceptance from the users regarding the suggestions. In this way, only 9% of participants indicated that they would not like any of the suggestions, while the remaining 91% of participants indicated that they would enjoy such behaviour.

6.3.8 Results and Discussion of the User Trial

After the trial, it was analysed the user usage based on log data from the platform and data stored in the graph database. The log files were created using Apache log4j API and were saved in a structure consisting of a date-time stamp and directories relating to each individual. The database dumps consisted of the Neo4J database containing all the accumulated data.

The log data enabled a better understanding among the participants' activities. The test aimed in co-located collaboration using theirs interests as it represented challenge to CAFCA.

The context enrichment offered advantage in situations where the users have broader interests, leading the social matching mechanism for more precise results. Results showed slight differences in terms of group invitation according to the results and this behaviour should be complete opposite with individuals not sharing any interests in the same wide field.

The post survey indicated that most users are favourable with the suggestions made by CAFCA in real world situations. The differences between engine modes are very subtle from the point of view of the users and consequently its behaviour should be related to the situation. Consequently, the administrator is in the position of choosing one of the engine modes, depending on his goals. Furthermore, using different context information, by including additional rules, may lead to better results depending on the heterogeneity of the group.

Regarding implications for framework usability, the users had demonstrated since the beginning that mobile devices would be more practical, providing more acceptability for trial. Previously tests with other SOCIETIES services demonstrated that is confusing for the participants to change among the applications. Integrating other existing collaborative applications on mobile devices will be challenging for the users. Many applications on desktop provide means to use for instance VoIP and chat on the same interface in a multiplatform fashion (E.g. Jitsi[11]), while mobile devices are recently supporting this.

Worth noting, is the CAFCA social matching mechanism that was used by another service called Shared Calendar[12]. The service used an external web service, Google Calendar, working as a joint calendar. The objective was to identify other participants with high overlap of expertise and interests in order to suggest similar events in the calendar. However, deep experimental tests were not conducted, being planned only for demonstration purposes.

[11] https://jitsi.org/

[12] https://github.com/societies/SOCIETIES-SCE-Services

6.4 Performance based on the context information

The performance and computational complexity of the rule engine were directly linked to the selected context information used by the rules and by the number of people analysed. In priority mode, the performance is expected to be better than the relevance mode, since each rule will not consider all the community individuals. On the other hand, in relevance mode each rule will comprise matrices that need to be calculated including all individual as described in Chapter 5. The performance of the priority and relevance mode are presented in table 6.8 and 6.9 in CPU time.

The rule engine was designed to consider small groups ranging from 10 to 20 people; however, it was simulated tests until 50 people. The individuals and their context information were randomly generated and subsequently included in the graph database. The rules and the context samples were based on information given by the participants in the questionnaire. The performance measure was done in an Intel i7 2.4 GHz with 8 GB RAM.

First, it was measured the priority mode considering the amount of time to process the rules. As illustrated in Table 6.8, the number of rules inserted does not sharply influence the processing time. In relevance mode, as the number of rules increase the processing time significantly intensifies. In the pilot test with two rules for each mode, the priority had a mean value of 11ms, while the relevance had 70ms.

Table 6.7: Priority Mode

	Rule 1	Rule 2	Rule 3	Rule 4	Rule 5
10 people	0.01 s	0.01 s	0.01 s	0.01 s	0.01 s
20 people	0.01 s	0.01 s	0.01 s	0.01 s	0.01 s
30 people	0.011 s	0.016 s	0,031 s	0.031 s	0.031 s
40 people	0.016 s	0.031 s	0.031 s	0.031 s	0.031 s
50 people	0.016 s	0.031 s	0.031 s	0.031 s	0.031 s

Table 6.8: Relevance Mode

	Rule 1	Rule 2	Rule 3	Rule 4	Rule 5
10 people	0.04 s	0.07 s	0.09 s	0.10 s	0.12 s
20 people	0.17 s	0.32 s	0.37 s	0.39 s	0.43 s
30 people	0.31 s	0.64 s	0.68 s	0.73 s	0.76 s
40 people	0.55 s	1.15 s	1.25 s	1.34 s	1.43 s
50 people	0.67 s	1.69 s	1.72 s	1.84 s	1.95 s

6.5 Conclusions

This chapter presented the test and evaluation of CAFCA framework in a pervasive environment with a group of users. The evaluation shows that the proposed work can enhance the performance of the human-system interaction in pervasive environment. The approach took into consideration precisely pervasive community of users, enabling runtime adaptation of collaborative tools in response to the context changes for promoting social interaction between conference participants.

In the case study, CAFCA was effectively integrated as a third party service to the SOCIETIES pervasive platform and tested with a small group of eleven participants. Additionally, the framework was also integrated to a chat application to provide the communication tool for interaction along the test.

7 CONCLUSIONS

The main contribution of this thesis is to provide a context-aware framework to support collaborative applications in pervasive environments. The research focused on the design and development of a framework solution with an innovative socio-technical approach to exploit collaboration in pervasive communities. The novelty of the Context-Aware Framework for Collaborative Applications (CAFCA) lies on social matching capabilities for session formation, communication and coordination of groupware for collaborative activities.

A study in the state of the art was performed to cover the actual development in pervasive systems, context-aware, social computing and communities of practices. Some of these fields present combined paradigms and others only isolated works (Lima et al., 2012). All the combined areas brought many benefits to the CAFCA design principles and development.

CAFCA offers functionalities not encountered in pervasive environments for collaborative work, thanks to the integration of context-aware framework with CSCW applications. These context-aware CSCW applications merged with social computing can bring great benefits for AmI participants by using context and social information to provide adaptability for collaborative applications. The architecture of CAFCA is modular and can be extend enabling also to be connected with many context management systems.

The framework follows an approach of social network analysis to identify situation relevant and socially related partners. In addition, the solution exceeds current approaches by using context and social information about the current users' situation, leading to proactive behaviour and adaptability for end-users. The solution also supports the integration of existing collaborative applications for situated end-user cooperation.

In order to validate the framework, a group chat application was integrated to perform the trial. The users' trial demonstrated the successful use of the framework proposed and how the planned features facilitate collaborative activities.

The CAFCA framework is classified as an infrastructure proposal. An infrastructure is responsible for supporting the construction or operation of other software. According to Edwards et al. (2003), the evaluation of an infrastructure is challenging since it is not easy for the end user to determine the separation between infrastructure features and user-visible features. Consequently, when users are using applications, they will not judge only the features, but the software as a whole.

Finally, context-aware CSCW applications merged with physical and social interaction can bring great benefits for the participant's usability. Furthermore, the author believes that context information and social context are central points to the development of pervasive systems for the future collaborative applications

In addition to this research work, the author had the opportunity of being involved in two European research projects SOCIETIES (Self-Orchestrating CommunIty ambiEnT Intelligence Spaces) and PERSIST (Personal Self Improving Smart Spaces) which provided means for evaluation and tests of the framework proposal.

7.1 Future Work

Challenges remain for interpretation of context and social information related to individuals and groups for CSCW in pervasive environments. In future, the author intends to validate CAFCA with other context-aware management systems for context interpretation and analysis. In addition, it is interesting an evaluation to verify usage and usability of the framework with other collaborative tools.

Some of the plans to explore in the future are:

- Interpret the history of context information from formed sessions to pre-collaborative and post-collaborative to provide other adaptation behaviours;
- Explore the use of collaborative session history with the goal to merge and split groups based on their evolution;
- Collection and storage of content generated by collaborative tasks (e.g. authors, conversations) to allow persistence information about the group, which can lead to form future sessions;
- Identify critical time moments in the collaboration, for example, the time when groups' membership changes most. This will imply the use of temporal characteristics in dynamic social network analysis;
- Manage sessions history of collaborative activities found among pervasive communities;

- Provide the user with explanations about adaptability behaviour of the framework taking into account different scenarios;

8 REFERENCES

Abowd, G., Dey, A., Brown, P., Davies, N., Smith, M., Steggles, P., 1999. Towards a Better Understanding of Context and Context-Awareness, in: Handheld and Ubiquitous Computing. pp. 304–307. doi:10.1007/3-540-48157-5_29

Al-Jaroodi, J., Al-Dhaheri, A., Al-Abdouli, F., Mohamed, N., 2009. A Survey of Security Middleware for Pervasive and Ubiquitous Systems. 2009 International Conference on Network-Based Information Systems 188–193. doi:10.1109/NBiS.2009.31

Amaral, V., Lima, S., Mota, T., Chainho, P., 2014. Exploring WebRTC Technology for Enhanced Real-Time Services, in: Rocha, Á., Correia, A.M., Tan, F.B., Stroetmann, K.A. (Eds.), New Perspectives in Information Systems and Technologies, Volume 2 SE - 5, Advances in Intelligent Systems and Computing. Springer International Publishing, pp. 43–52. doi:10.1007/978-3-319-05948-8_5

Arnaboldi, V., Conti, M., Delmastro, F., 2014. Cameo: a novel context-aware middleware for opportunistic mobile social networks. Pervasive and Mobile Computing 11, 148–167. doi:10.1016/j.pmcj.2013.09.010

Arrow, H., McGrath, J., Berdahl, J., 2000. Small groups as complex systems: Formation, coordination, development, and adaptation. SAGE Publications.

Baldauf, M., Dustdar, S., Rosenberg, F., 2007. A survey on context-aware systems. International Journal of Ad Hoc and Ubiquitous Computing 2, 263. doi:10.1504/IJAHUC.2007.014070

Bardram, J.E., Hansen, T.R., 2010. Context-Based Workplace Awareness. Computer Supported Cooperative Work (CSCW). doi:10.1007/s10606-010-9110-2

Bardram, J.E., Hansen, T.R., 2004. The AWARE architecture, in: Proceedings of the 2004 ACM Conference on Computer Supported Cooperative Work - CSCW '04. ACM Press, New York, New York, USA, p. 192. doi:10.1145/1031607.1031639

Bastian, M., Heymann, S., Jacomy, M., 2009. Gephi: An Open Source Software for Exploring and Manipulating Networks, in: International AAAI Conference on Weblogs and Social Media.

Beamon, B., 2011. Adaptive high level context reasoning in pervasive environments. University of Texas at Arlington.

Bellavista, P., Montanari, R., Das, S.K., 2013. Mobile social networking middleware: A survey. Pervasive and Mobile Computing 9, 437–453. doi:10.1016/j.pmcj.2013.03.001

Berger-Wolf, T.Y., Saia, J., 2006. A framework for analysis of dynamic social networks. International Conference on Knowledge Discovery and Data Mining 523.

Biegel, G., Cahill, V., 2004. A framework for developing mobile, context-aware applications, in: Second IEEE Annual Conference on Pervasive Computing and Communications, 2004. Proceedings of the. IEEE, pp. 361–365. doi:10.1109/PERCOM.2004.1276875

Boldrini, C., Conti, M., Delmastro, F., Passarella, A., 2010. Context- and social-aware middleware for opportunistic networks. Journal of Network and Computer Applications 33, 525–541. doi:10.1016/j.jnca.2010.03.017

Bonchi, F., Castillo, C., Gionis, A., Jaimes, A., 2011. Social Network Analysis and Mining for Business Applications. ACM Transactions on Intelligent Systems and Technology 2, 1–37. doi:10.1145/1961189.1961194

Borgatti, S., Everett, M., Freeman, L., 2002. Ucinet for Windows: Software for Social Network Analysis.

Borgatti, S.P., Everett, M.G., Johnson, J.C., 2013. Analyzing Social Networks, 1 edition. ed. SAGE Publications Ltd, London.

Borghoff, U.M., Schlichter, J.H., 2000. Computer-Supported Cooperative Work: Introduction to Distributed Applications.

Bourdieu, P., Wacquant, L.J.D., 1992. An Invitation to Reflexive Sociology. University Of Chicago Press; 1st edition.

Brézillon, P., Borges, M.M.R.S., Pino, J.J.A.J., Pomerol, J.-C.J., 2004. Context-Based Awareness in Group Work. FLAIRS Conference 575–580.

Cabitza, F., Locatelli, M., Sarini, M., Simone, C., 2006. CASMAS: Supporting Collaboration in Pervasive Environments, in: Fourth Annual IEEE International Conference on Pervasive Computing and Communications (PERCOM'06). pp. 286–295. doi:10.1109/PERCOM.2006.15

Cabitza, F., Simone, C., 2013. Computational Coordination Mechanisms: A tale of a struggle for flexibility. Computer Supported Cooperative Work (CSCW) 22, 475–529. doi:10.1007/s10606-013-9187-5

Cheaib, N., Mallem, M., Otmane, S., 2011. Tailorable Groupware Design Based on the 3C Model. International Journal of Cooperative Information Systems 20, 405 – 439. doi:10.1142/S0218843011002286

Cheaib, N., Otmane, S., Mallem, M., 2008. Integrating internet technologies in designing a tailorable groupware architecture, in: 2008 12th International Conference on Computer Supported Cooperative Work in Design. IEEE, pp. 141–147. doi:10.1109/CSCWD.2008.4536973

Chen, G., Kotz, D., 2000. A Survey of Context-Aware Mobile Computing Research. Technical Report TR2000-381 1–16.

Chen, H., Finin, T., Joshi, A., 2003. An Intelligent Broker for {Context-Aware} Systems. Adjunct Proceedings of Ubicomp 12 – 15.

Chen, H.L., 2004. An Intelligent Broker Architecture for Pervasive Context-Aware Systems. Vasa. University of Maryland.

Chihani, B., 2013. Enterprise context-awareness: empowering service users and developers. Institut Mines-Telecom, Telecom SudParis.

Chihani, B., Bertin, E., Jeanne, F., Crespi, N., 2011. Context-aware systems: a case study. Digital Information and Communication Technology and Its Applications 2, 1–15.

Chin, A., Zhang, D., 2014. Introduction, in: Chin, A., Zhang, D. (Eds.), Mobile Social Networking SE - 1, Computational Social Sciences. Springer New York, pp. 1–7. doi:10.1007/978-1-4614-8579-7_1

Chung, W., 2006. Dynamic social network analysis. Dartmouth College.

Chung, W., Savell, R., Schütt, J.-P., Cybenko, G., 2006. Identifying and Tracking Dynamic Processes in Social Networks, in: Proc. SPIE 6201, Sensors, and Command, Control, Communications, and Intelligence (C3I) Technologies for Homeland Security and Homeland Defense V, 620105. pp. 1–12. doi:10.1117/12.670127

Coleman, J., Katz, E., Menzel, H., 1957. The diffusion of an innovation among physicians. Sociometry 20, 253–270. doi:10.2307/2785979

Conti, M., Das, S.K., Bisdikian, C., Kumar, M., Ni, L.M., Passarella, A., Roussos, G., Tröster, G., Tsudik, G., Zambonelli, F., 2012. Looking ahead in pervasive computing: Challenges and opportunities in the era of cyber–physical convergence. Pervasive and Mobile Computing 8, 2–21. doi:10.1016/j.pmcj.2011.10.001

De Silva, V., Xiyu Shi, Evans, B., Trufero, J.P., Peters, G., 2013. Designing a QoE aware intelligent home gateway for future smart homes.

Divitini, M., Farshchian, B.A., Samset, H., 2004. UbiCollab: collaboration support for mobile users, in: Proceedings of the 2004 ACM Symposium on Applied Computing SAC 04. p. 1191. doi:10.1145/967900.968141

Doolin, K., Roussaki, I., Roddy, M., Kalatzis, N., Papadopoulou, E., Taylor, N., Liampotis, N., McKitterick, D., Jennings, E., Kosmides, P., 2012. SOCIETIES: where pervasive meets social 30–41.

Doolin, K., Taylor, N., Crotty, M., Roddy, M., Jennings, E., Roussaki, I., McKitterick, D., 2014. Enhancing Mobile Social Networks with Ambient Intelligence, in: Chin, A., Zhang, D. (Eds.), Mobile Social Networking SE - 7, Computational Social Sciences. Springer New York, New York, NY, pp. 139–163. doi:10.1007/978-1-4614-8579-7

Dourish, P., Edwards, W.K., Paul Dourish, W.K.E., 2000. A Tale of Two Toolkits: Relating Infrastructure and Use in Flexible CSCW Toolkits. Computer Supported Cooperative Work (CSCW) 9, 33–51. doi:10.1023/A:1008709725729

Edwards, W.K., 2005. Putting computing in context. ACM Transactions on Computer-Human Interaction 12, 446–474. doi:10.1145/1121112.1121117

Edwards, W.K., 1994. Session management for collaborative applications, in: Proceedings of the 1994 ACM Conference on Computer Supported Cooperative Work - CSCW ’94. ACM Press, New York, New York, USA, pp. 323–330. doi:10.1145/192844.193038

Edwards, W.K., Bellotti, V., Dey, A.K., Newman, M.W., Dey, A.K., 2003. The challenges of user-centered design and evaluation for infrastructure, in: Proceedings of the Conference on Human Factors in Computing Systems - CHI ’03. p. 297. doi:10.1145/642611.642664

Ellis, C., Wainer, J., 1994. A conceptual model of groupware, in: Proceedings of the 1994 ACM Conference on Computer Supported Cooperative Work - CSCW ’94. ACM Press, New York, New York, USA, pp. 79–88. doi:10.1145/192844.192878

Ellis, C.A., Gibbs, S.J., Rein, G., 1991. Groupware: some issues and experiences. Communications of the ACM 34, 39–58. doi:10.1145/99977.99987

Erickson, T., 2013. Social Computing, in: Dam, M.S. and R.F. (Ed.), The Encyclopedia of Human-Computer Interaction, 2nd Ed. The Interaction Design Foundation, Aarhus, Denmark.

Farshchian, B., Divitini, M., 2010. Collaboration Support for Mobile Users in Ubiquitous Environments, in: Handbook of Ambient Intelligence and Smart Environments SE - 7. pp. 173–199. doi:10.1007/978-0-387-93808-0_7

Fischer, G., 2001. Communities of interest: Learning through the interaction of multiple knowledge systems. Proceedings of the 24th IRIS Conference 1–14.

Fitzpatrick, G., Ellingsen, G., 2012. A Review of 25 Years of CSCW Research in Healthcare: Contributions, Challenges and Future Agendas. Computer Supported Cooperative Work (CSCW) 22, 609–665. doi:10.1007/s10606-012-9168-0

Floch, J., Angermann, M., 2012. Exploring Cooperating Smart Spaces for Efficient Collaboration in Disaster Management, in: ISCRAM. pp. 1–5.

Foell, S., Boessling, P., Linner, D., Radusch, I., Steglich, S., 2007. Self-Organizing Pervasive Online Communities. Eighth International Symposium on Autonomous Decentralized Systems (ISADS'07) 403–410. doi:10.1109/ISADS.2007.69

Forte, A., Dickard, M., Magee, R., Agosto, D.E., 2014. What Do Teens Ask Their Online Social Networks ? Social Search Practices among High School Students, in: CSCW. pp. 28–37. doi:10.1145/2531602.2531723

Fox, G., 2005. Global multimedia collaboration system, in: Proceedings of the 2005 International Symposium on Collaborative Technologies and Systems, 2005. IEEE, pp. 11–13. doi:10.1109/ISCST.2005.1553280

Franke, J., Cheng, B., 2013. Real-time privacy-preserving cobrowsing with element masking, in: 2013 17th International Conference on Intelligence in Next Generation Networks (ICIN). IEEE, pp. 86–93. doi:10.1109/ICIN.2013.6670898

Fuchs, C., Hofkirchner, W., Schafranek, M., Raffl, C., Sandoval, M., Bichler, R., 2010. Theoretical Foundations of the Web: Cognition, Communication, and Co-Operation. Towards an Understanding of Web 1.0, 2.0, 3.0. Future Internet. doi:10.3390/fi2010041

Fuks, H., Raposo, A., Gerosa, M., 2007. The 3C Collaboration Mode. Encyclopedia of E-Collaboration 637–644. doi:10.4018/978-1-59904-000-4

Fuks, H., Raposo, A., Gerosa, M.A., Pimentel, M., Filippo, D., Lucena, C., 2008. Inter- and intra-relationships between communication coordination and cooperation in the scope of the 3C Collaboration Model, in: 2008 12th International Conference on Computer Supported Cooperative Work in Design. IEEE, pp. 148–153. doi:10.1109/CSCWD.2008.4536971

Gava, V.L., Spinola, M.D.M., Tonini, A.C., Medina, J.C., 2012. the 3C Cooperation Model Applied To the Classical Requirement Analysis. Journal of Information Systems and Technology Management 9, 235–264. doi:10.4301/S1807-17752012000200003

Genco, A., Sorce, S., 2010. Pervasive Systems and Ubiquitous Computing. WIT Press.

Gilman, E., Davidyuk, O., Su, X., Riekki, J., 2013. Towards interactive smart spaces. Journal of Ambient Intelligence and Smart Environments 5, 5–22. doi:10.3233/AIS-120189

Goldberg, D., Nichols, D., Oki, B.M., Terry, D., 1992. Using collaborative filtering to weave an information tapestry. Communications of the ACM. doi:10.1145/138859.138867

Gomes, R.L., Rivera, G. de J., Willrich, R., Lima, C.V., Courtiat, J.-P., 2011. A Loosely Coupled Integration Environment for Collaborative Applications. IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans 41, 905–916. doi:10.1109/TSMCA.2011.2105866

Gomes, R.L., Rivera, G.J.H., Courtiat, J.P., 2005. Loosely-Coupled Integration of CSCW Systems, in: Distributed Applications and Interoperable Systems, Lecture Notes in Computer Science, Olume 3543. pp. 1128–1131.

Gower, J.C., 1971. A General Coefficient of Similarity and Some of Its Properties. Biometrics 27, 857–871. doi:10.2307/2528823

Granovetter, M., 1973. The Strength of Weak Ties. American journal of sociology 78, 1360–1380. doi:10.1037/a0018761

Grimm, R., Davis, J., Hendrickson, B., Lemar, E., MacBeth, A., Swanson, S., Anderson, T., Bershad, B., Borriello, G., Gribble, S., Wetherall, D., 2001. Systems directions for pervasive computing, in: Proceedings Eighth Workshop on Hot Topics in Operating Systems. IEEE Comput. Soc, pp. 147–151. doi:10.1109/HOTOS.2001.990075

Grimm, R., Wetherall, D., Davis, J., Lemar, E., Macbeth, A., Swanson, S., Anderson, T., Bershad, B., Borriello, G., Gribble, S., 2004. System support for pervasive applications. ACM Transactions on Computer Systems 22, 421–486. doi:10.1145/1035582.1035584

Groppe, J., Mueller, W., 2005. Profile Management Technology for Smart Customizations in Private Home Applications, in: 16th International Workshop on Database and Expert Systems Applications (DEXA'05). IEEE, pp. 226–230. doi:10.1109/DEXA.2005.156

Gross, T., Stary, C., Totter, A., 2005. User-Centered Awareness in Computer-Supported Cooperative Work-Systems: Structured Embedding of Findings from Social Sciences. International Journal of Human-Computer Interaction 18, 323–360. doi:10.1207/s15327590ijhc1803

Grudin, J., 1994. Computer-supported cooperative work: history and focus. Computer 27, 19–26. doi:10.1109/2.291294

Grudin, J., Poltrock, S., 2012. Taxonomy and Theory in Computer Supported Cooperative Work. The Oxford Handbook of Organizational Psychology 2, 1323–1348. doi:10.1093/oxfordhb/9780199928286.013.0040

Gunawardena, C.N., Hermans, M.B., Sanchez, D., Richmond, C., Bohley, M., Tuttle, R., Beth, M., 2010. A theoretical framework for building online communities of practice with social networking tools. Educational Media International 46, 3–16. doi:10.1080/09523980802588626

Gustavsen, R.M., 2002. Condor - an application framework for mobility- based context-aware applications, in: UBICOMP 2002.

Gutwin, C., Greenberg, S., 2002. A Descriptive Framework of Workspace Awareness for Real-Time Groupware. Computer Supported Cooperative Work (CSCW) 11, 411–446. doi:10.1023/A:1021271517844

Gutwin, C., Greenberg, S., Roseman, M., 1996. Workspace Awareness in Real-Time Distributed Groupware: Framework, Widgets, and Evaluation 281–298.

Guy, I., Jacovi, M., Perer, A., Ronen, I., Uziel, E., Carmel, M., 2010. Same Places , Same Things , Same People ? Mining User Similarity on Social Media. Cscw 2010 41–50. doi:10.1145/1718918.1718928

Haake, J., Hussein, T., Joop, B., Lukosch, S., Veiel, D., Ziegler, J., 2009. Context Modelling for Adaptive Collaboration.

Haake, J.M., Hussein, T., Joop, B., Lukosch, S., Veiel, D., Ziegler, J., 2010. Modeling And Exploiting Context For Adaptive Collaboration. International Journal of Cooperative Information Systems 19, 71–120. doi:10.1142/S0218843010002115

Hanneman, R.R.A., Riddle, M., 2005. Introduction to social network methods.

Harary, F., 1969. Graph theory. Addison-Wesley Pub. Co.

Henri, F., Pudelko, B., 2003. Understanding and analysing activity and learning in virtual communities. Journal of Computer Assisted Learning 19, 474–487. doi:10.1046/j.0266-4909.2003.00051.x

Hinssen, P.J.H., 1998. What Difference Does it Make? The Use of Groupware in Small Group, Raising Boys' Achievement in Schools. Telematica Instituut, The Netherlands, Netherlands.

Hong, J., Suh, E., Kim, S.-J., 2009. Context-aware systems: A literature review and classification. Expert Systems with Applications 36, 8509–8522. doi:10.1016/j.eswa.2008.10.071

Hussain, S.S., 2010. Integrating end-user support and negotiations to specify requirements for context-based adaptations in a collaboration environment, in: Proceedings of the 2nd ACM SIGCHI Symposium on Engineering Interactive Computing Systems - EICS '10. ACM Press, New York, New York, USA, p. 329. doi:10.1145/1822018.1822073

Jennings, E., 2011. SOCIETIES Public Deliverable 2.1 - Specification of Initial Scenarios and User Requirements.

Jennings, E., Roddy, M., Leckey, A., Feigenblat, G., 2014. Evaluation Of Early Prototypes For Social Mobile And Pervasive Computing With Scripted Role-Play, in: 8th Annual Irish Human Computer Interaction (iHCI). Dublin.

Jesus Hoyos-Rivera, G., Gomes, R.L., Willrich, R., Courtiat, J.-P., 2006. CoLab: A New Paradigm and Tool for Collaboratively Browsing the Web. IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans 36, 1074–1085. doi:10.1109/TSMCA.2006.883173

Johansen, R., 1988. GroupWare: Computer Support for Business Teams. The Free Press New York, NY, USA.

Johnson, D., 2013. Mobile Support in CSCW Applications and Groupware Development Frameworks. International Journal of Interactive Mobile Technologies (iJIM) 7, 54–62. doi:10.3991/ijim.v7i2.2469

Joung, Y.-J., Chien, P.-H., 2008. P2PConf: A Medium-Size P2P Internet Conference with Effective Floor Control, 2008 International Conference on Information Networking. doi:10.1109/ICOIN.2008.4472792

Kalatzis, N., Liampotis, N., Roussaki, I., Kosmides, P., Papaioannou, I., Xynogalas, S., 2013. Community context management research and challenges in pervasive and social computing, in: 2013 IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOM Workshops). IEEE, pp. 109–114. doi:10.1109/PerComW.2013.6529466

Kalatzis, N., Liampotis, N., Roussaki, I., Kosmides, P., Papaioannou, I., Xynogalas, S., Zhang, D., Anagnostou, M., 2014. Cross-community context management in Cooperating Smart Spaces. Personal and Ubiquitous Computing 18, 427–443. doi:10.1007/s00779-013-0654-2

Karat, J., Vanderdonckt, J., Barbosa, S., 2009. Awareness systems: Advances in theory, methodology and design. Springer London. doi:10.1007/978-1-84882-477-5

Katz, N., Lazer, D., Arrow, H., Contractor, N., 2004. Network Theory and Small Groups. Small Group Research 35, 307–332. doi:10.1177/1046496404264941

Kernchen, R., Bonnefoy, D., Battestini, A., Mrohs, B., Wagner, M., Klemettinen, M., 2006. Context-Awareness in MobiLife, in: IST Mobile Summit 2006.

Kim, K., Fox, G.C., 2011. Modeling, simulation, and practice of floor control for synchronous and ubiquitous collaboration. Multimedia Tools and Applications 53, 213–236. doi:10.1007/s11042-010-0508-0

Kim, M., Jung, Y., Lee, J., Kim, M., 2006. Ubiquitous Computing Systems, Lecture Notes in Computer Science. Springer Berlin Heidelberg, Berlin, Heidelberg. doi:10.1007/11890348

Knappmeyer, M., Baker, N., Liaquat, S., Tönjes, R., 2009. A Context Provisioning Framework to Support Pervasive and Ubiquitous Applications, in: Barnaghi, P., Moessner, K., Presser, M., Meissner, S. (Eds.), Smart Sensing and Context, Lecture

Notes in Computer Science. Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 93–106. doi:10.1007/978-3-642-04471-7_8

Koch, M., 2008. CSCW and Enterprise 2.0 - Towards an Integrated Perspective, in: Proceedings of the 21st Bled eConference: eCollaboration: Overcoming Boundaries Through Multi-Channel Interaction. pp. 416–427. doi:10.1111/j.1365-2656.2008.01372.x

Koch, M., Ott, F., Richter, A., 2014. Socio-technically Integrated Access to Virtual Communities with CommunityMirrors, in: Leimeister, J.M., Rajagopolan, B. (Eds.), Virtual Communities. M. E. Sharpe, pp. 145–161.

Koltay, T., 2010. The Web 2.0 Contradiction: Commercial and Library Use, Library Philosophy and Practice (e-journal). University of Idaho Library.

Kossinets, G., Kleinberg, J., Watts, D., 2008. The structure of information pathways in a social communication network, in: Proceeding of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining - KDD 08. ACM Press, New York, New York, USA, p. 435. doi:10.1145/1401890.1401945

Kulkarni, D., Ahmed, T., Tripathi, A., 2012. A Generative Programming Framework for Context-Aware CSCW Applications. ACM Transactions on Software Engineering and Methodology 21, 1–35. doi:10.1145/2089116.2089121

Kulyk, O., 2010. Do you know what I know? : situational awareness of co-located teams in multidisplay environments. University of Twente, Enschede, The Netherlands. doi:10.3990/1.9789036529389

Kulyk, O., van der Veer, G., van Dijk, B., 2008. Situational awareness support to enhance teamwork in collaborative environments. Proceedings of the 15th European conference on Cognitive ergonomics the ergonomics of cool interaction - ECCE ’08 1. doi:10.1145/1473018.1473025

Kumar, M., Shirazi, B.A., Das, S.K., Sung, B.B.Y., Levine, D., Singhal, M., 2003. PICO: a middleware framework for pervasive computing. IEEE Pervasive Computing 2, 72–79. doi:10.1109/MPRV.2003.1228529

Lave, J., Wenger, E., 1991. Situated Learning: Legitimate Peripheral Participation. Cambridge University Press.

Lazarsfeld, P.F., Merton, R.K., 1954. Friendship as a Social Process: A Substantive and Methodological analysis. Freedom and Control in Modern Society 18, 18–66. doi:10.1111/j.1467-8705.2012.02056_3.x

Lerman, K., 2007. Social Information Processing in News Aggregation. IEEE Internet Computing 11, 16–28. doi:10.1109/MIC.2007.136

Li, Q., Liu, S., Pan, Y., 2012. A cooperative construction approach for SaaS applications, in: Proceedings of the 2012 IEEE 16th International Conference on Computer

Supported Cooperative Work in Design, CSCWD 2012. pp. 398–403. doi:10.1109/CSCWD.2012.6221849

Lima, C., Antunes, M., Gomes, D., Aguiar, R., 2013. Social Awareness in Pervasive Communities for Collaborative Work, in: Botía, J.A., Charitos, D. (Eds.), Intelligent Environments Workshops, Volume 17 of Ambient Intelligence and Smart Environments. IOS Press, Athens, Greece, pp. 110 – 115. doi:10.3233/978-1-61499-286-8-110

Lima, C., Antunes, M., Gomes, D., Aguiar, R., Mota, T., 2014. A Context-Aware Framework for Collaborative Activities in Pervasive Communities. International Journal of Distributed Systems and Technologies 5, 31–43. doi:10.4018/ijdst.2014040103

Lima, C., Gomes, D., Aguiar, R., 2012. Pervasive CSCW for smart spaces communities, in: 2012 IEEE International Conference on Pervasive Computing and Communications Workshops. IEEE, pp. 118–123. doi:10.1109/PerComW.2012.6197462

Lima, C.V., Antunes, M., Gomes, D., Aguiar, R., 2013. A context-aware framework for CSCW applications in enterprise environments, in: Proceedings of the IADIS International Conference Web Based Communities and Social Media 2013, Proceedings of the IADIS International Conference Collaborative Technologies 2013. pp. 11–18.

Lin, Y.-R., Chi, Y., Zhu, S., Sundaram, H., Tseng, B.L., 2009. Analyzing communities and their evolutions in dynamic social networks. ACM Transactions on Knowledge Discovery from Data 3, 1–31. doi:10.1145/1514888.1514891

Lin, Y.-R., Chi, Y., Zhu, S., Sundaram, H., Tseng, B.L., 2008. FacetNet: A Framework for Analyzing Communities and Their Evolutions in Dynamic Networks. Proceeding of the 17th international conference on World Wide Web - WWW '08 685. doi:10.1145/1367497.1367590

Locatelli, M., Loregian, M., Vizzari, G., 2008. An Agent Model for Collaborative Ubiquitous-Computing Environments. AT2AI-6 Working Notes, From Agent Theory to Agent Implementation, 6th Int. Workshop.

López-Jaquero, V., Vanderdonckt, J., Montero, F., González, P., 2008. Towards an Extended Model of User Interface Adaptation : The I SATINE Framework, in: Engineering Interactive Systems. pp. 374–392. doi:10.1007/978-3-540-92698-6_23

Lugano, G., 2012. Social Computing: A Classification of Existing Paradigms, in: 2012 International Conference on Privacy, Security, Risk and Trust and 2012 International Confernece on Social Computing. IEEE, pp. 377–382. doi:10.1109/SocialCom-PASSAT.2012.54

Lugano, G., 2010. Digital community design: exploring the role of mobile social software in the process of digital convergence. University of Jyväskylä.

Lugano, G., 2007. Mobile social software: definition, scope and applications, in: eChallenges2007. pp. 1434–1441.

Markopoulos, P., 2009. A Design Framework for Awareness Systems, in: Markopoulos, P., De Ruyter, B., Mackay, W. (Eds.), Awareness Systems SE - 2, Human-Computer Interaction Series. Springer London, pp. 49–72. doi:10.1007/978-1-84882-477-5_2

McFaddin, S., Coffman, D., Han, J.H., Jang, H.K., Kim, J.H., Lee, J.K., Lee, M.C., Moon, Y.S., Narayanaswami, C., Paik, Y.S., Park, J.W., Soroker, D., 2008. Modeling and Managing Mobile Commerce Spaces Using RESTful Data Services. The Ninth International Conference on Mobile Data Management (mdm 2008) 81–89. doi:10.1109/MDM.2008.38

Mena, T. Ben, Saoud, N.B.-B., Ahmed, M. Ben, Pavard, B., 2007. Towards a Methodology for Context Sensitive Systems Development. Modeling and Using Context 56–68. doi:10.1007/978-3-540-74255-5_5

Meyer, E.M., Wichmann, D., Büsch, H., Boll, S., 2011. Supporting Mobile Collaboration in Spatially Distributed Workgroups with Digital Interactive Maps. Mobile Networks and Applications 17, 365–375. doi:10.1007/s11036-010-0288-9

Monge, P.R., Contractor, N.S., 2003. Theories of Communication Networks. Oxford University Press.

Mørch, A., 1997. Three levels of end-user tailoring: customization, integration, and extension. Computers and design in context 51–76.

Muller, M., Ehrlich, K., Matthews, T., Perer, A., Ronen, I., Guy, I., Street, O.R., Road, H., Jose, S., Carmel, M., 2012. Diversity among Enterprise Online Communities: Collaborating , Teaming , and Innovating through Social Media. Proceedings of the 2012 ACM annual conference on Human Factors in Computing Systems - CHI '12 2815–2824. doi:10.1145/2207676.2208685

Narayanan, A., Shmatikov, V., 2009. De-anonymizing Social Networks, in: 2009 30th IEEE Symposium on Security and Privacy. IEEE, pp. 173–187. doi:10.1109/SP.2009.22

Nielsen, J., 1996. Multimedia and Hypertext: The Internet and Beyond. Academic Press, Boston, MA.

Nino, C., Barbosa, J., Pinto, S.C.C.S., Barbosa, D., 2009. Enhancing Collaboration in a Scenario of Ubiquitous Computing, in: 2009 Simposio Brasileiro de Sistemas Colaborativos. Ieee, pp. 44–52. doi:10.1109/SBSC.2009.24

Oulasvirta, A., 2009. Social Inference Through Technology, in: Markopoulos, P., De Ruyter, B., Mackay, W. (Eds.), Awareness Systems SE - 5, Human-Computer Interaction Series. Springer London, pp. 125–147. doi:10.1007/978-1-84882-477-5_5

Papadopoulou, E., 2014. SOCIETIES Public Deliverable D5.8 - Final Report - CSS Individual and Community Experience.

Papadopoulou, E., Gallacher, S., Taylor, N., 2014. Combining Pervasive Computing With Social Networking for a Student Environment 11–19.

Pentland, A., 2005. Socially aware computation and communication. Computer 38, 33–40. doi:10.1109/MC.2005.104

Pichiliani, M.C., Hirata, C.M., 2009. A technical comparison of the existing approaches to support collaboration in non-collaborative applications, in: 2009 International Symposium on Collaborative Technologies and Systems. IEEE, pp. 314–321. doi:10.1109/CTS.2009.5067496

Poltrock, S., Grudin, J., 1999. CSCW, groupware and workflow: experiences, state of art, and future trends, in: CHI '99: CHI '99 Extended Abstracts on Human Factors in Computing Systems. pp. 120–121. doi:http://doi.acm.org/10.1145/632716.632791

Prehofer, C., van Gurp, J., di Flora, C., 2007. Towards the web as a platform for ubiquitous applications in smart spaces, in: Second Workshop on Requirements and Solutions for Pervasive Software Infrastructures (RSPSI), at Ubicomp.

Prehofer, C., van Gurp, J., Stirbu, V., Satish, S., Tarkoma, S., di Flora, C., Liimatainen, P.P., 2010. Practical Web-Based Smart Spaces. IEEE Pervasive Computing 9, 72–80. doi:10.1109/MPRV.2009.88

Prekop, P., Burnett, M., 2003. Activities, context and ubiquitous computing. Computer Communications 26, 1168–1176. doi:10.1016/S0140-3664(02)00251-7

Prell, C., 2006. Social Capital as Network Capital: Looking at the Role of Social Networks Among Not-For-Profits. Sociological Research Online 11.

Putnam, R.D., 2001. Bowling Alone: The Collapse and Revival of American Community, 1st ed. Touchstone Books by Simon & Schuster, New York, New York, USA.

Raychoudhury, V., Cao, J., Kumar, M., Zhang, D., 2013. Middleware for pervasive computing: A survey. Pervasive and Mobile Computing 9, 177–200. doi:10.1016/j.pmcj.2012.08.006

Rheingold, H., 1993. The Virtual Community: Homesteading on the Electronic Frontier, Book. doi:10.1561/1500000001

Ridler, T., Calvard, S., 1978. Picture Thresholding Using an Iterative Selection Methods. IEEE Transaction on Systems, Man and Cybernetics 8, 630–632. doi:10.1109/TSMC.1978.4310039

Rittenbruch, M., Viller, S., Mansfield, T., 2007. Announcing activity: design and evaluation of an intentionally enriched awareness service. Human-Computer Interaction 22, 137–171. doi:10.1080/07370020701307930

Roddy, M., 2014. SOCIETIES Public Deliverable 8.9 - Final Evaluation Report.

Román, M., Hess, C., Cerqueira, R., Ranganathan, A., Campbell, R.H., Nahrstedt, K., 2002. Gaia : A Middleware Platform for Active Spaces. ACM SIGMOBILE Mobile Computing and Communications Review 6, 65–67. doi:10.1145/643550.643558

Roussaki, I., 2011. SOCIETIES Public Deliverable 2.2 - Scenario Description, Use Cases and Technical Requirements specification.

Sakurai, Y., Gonzalez, A.J., Nguyen, J., Takada, K., Uchida, K., Tsuruta, S., 2008. Web-Based Intelligent CSCW Exploiting Context-Based Reasoning, in: Signal Image Technology and Internet Based Systems, 2008. SITIS '08. IEEE International Conference on. pp. 490–497. doi:10.1109/SITIS.2008.95

Sandkuhl, K., Borchardt, U., 2014. Perspectives in Business Informatics Research, International Conference on Business Informatics Research, Lecture Notes in Business Information Processing. Springer International Publishing, Cham. doi:10.1007/978-3-319-11370-8

Santos, R.O., Oliveira, F.F., Gomes, R.L., Martinello, M., Guizzardi, R.S.S., 2011. LiCoB: Lightweight Collaborative Browsing. International Journal of Web Portals 3, 17–32. doi:http://dx.doi.org/10.4018/jwp.2011010102

Schilit, B., Adams, N., Want, R., 1994. Context-aware computing applications. Workshop on Mobile Computing Systems and Applications 85–90. doi:10.1109/MCSA.1994.512740

Schjørring, J.W., 2004. From context-adaptive applications to context-aware actors: Context-aware computing in a CSCW perspective. University of Copenhagen.

Schmidt, A., Gross, T., Billinghurst, M., 2004. Introduction to special issue on Context-Aware Computing in CSCW. Computer Supported Cooperative Work (CSCW) 221–222.

Schneider, D., Moraest, K., Souzat, J.M. De, Esteves, M.G.P., Moraes, K., de Souza, J.M., 2012. CSCWD: Five characters in search of crowds, in: Proceedings of the 2012 IEEE 16th International Conference on Computer Supported Cooperative Work in Design (CSCWD). IEEE, pp. 634–641. doi:10.1109/CSCWD.2012.6221885

Schreurs, B., de Laat, M., 2014. The Network Awareness Tool: A web 2.0 tool to visualize informal networked learning in organizations. Computers in Human Behavior. doi:10.1016/j.chb.2014.04.034

Schuster, D., Rosi, A., Mamei, M., Springer, T., Endler, M., Zambonelli, F., 2013. Pervasive social context. ACM Transactions on Intelligent Systems and Technology 4, 1. doi:10.1145/2483669.2483679

Shirky, C., 2005. A Group is Its Own Worst Enemy, in: The Best Software Writing I. pp. 183–209. doi:10.1007/978-1-4302-0038-3_23

Stahl, G., 2013. Theories of Collaborative Cognition: Foundations for CSCL and CSCW Together. Computer-Supported Collaborative Learning 1–20.

Stahl, G., 2011. Theories of cognition in CSCW. ECSCW 2011: Proceedings of the 12th European Conference on Computer Supported Cooperative Work 24–28. doi:10.1007/978-0-85729-913-0_11

Steinmacher, I., Chaves, A.P., Gerosa, M.A., 2012. Awareness Support in Distributed Software Development: A Systematic Review and Mapping of the Literature. Computer Supported Cooperative Work (CSCW) 22, 113–158. doi:10.1007/s10606-012-9164-4

Sutcliffe, A., 2005. Extending Small Group Theory for Analysing Complex Systems, in: Science, E.J.C.W.D. of computer (Ed.), Proceedings of 2nd Workshop of Complexity and Design and Engineering. GIST report 2005-1, Glasgow, pp. 139–148.

Tantipathananandh, C., Berger-Wolf, T., Kempe, D., 2007. A framework for community identification in dynamic social networks. International Conference on Knowledge Discovery and Data Mining 717. doi:10.1145/1281192.1281269

Tao Gu, Pung, H.K., Da Qing Zhang, 2004. A middleware for building context-aware mobile services, in: 2004 IEEE 59th Vehicular Technology Conference. VTC 2004-Spring (IEEE Cat. No.04CH37514). IEEE, pp. 2656–2660. doi:10.1109/VETECS.2004.1391402

Taylor, N., 2013. SOCIETIES Public Deliverable 5.7 - Second Integrated Prototype - CSS Individual and Community Experience.

Taylor, N., 2008. Personal eSpace and personal smart spaces. Proceedings - 2nd IEEE International Conference on Self-Adaptive and Self-Organizing Systems Workshops, SASOW 2008 156–161. doi:10.1109/SASOW.2008.23

Taylor, N., Robertson, P., Farshchian, B., Doolin, K., Roussaki, I., Marshall, L., Mullins, R., Druesedow, S., Dolinar, K., 2011. Pervasive Computing in Daidalos. IEEE Pervasive Computing 10, 74–81. doi:10.1109/MPRV.2010.24

Teknomo, K., 2014. Similarity Measurement [WWW Document]. URL http://people.revoledu.com/kardi/tutorial/Similarity/ (accessed 3.1.13).

Terken, J., Subramanian, S., Zancanaro, M., 2007. Special issue on user-centred design and evaluation of ubiquitous groupware. Personal and Ubiquitous Computing 12, 179–180. doi:10.1007/s00779-007-0143-6

Tripathi, A.R., Kulkarni, D., Ahmed, T., 2005. A specification model for context-based collaborative applications. Pervasive and Mobile Computing 1, 21–42. doi:10.1016/j.pmcj.2005.01.005

Valente, T.W., 1995. Network models of the diffusion of innovations, Computational and Mathematical Organization Theory. Hampton Press (NJ). doi:10.1007/BF00240425

Velozo, J., Sagara, J., 2010. Social Relations as the Key for Design Team Effectiveness. First International Conference on Design Creativity 1–6.

Venezia, C., Taylor, N., Williams, H., Doolin, K., Roussaki, I., 2009. Novel Pervasive Computing Services Experienced through Personal Smart Spaces. 2009 Tenth International Conference on Mobile Data Management: Systems, Services and Middleware 484–489. doi:10.1109/MDM.2009.85

Wang, B., Bodily, J., Gupta, S.K.S., 2004. Supporting Persistent Social Groups in Ubiquitous Computing Environments Using Context-Aware Ephemeral Group Service, in: PerCom '04. pp. 287–296. doi:10.1109/percom.2004.1276866

Wang, F.-Y., Carley, K.M., Zeng, D., Mao, W., 2007. Social Computing: From Social Informatics to Social Intelligence. Intelligent Systems, IEEE 22, 79–83. doi:10.1109/MIS.2007.41

Wang, Z., Zhou, X., Yu, Z., Wang, H., Ni, H., 2010. Quantitative Evaluation Of Group User Experience In Smart Spaces. Cybernetics and Systems 41, 105–122. doi:10.1080/01969720903584209

Want, R., 2010. An Introduction to Ubiquitous Computing, Ubiquitous Computing Fundamentals. Chapman and Hall/CRC.

Weiser, M., 1991. The Computer for the 21st Century. Scientific American 265, 94–104. doi:10.1145/329124.329126

Wenger, E., 1998. Communities in practice: Learning, meaning, and identity. Cambridge University Press;

Wenger, E., McDermott, R., Snyder, M., 2002. Cultivating Communities of Practice: A Guide to managing Knowledge - Seven Principles for Cultivating Communities of Practice, Working Knowledge for Bussiness Leaders. Harvard Business School Press.

Wood, L., Buscher, M., van Veelen, B., van Splunter, S., 2012. Agile Response and Collaborative Agile Workflows, in: 2012 IEEE 21st International Workshop on Enabling Technologies: Infrastructure for Collaborative Enterprises. IEEE, pp. 358–363. doi:10.1109/WETICE.2012.89

Wu, A., DiMicco, J.M., Millen, D.R., 2010. Detecting professional versus personal closeness using an enterprise social network site. Proceedings of the 28th international conference on Human factors in computing systems - CHI ’10 1955. doi:10.1145/1753326.1753622

Yeung, C.A., Liccardi, I., Lu, K., Seneviratne, O., Berners-Lee, T., 2009. Decentralization: The future of online social networking. W3C Workshop on the Future of Social Networking Position Papers 2, 2–7.

Youngblood, G.M., Holder, L.B., Cook, D.J., 2005. Managing Adaptive Versatile Environments. Third IEEE International Conference on Pervasive Computing and Communications 351–360. doi:10.1109/PERCOM.2005.23

Yu, P., Ma, X., Cao, J., Lu, J., 2013. Application mobility in pervasive computing: A survey. Pervasive and Mobile Computing 9, 2–17. doi:10.1016/j.pmcj.2012.07.009

Yu, Z., Zhang, D., Eagle, N., Cook, D., 2013. Introduction to the special section on intelligent systems for socially aware computing. ACM Transactions on Intelligent Systems and Technology 4, 1. doi:10.1145/2483669.2483678

Yu, Z., Zhou, X., 2014. Socially Aware Computing: Concepts, Technologies, and Practices, Mobile Social Networking. Springer New York, New York, NY. doi:10.1007/978-1-4614-8579-7

Zacarias, M., Pinto, H.S., Magalhães, R., Tribolet, J., 2010. A “context-aware” and agent-centric perspective for the alignment between individuals and organizations. Information Systems 35, 441–466. doi:10.1016/j.is.2009.03.014

Zafar, M., Baker, N., Moltchanov, B., 2009. Context management architecture for future internet services. Applied Sciences.

Ziebart, B.D., Roth, D., Campbell, R.H., Dey, A.K., 2005. Learning Automation Policies for Pervasive Computing Environments, in: Second International Conference on Autonomic Computing (ICAC’05). IEEE, pp. 193–203. doi:10.1109/ICAC.2005.37

9 Appendix A: Pre-Questionnaire

Q.1 Please inform your email

____

Q.2 Please inform your gender

- Male
- Female

Q.3 About the second stage of the trial: Do you think that you were invited to the appropriate chat rooms?

- Aquatic Ecotoxicology
- Terrestrial Ecotoxicology
- Aquatic Invertebrates
- Terrestrial Invertebrates
- Oxidative Stress Biomarkers
- Bioaccumulation and Biomagnification
- Metabolomics
- Proteomics
- Nanoparticles
- Pesticides
- Metals
- R Statistical Software
- IBM SPSS Statistics
- Minitab Statistics Package
- Other (specify):

Q.4 In the grid below, please list the people you have the same routine work in terms of class of compounds, organisms and methodologies.

	Compound Classes	Organisms	Methodologies	None
P1				
P2				
P3				
P4				
P5				
P6				
P7				
P8				
P9				
P10				
P11				

Q.5 Which of the following people you consider working together on a daily basis?

- Note 1: Choose more than one person per line is possible.
- Note2: Disregard your own name.

	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	P11
P1											
P2											
P3											
P4											
P5											
P6											
P7											
P8											
P9											
P10											
P11											

Q.6 Please indicate your degree of similarity to the following people based on their professional interests. E.g. If you work with the same organism, same kind of test/methodology and the same compound, assign a higher value. If you are working with the same organisms, same methodology but different compounds; assign a lower value and so on).

	Very Low	Low	Medium	High	Very High
P1					
P2					
P3					
P4					
P5					
P6					
P7					
P8					
P9					
P10					
P11					

Q.7 When social network requests information related to your profession you usually provide? E.g. Skills, specialties, professional interests.

Never	Rarely	Sometimes	Often	Very Often

Q.8 If you use a social network site, please indicate below

- LinkedIn
- Research Gate
- Facebook
- Google +
- Twitter
- Foursquare

10 APPENDIX B: POST-QUESTIONNAIRE

Q.1 Write the username that you used in the trial

____

Q.2 About the first stage of the trial: Do you think that you were invited to the appropriate chat rooms?

- Yes
- No

Q.3 About the second stage of the trial: Do you think that you were invited to the appropriate chat rooms?

- Yes
- No

Q.4 Regarding both stages of the trial: which one do you think is most suitable for your interests?

- Stage 1
- Stage 2
- Both are the same for me

Q.5 By using this type of applications where is based on your professional information, do you fear for your privacy?

	1	2	3	4	5	
Yes, I do not trust my information						No, I totally trust

Q.6 Would you find useful an application that suggests you to participate in discussion groups with common interests?

	1	2	3	4	5	6	7	8	9	10	
Not useful											Very useful

Q.7 Please feel free to give your opinion/suggestions

____

11 APPENDIX C: UML CLASS DIAGRAMS

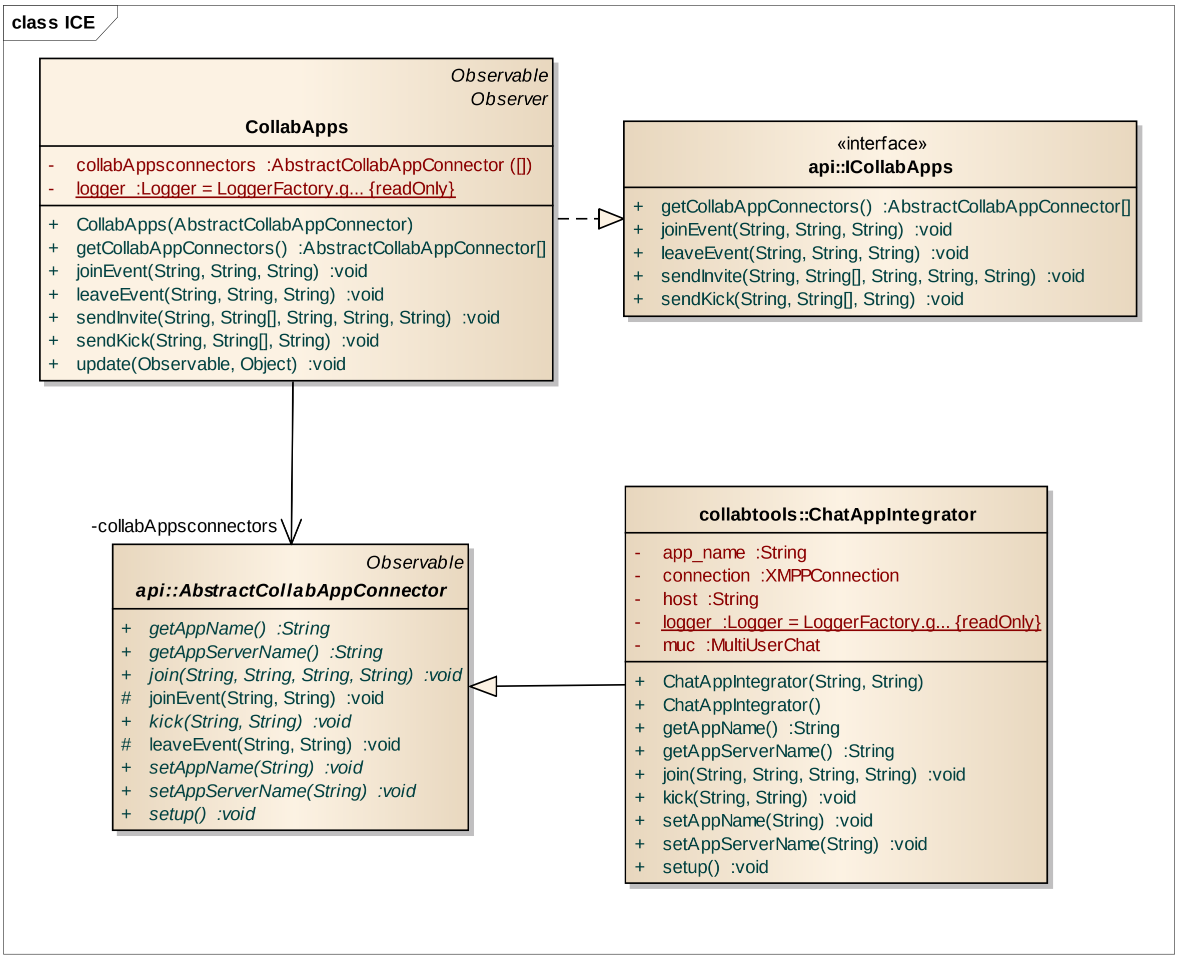

Figure 11.1: Collaborative applications integration class diagram.

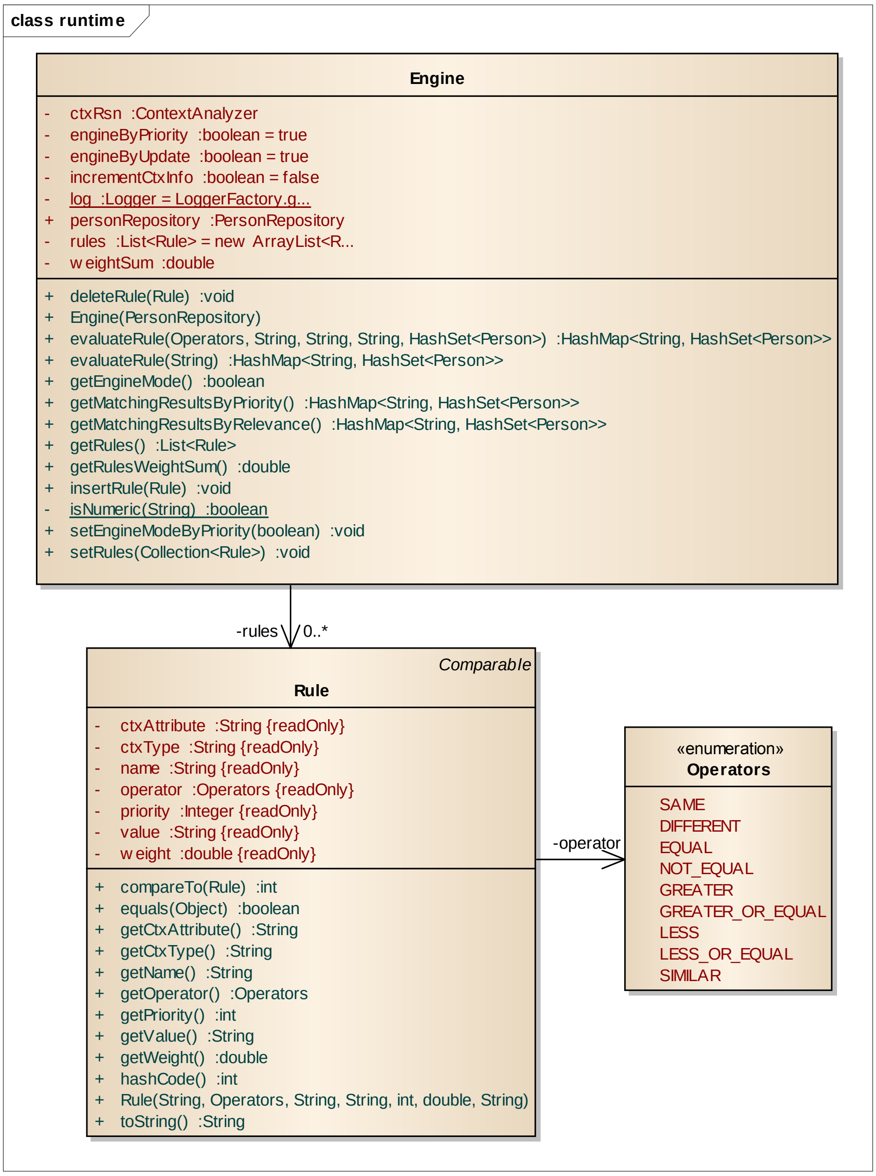

Figure 11.2: Rule engine and rule class diagrams.

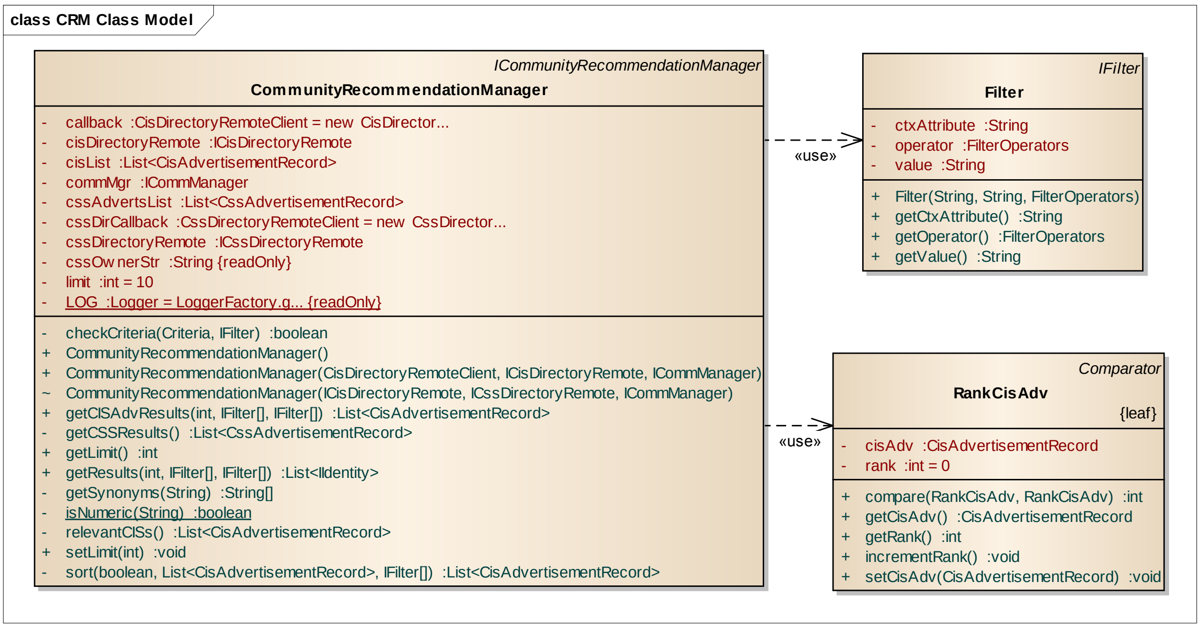

Figure 11.3: Community Recommendation Manager Class diagram.