A CONTEXT-AWARE FRAMEWORK FOR CSCW APPLICATIONS IN ENTERPRISE ENVIRONMENTS

Christopher Viana Lima, Mário Antunes, Diogo Gomes and Rui Aguiar
University of Aveiro. Instituto de Telecomunicações – IT - Campus Universitário de Santiago, Aveiro, Portugal.

ABSTRACT

Future pervasive environments will take into consideration physical and digital social relationships. Nowadays it is important to use collective intelligence, where the interpretation of context information can be harnessed as input for context-aware applications, especially for group collaboration. For collaborative applications this represents opportunities, but also new challenges in terms of using collective information for adaptability and personalization in pervasive environments. This paper presents the challenges in design and development of a context-aware framework CSCW supporting pro-behaviour capabilities in pervasive communities.

KEYWORDS

Pervasive computing, social computing, cooperation, CSCW.

1. INTRODUCTION

Much effort has been dedicated to design collaborative applications that support groups in synchronous collaboration (e.g. audio conference, shared screens), and asynchronous collaboration (e.g. wiki, blog). These efforts generally aim to create Integrated Collaborative Environments (ICE) to promote collaboration by providing mutual awareness, communication and coordination (Farshchian 2010).

(Grudin 1988) raised problems in development and success of CSCW (Computer Supported Cooperative Work) applications to a critical masses of users, in which systems benefits only a few users and not the entire collective. The context information available in ubiquitous and pervasive environments can be used to predict needs, personalize and negotiate services anywhere and anytime. Context can be any relevant information available about location, services, devices and their relationships (Dey and Abowd 2000).

An important aspect of pervasive environments is the integration of mobile devices and sensors with the environment that surrounds the user. The migration among smart environments should be done as naturally as possible, seamlessly adapting the user's application. In today's scenarios, pervasive systems lack the ability to deal with social group context, besides the individual. (Ackerman 2000) highlighted that human computer interaction and CSCW systems need to understand how people really work and live in groups, organizations, and other forms of collective life. Thus, social networks provide opportunity to access detailed contextual information that is hard to achieve in other ways.

Collective intelligence takes an important role, having influenced the adoption of Web 2.0 tools inside companies and organizations. A questionnaire, which the first author helped to develop the storyboards, was conducted at Intel's Campus, Ireland in 2011. The results demonstrate that almost all participants agreed to share their professional information such as: business cards, LinkedIn profile, educational information and expertise areas.

The use of context information offer several possibilities to enhance user experience especially in cooperative tasks. In order to investigate the benefits of CSCW, context-aware groups and social paradigms, we propose a context-aware framework for collaborative work, which envisages cooperation in pervasive environments. The goals of the framework are:

- Provide adaptability for communication, coordination and relationships to CSCW in enterprise scenarios.
- Facilitate collaborative work by providing pro-active behaviour for CSCW tools.

- Enhance collaboration work taking into account several aspects, such as mobility of people and physical distribution.
- Provide cooperation among users based on physical and social interaction

The remainder of this paper is organized as follows: Section 2 presents a background regarding the integration of CSCW with pervasive environments, context-aware group and social computing. Section 3 discusses the related work regarding the involved areas. Section 4 introduces the framework architecture. Section 5 presents the framework implementation. Finally, Section 6 provides a conclusion and directions for future work.

2. BACKGROUND

One of the most important issues for pervasive systems is to create an infrastructure to support several users in different environments. Smart spaces are context aware in nature, where location/proximity (e.g. home, office), activities (e.g. walking, driving) and environment features (e.g. hot, cold) enabling users to interact with computational resources to perform their activities. On other hand, personal smart spaces introduce the concept of integrating smart spaces and mobile systems in the same platform system (Taylor 2011).

Furthermore, a natural step in this direction is to share resources and social information to support more social aspects among users.

2.1 Pervasive Communities

SOCIETIES project introduced the notion of a Cooperating Smart Space (CSS) to define the merging of social computing and pervasive computing. Each CSS consists 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 CSSs members of the same communities.

There are many criteria that can be used to form a community of individuals or organizations, such as: geographic location, similar preferences, common features or interests (personal, business-related), common experiences, communication/interaction or authority/hierarchy and other forms of social relationships (Roussaki 2012).

The main focus of the platform is to facilitate creation, organisation and management of communities, providing better experience for individuals and the communities (Papadopoulou 2010). The SOCIETIES platform architecture supports mobile and/or desktop devices. Each component is responsible for a specific task, providing functionalities via APIs for internal components and third party services.

Besides the community formation, the platform provides a set of components that allow third party services to be built on top of it. Some of the essentials components are: device management, privacy and trust, context, community orchestration and personalization. The notion of community of interests can provide a different impact in the CSCW area. The CSCW can take advantage of pervasive communities in several scenarios as described in (Lima 2012), in which users with the same degree of similarities can be grouped into several sessions. Two scenarios examples that we emphasize are:

- E-learning (CSCL): university students, communities of courses teams
- Enterprise: task teams, departments

2.2 Context-Aware Applications

Usually pervasive platforms provide means to receive context information and process this information in order to build context-aware applications. Most of context management systems rely on a Context Broker architecture similar to CoBrA, as defined in (Taylor 2011).

A context management system 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 (Kalatzis 2011). Context information describes the users' environment where context-aware applications can decide how to behave based on this information. Below is presented some of the components usually found in context management systems. It is important to stress that some of them can slightly change or even be optional like the history management.

- Context Broker: This component is responsible for providing current, past and future estimates of context data to application consumers.
- Context DB Management: This component is responsible for the management of context information (i.e. retrieval, update, addition, removal).
- Context History Management: This component comprises a Database Management System (DBMS) for persistent storage of historical context data (Context History) for both individuals and communities.
- Context Source Management: This component is responsible for managing the external context sources available and for the collection of raw context data from the registered context sources, including Social Network Sites (SNS) and individual's location.

The use of context information can also benefit applications from a group perspective, particularly in person-to-person communications. The collaboration can be determined based on the needs of location, presence and availability information of the participants and media capabilities.

2.3 Social Computing Integration

While the majority of human interaction takes place in the physical world, there is steady increase of interpersonal interactions taking place in digital environments (Roussaki 2012).

Along with context information available in SNS, the context management can take benefit not only from physical world sensors, but from the digital world as well. SNSs can provide additional information about non-physical information (e.g.: mood and social background) enriching the context information found in usual pervasive platforms.

Enabling collaborative applications to take advantage of available information in traditional SNS (e.g. Facebook, LinkedIn) and enterprise SNS (e.g. Chatter[1]), can be beneficial as aggregators of inferences and suggestions for collaborative applications.

Nowadays tendencies show that is possible to make available not only information from a unique SNS but from many sources. Recent initiatives tried to create open standards for communication, embracing a Decentralized Social Networking Systems paradigm. However, many applications and systems use connectors to be able to communicate with the different SNSs. Thus it is necessary to implement individual connectors for each SNS.

Social computing has become a relevant field in the current organizations, mainly due to the emergence of Web 2.0. Despite being a relatively new area, social computing integration with other applications is rising inside and outside the SNS domain due its useful content for applications.

3. RELATED WORK

The pervasive platforms need to be strongly proactive in order to minimize users' interaction. The interaction with user, user-to-environment and user-to-group can be explored in depth for collaborative activities in pervasive environments. Typically, these systems do not move with the user as shown in MavHome (Youngblood et al 2005) or as in smart homes scenarios (Intille 2002).

Especially in smart spaces, social awareness is important to create a group of users that cooperates effectively and successfully. Considering this, smart spaces should enable users to create dynamic or static groups of interest to share common goals and tasks. According to (Wenger and Snyder 2000), a community is spontaneously built and legitimizes various degrees of members based on its internal rules.

Some studies regarding groups in pervasive environments are available in the literature. In PICO (Pervasive Information Community Organization), (Kumar 2003) presented a framework with the objective to create pervasive communities that can collaborate proactively in areas such as telemedicine, military and crisis management. This project aimed to achieve a sequence of events that can lead to the creation of communities. More recent approaches attempt to tackle the problem of pervasive communities supporting also mobile users. POPEYE project approached spontaneous virtual communities that can be formed in a P2P fashion for

[1] http://www.salesforce.com/chatter/

ISBN: 978-972-8939-92-2 © 2013 IADIS

collaborative work (Meyer 2008). However, the main project focus is in transport issues, lacking in services integration and social computing.

There are few works related that tries to approach CSCW and context information. Intermezzo (Edwards 2005) is a collaboration toolkit supporting the coordination information sharing, user awareness, session management and policy control. It addresses dynamic flexibility by allowing applications to modify the behavior of the toolkit based on the situational context of the collaboration. However, the main drawback is the application integration, which takes very specific details into account in order to adapt it to the user needs. The ECOSPACE Project (Prinz 2006) proposes an ICE to support semantic description of collaborative services for orchestration and choreography. 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 as a design aspect.

Recently (Haake 2010), proposed a generic framework for context modeling in collaboration environment to provide adaptive support. The conceptual approach proposes use of semantic languages to structure context information. However, the integration does not take into consideration aspects of social awareness, communication and integration with tools.

In summary, the proposals involving context-aware groups for collaborative activities still lack on environment integration, mobility or social capital where synergy could enhance collaborative experiences.

4. FRAMEWORK ARCHITECTURE

This section introduces the architecture of the context-aware framework for CSCW applications aiming to meet the objectives raised. The proposed framework is divided into 3 layers to allow modularity of the components involved. The modularity is a key factor since changes in one layer will not compromised the others. Fig. 2 illustrates the architecture. Each layer addresses a well-defined part and the details are described in the following subsections.

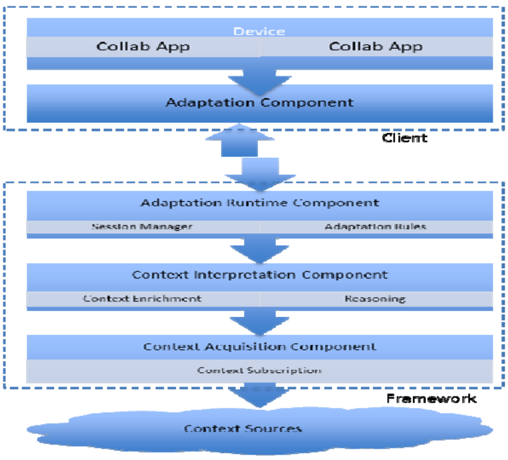

Figure 1. Framework Architecture.

4.1 Context Acquisition Component

The context acquisition is the first layer of the framework and is responsible for collecting relevant knowledge from the sources to the subsequent layer. As it is shown in use case studies in (Lima 2012), there are relevant context categories appropriate for enterprise scenarios. Some of the categories that might be used include: geographic location, current availability, current computational (mobile/desktop) resources and professional interests.

The context information is acquired via an external Context Broker, which communicates with the context sources. The Context Broker is responsible for gathering context information from the users through heterogeneous sources which can range from mobile sensors to SNS. The mobile devices provide location, availability and resources. On the other hand, SNSs provide information related to user profile such as interests, professional position and company. The information retrieved we classify 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. While short-term context comprises data that changes frequently such as location and availability (e.g. user busy, away).

Generally, pervasive systems have by default a specific component to manage context (as seen in section 2) allowing it to capture and retrieve context information. Thus, we believe the main focus of the framework needs to reside in analyses and adaptation, abstracting the concern to deal with context sources.

4.2 Context Interpretation Component

The context interpretation component models the knowledge and process the information collected from the previously layer. In order to translate the context to the framework this information is stored in a graph database, where each node represents a person. The user context information is used to assign edges representing relations existing among the users, based on the long and short-term context information.

The long-term information is found as properties of the person node as illustrated by Fig. 2A. While short-term is found as a new graph that is created according to demand as illustrated by Fig. 2B. 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 an individual. This separation allows this layer to make decisions on how to link individuals from the same community and assigning weights through the context information available.

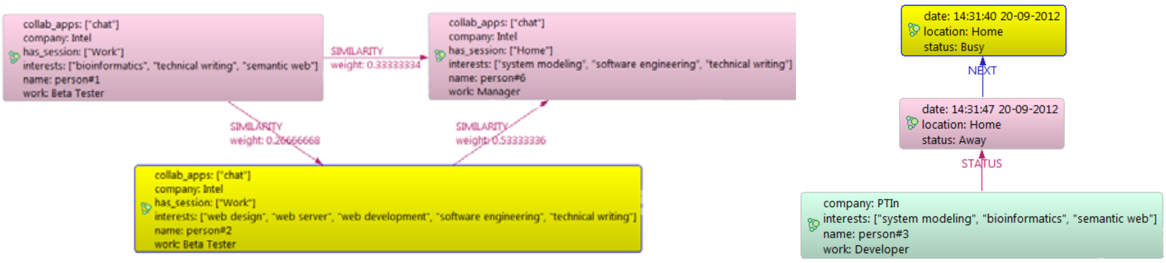

Figure 2. A) Sample graph of individuals with similarity weights. B) Short-term context information

In order to extend the context information collected from the previous component, the component performs an enrichment of context depending on the nature of the data. The information can be expressed in numeric or text values.

In cases of texts values, they can be submitted to externals semantic services for analysis, returning synonymous words that are aggregated with the existing information. This enrichment adds more information to apply decision conditions that takes place in the adaption layer.

After the enrichment it is possible to assign similarities among the persons in graph. This similarity enables to associate weights between person nodes and is calculated by dividing the matched information by the total available. This is applied for both and the result is divided by two as represented below:

$$S = (matched/node1) + (matched/node2) / 2$$

This component can also be requested to report which thresholding should be used to select relevant persons for a given weight. We adapted a method called automatic thresholding (Sahoo 1988) which it is often used in image processing for segmentation. The thresholding mean value is calculated in real-time based on the weights assigned previously. The weights values can be verified in the arrows among persons as illustrated in Fig 2A.

The numeric values can be enriched using historical information available, as an example it can be used to generate temporal series. This is helpful in case certain events to not occur regularly.

Finally, all the context information is checked automatically and periodically by the awareness monitor framework, depending on their nature. Long-term is verified in longer intervals, while short term is observed in real-time.

4.3 Adaptation Runtime Component

This component is responsible for managing the collaboration sessions and performs actions on the collaboration tools via rules. The adaptation rules are defined with: "weight" and "rule", e.g.: *“Rule1: individuals with same location”,”Rule2: individuals that work at the same department” “Rule3: individuals with similar interests”.* The rules have the following operators: same, different, equal, not equal, greater or equal, less or equal and similar. Adaptation rules can also perform a status checking to choose which communication channel is more appropriate for collaboration or which available communication channel is common to all parties. E.g. chat, voice.

For each stipulated rule it is assigned weights which identify the relevance for the framework start and maintain the sessions. The advantage of this format is to assign priorities to each rule without updating specific details. The rules and weights are formally represented below:

$$P = w1 \times Rule\ 1 + w2 \times Rule\ 2 + \ldots.\ w\text{-}n \times Rule\ \text{-}n$$

In the server application side it is deployed an extension component from the adaptation layer, which receives and performs actions on the collaborative application. The actions are sent to this external component, which has a wrapper in charge to receive and perform the actions on client application. The component executes actions such as: invite user, kick user, change moderator, etc.

The collaborative tools for the actual prototype can be used as usual client applications. Thus they are not responsible to manage sessions and do not need to be aware of the behaviour of others applications that are working at same time. Beside adaptation conditions, this layer is responsible for managing the collaborative sessions. The session manager coordinates the session members and their roles. A role defines a set of operations that can be executed by its members. Thus, each session is composed of members, roles and one or more applications in common.

5. IMPLEMENTATION

In order to build the framework prototype the architecture was modelled in UML and was developed in Java. To implement the graphs we choose the Neo4j graph database engine[2], which offers a graph-oriented model for data representation. In Neo4j graph structures consists of nodes, relationships and properties. For the framework, the nodes stores persons and also short-term context information for each one as illustrated in Fig 2. The relationships among persons is associated with the degree of similarity and is expressed in weight, while the short-term information has a timestamp and is represented as a LIFO (Last-In-First-Out), where the last update is more recent.

As we briefly described in section 4, we enrich the text information via external semantic sources. In preliminary test, the framework uses the Alchemy API[3] connector deployed at the Context Acquisition Component. The content to be analysed is submitted to the respective cloud and return results in JSON or XML format. Nowadays, there are several possibilities of extracting semantic meta-data such as: concept, sentimental, text categorization, etc. Other similar cloud services that can be integrated in the framework for content analyses are Synonyms[4] and Coginov[5].

The enrichment of context information is optional step by the framework and it is done via connectors just as SNSs connectors, it needs to be implemented by the developer. However, it is intended to implement other connectors for future tests.

6. PRELIMINARY TESTS

As part of the evaluation, we created an environment consisting in deploying the SOCIETIES Platform and the Framework as a third-party service on the same host.

[2] http://neo4j.org/
[3] http://www.alchemyapi.com/
[4] http://www.synonym.com
[5] http://coginov.com/

The service work at the top of platform which provide other core components, but for this proof concept was just necessary to integrate the community manager and the context management components. For context subscription it was implemented a connector to receive updates from Context Broker.

After communities' creation, it is possible to select one to start populating the framework mapping each CSS users to one-person node. The initial population starts with long-term context information and is structured as shown in Fig 2. Next step is to acquire the first short-term information available.

For this experience we have adopted as long-term information: job position, interests and company. For short-term: location and status.

With all initial information collected the framework starts the context monitor. The monitor checks for update changes for the two types of context. In order to simulate the behaviour of a real scenario, an automated script randomically changed the location and the status for each person.

Regarding the rules, we stipulated 3 rules and assign weights for: location, interests and presence. Thus, depending on location, interests and on the status, the monitor checks whether it is appropriate start a collaborative session or not. The rules were expressed as follows:

- First rule – Location: same location with weight 40%.
- Second rule - Interests: interests > thresholding with weight 40%.
- Third rule - Availability: status != busy with weight 20%.

The collaborative applications may optionally reside in the same machine of the Platform and the Framework, but that is not mandatory. As our aim is to test desktop and mobile devices, it is important that client application be compatible with most traditional protocols, for instance Openfire (XMPP) for IM or the Asterisk PBX (SIP) for VoIP. Therefore for the preliminary test, only a chat application was used to investigate the expected behaviour of the Framework.

We performed the test with the chat client in Android device (Xabber) and one in a Windows desktop (Jitsi) to verify invitations and the orchestration of participants among the collaborative sessions. The preliminary test evaluated the behaviour and predictability of the actions taken by the framework.

An evaluation in a real enterprise scenario will be conducted in a conference in which will be held in Intel's Campus, Ireland. The enterprise scenario analysis will be collected in April 2013.The scenario has been designed from a storyboard refined by a questionnaire that was conducted with Intel employees in 2011[6].

7. CONCLUSION AND FUTURE WORK

This paper proposes a framework to provide pro-active behaviour in asynchronous collaborative tools, exploring context information from physical and social interaction for pervasive environments.

We achieve these goals with the following contributions:

- Aggregating context information available from real-world source and SNSs to orchestrate collaborative activities.
- Introducing the Context-Aware Framework for CSCW Applications architecture and concepts for enterprise scenarios, presenting a modular design allowing reusing, maintaining and extending the components.
- Presenting an auto-threshold calculation, enabling to select relevant users for a given similarity.
- Demonstrating a classification of context information in short-term and long-term for approaching collaborative activities in pervasive scenarios.

The current approach aims to minimize the interaction of pervasive groups in collaboration environments by avoiding implicit interaction of the participants. The architecture focus in consumption of context information, abstracting the direct use of context sources which can be provided from the existent pervasive platform. Albeit the approach presented in this paper is mostly architectural driven, the framework prototype was implemented demonstrating its potentials and is available online as a third party service of the SOCIETIES project

Many challenges remain for interpretation of existing context related to individuals and groups for CSCW. In the future the authors intent to investigate other relevant methods for interpretation and analyses of context. Moreover, the enterprise evaluation we will intend to verify performance, usage and usability of the

[6] http://www.ict-societies.eu/files/2011/11/D8.1_public.pdf

Framework in real situations and with more collaborative tools. Finally, we aim to focus in the session management and the floor control permitting to provide an intelligent dissemination mechanism.

Context-aware CSCW applications merged with physical and social interaction can bring great benefits for the participants' usability. Furthermore, we believe that pervasive and social context are central point to the development of systems for the future collaborative applications.

ACKNOWLEDGEMENT

The authors wish to thank colleagues of SOCIETIES project, without whom this paper would not have been possible. The presented research is supported by FCT - Fundação para a Ciência e a Tecnologia (SFRH/BD/69381/2010).

REFERENCES

Grudin, J., 1988 Why CSCW applications fail: problems in the design and evaluation of organization of organizational interfaces. *Proceedings of the ACM conference on Computer-supported cooperative work*. USA, pp. 85-93.

Ackerman, M., 2000.The Intellectual Challenge of CSCW: The Gap Between Social Requirements and Technical Feasibility. *Human-Computer Interaction Journal*, v15, pp. 179-203.

Youngblood M.G. et al, 2005. Managing Adaptive Versatile Environments, *3rd IEEE International Conference on Pervasive Computing and Communications*, pp. 351-360.

Intille, S.S., 2002 Designing a home of the future. IEEE Pervasive Computing, vol.1, no. 2, pp. 76–82.

Farshchian, B and Divitini M. , 2010 *Collaboration Support for Mobile Users in Ubiquitous Environments. Handbook of Ambient Intelligence and Smart Environments*, pp.173-199.

Meyer E.M., et al, 2008. Location-based mapping services to support collaboration in spatially distributed workgroups. 4th International Conference on Collaborative Computing: Networking, Applications and Worksharing. pp.732-745.

Papadopoulou, E. et al, 2010.Personal Smart Spaces as a Basis for Identifying Users in Pervasive Systems, in Proc. Int. *Workshop on Ubiquitous Service Systems and Technologies (USST 2010)*, IEEE CS Press, pp. 88 – 93.

Roussaki I., et al, 2012. Context-awareness in wireless and mobile computing revisited to embrace social networking. *Communications Magazine, IEEE*, vol.50, no.6, pp.74-81.

Kalatzis N., et al, 2011. Managing Context Data in Personal Smart Spaces. *PECCS 2011 - Proceedings of the 1st International Conference on Pervasive and Embedded Computing and Communication Systems*, Portugal, pp. 5-7.

Lima, C., et al, 2012. Pervasive CSCW for smart spaces communities. *Pervasive Computing and Communications Workshops (PERCOM Workshops)*, IEEE International Conference, vol., no., pp.118-123, Switzerland, pp. 19-23.

Taylor, N.K, et al, 2011. Pervasive Computing in Daidalos. *Pervasive Computing, IEEE* , vol.10, no.1, pp.74-81.

Haake, J., et al, 2010. Modeling and exploiting context for adaptive collaboration. International Journal of Cooperative Information Systems, 19(1-2) pp. 71–120.

Prinz, W. et al, 2006. ECOSPACE - towards an integrated collaboration space for eProfessionals*. In Proceedings of 2nd International Conference on Collaborative Computing: networking, applications and worksharing*, pp. 39–45, Atlanta..

Edwards W. K., 2005. Putting Computing in Context: An Infrastructure to support extensible context-enhance collaborative Applications, *ACM Trans. Computer-Human Interactions* Nr. 4. Vol. 12, pp. 446-474.

Wenger E. and Snyder W., 2000.*Communities of practice: the organizational frontier. Harvard Business Review*, pp. 139-145.

Kumar M., et al., 2003. PICO: A Middleware Framework for Pervasive Computing*, IEEE Pervasive Computing*, Vol.2, No.3.

Dey, A. and Abowd, D., 2000. Towards a better understanding of context and context-awareness*. In Proceedings of CHI2000 Workshop on the What, Who, Where, When and How of Context-Awareness*, The Hague, Netherlands.

Sahoo P. K, et al, 1988. *A survey of thresholding techniques, Comput, Vision Graphics Image Process*., vol. 41, pp. 233-260.