Università degli Studi di Torino

Scuola di Dottorato in Scienza e Alta Tecnologia

Tesi di Dottorato di Ricerca in Scienza e Alta Tecnologia

Indirizzo: Informatica

A CONTEXTUAL APPROACH TO ACTIVITY AWARENESS SUPPORT IN DISTRIBUTED WORKSPACES

GIANNI BOSIO

Tutor: LILIANA ARDISSONO

Ciclo XXIII, Gennaio 2012

GIANNI BOSIO

A CONTEXTUAL APPROACH TO ACTIVITY AWARENESS SUPPORT IN DISTRIBUTED WORKSPACES

ABSTRACT

In this thesis we analyze how to handle the provision of activity awareness information in a distributed collaborative environment. We introduce a holistic support for activity awareness, which is handled in a way that is relative to the activity contexts users are involved in: in other words, awareness information is presented to the users and structured on a contextual basis related to their current contexts of activity. The awareness support is applied to a web-based collaborative environment, that offers complementary functions instead of linking separate applications and workspaces and provides unified access across multiple collaborations.

We present an experimental evaluation, where we want to test our awareness support from a twofold perspective: 1) the synchronous delivery of notifications, describing the events that occurred in users' contexts of activity, which is handled with several policies for different granularity levels, on the basis of the users' current activities; 2) a visualization model supporting an asynchronous, incremental access to awareness information.

Our results show that 1) our policies for notifications reduce the levels of workload on users, while supporting an up-to-the-moment understanding of the interaction with their shared contexts; and 2) our visualization model outperforms standard awareness spaces which provide a direct access to awareness events because it enables users to retrieve the relevant information quicker and more precisely.

PUBLICATIONS

Some ideas and figures have appeared previously in the following publications.

The candidate contributed to the following publications by discussing the ideas and main concepts, conducting user tests and writing sections of the final paper.

- L. Ardissono, G. Bosio, A. Goy, and G. Petrone. Context-Aware Notification Management in an Integrated Collaborative Environment. Proc. of UMAP 2009 workshop *Adaptation and Personalization for Web 2.0*, pp. 21-30. Trento, 2009. CEUR Workshop Proceedings, ISSN 1613-0073 (online http://ceur-ws.org/Vol-485).
- L. Ardissono, G. Bosio, A. Goy, G. Petrone and M. Segnan. Managing context-dependent workspace awareness in an e-collaboration environment. Proc. of WI/IAT09 workshop *Intelligent Web Interaction* (IWI 2009), pp. 42-45, IEEE. ISBN 978-0-7695-3801-3. Milano, 2009.
- L. Ardissono, G. Bosio and M. Segnan. An Activity Awareness Visualization Approach Supporting Context Resumption in Collaboration Environments. Proceedings of the *International Workshop on Adaptive Support for Team Collaboration 2011* (ASTC 2011) - CEUR Workshop Proceedings, Vol. 743, pp. 15-25, CEUR, 2011. ISSN: 1613-0073
- L. Ardissono and G. Bosio. Context-dependent awareness support in open collaboration environments. *User Modeling and User-Adapted Interaction*, pages 1-32, 2011. ISSN 0924-1868

- L. Ardissono, G. Bosio. Context-dependent notification management for awareness support in collaborative environments. *Advances in Dynamic and Static Media for Interactive Systems: Communicability, Computer Science and Design*, pp. 127-133, Blue Herons editions, 2011. ISBN: 978-88-96471-08-1

- L. Ardissono, G. Bosio and M. Segnan. A Visualization Model Supporting an Efficient Context Resumption in Collaboration Environments. *Lecture Notes in Computer Science n. 7138, Advances in User Modeling: Selected papers from UMAP 2011 Workshops*. pp. 5-17, Springer Verlag, 2012

The candidate contributed to the following publications by discussing the ideas and main concepts and writing minor sections of the final paper.

- L. Ardissono, G. Bosio, A. Goy, G. Petrone and M. Segnan. Open, collaborative task management in Web 2.0. *Proc. of IADIS International Conference Collaborative Technologies 2010*, IADIS Press. Freiburg, Germany, 2010, ISBN: 978-972-8939-21-2.

- L. Ardissono, G. Bosio, A. Goy, G. Petrone and M. Segnan, F. Torretta. Collaborative Service Clouds. *Int. Journal of Information Technology and Web Engineering*, IGI Global, 5(4), pp. 23-39, 2010. ISSN: 1554-1045

- L. Ardissono, G. Bosio, A. Goy, G. Petrone and M. Segnan, F. Torretta. Collaboration support for activity management in a Personal Cloud. I*nt. Journal of Distributed Systems and Technologies (IJDST)*, 2(4), pp. 30-43. IGI Global. ISSN: 1947-3532. EISSN: 1947-3540

- L. Ardissono, G. Bosio, A. Goy, G. Petrone and M. Segnan. Integration of Cloud Services for Web Collaboration: A User-Centered Perspective. *Models for Capitalizing on Web Engineering Advancements: Trends and Discoveries*, Ghazi I. Alkhatib (Ed.), pp. 1-19. IGI Global.

The candidate is the sole author of the following publication:

- G. Bosio. A user perspective on cloud computing. In *Proceedings of The 3rd International Conference on Advances in Human-oriented and Personalized Mechanisms, Technologies, and Services* (CENTRIC 2010), pages 1-4, Nice, France, 2010. IEEE.

CONTENTS

1 INTRODUCTION 1
2 BACKGROUND 7
2.1 Context Management 7
2.1.1 Definitions of Context 7
2.1.2 Context-aware Systems 10
2.1.3 Context Ambiguity 11
2.1.4 Relevant Implementations 12
2.2 Awareness at a Glance 17
2.2.1 Workspace Awareness 18
2.2.2 Activity Awareness 19
2.2.3 Relevant Implementations 20
2.3 Notification Management 23
2.3.1 The Role of Interruptions 23
2.3.2 Notifying Awareness 25
2.3.3 Relevant Implementations 27
2.4 Awareness Spaces 29
2.4.1 Asynchronous Awareness Representations 29
2.4.2 Relevant Implementations 31
2.5 Tag clouds 33
3 CONTEXT-BASED ACTIVITY AWARENESS SUPPORT 36
3.1 A Context-Aware Framework 36
3.1.1 Context Modeling in an Open Environment 36
3.1.2 Contextualization of Events 41
3.1.3 Recognizing the Current Context 44
3.2 Synchronous Awareness Information Management 47

3.2.1 Notification Policies 47
3.2.2 Presentation of Awareness Information 49
3.3 Asynchronous Awareness Information Management 51
3.3.1 Higher Layer: Awareness Cloud 52
3.3.2 Middle Layer: Hierarchical, Browsable Views on Awareness Information 55
3.3.3 Lower Layer: Context-dependent Awareness Space 57
3.4 Technical Specifications 58
3.4.1 Architecture 58
3.4.2 Awareness Information Management 61
3.4.3 Implementation Details 62
4 EXPERIMENTAL EVALUATION 64
4.1 Experimental settings 64
4.1.1 Workload Reduction 64
4.1.2 Performance improvement 65
4.1.3 Apparatus 66
4.2 First Experiment 66
4.2.1 Subjects 66
4.2.2 Experimental Design 67
4.2.3 Procedure 68
4.2.4 Measures 71
4.2.5 Results 71
4.3 Second Experiment 74
4.3.1 Subjects 74
4.3.2 Experimental Design 74
4.3.3 Procedure 75
4.3.4 Results 77
4.4 Analysis 79
4.4.1 Hypothesis H1 and H2 79

4.4.2 Hypothesis H3 82
4.5 Threats to validity 83
4.5.1 First Experiment 83
4.5.2 Second Experiment 84
5 COMPARISON WITH RELATED WORK 86
5.1 Motivations 86
5.2 Context 86
5.3 Awareness 90
5.3.1 Synchronous Awareness Management 90
5.3.2 Asynchronous Awareness Management 92
6 CONCLUSIONS 94
A APPENDIX 100
A.1 Experiment I: NASA-TLX Questionnaire 100
A.2 Experiment II: Post-test Questionnaire 101
A.3 Translation of italian terms in figures 101

BIBLIOGRAPHY 103

LIST OF FIGURES

Figure 1 Portion of the User Interface offered by the CTM for the specification and management of activity contexts. 40
Figure 2 Algorithm for updating the user's current focus of attention. The algorithm is written in pseudo-code and the dot notation is used to refer to the event features. 45
Figure 3 A notification message. 49
Figure 4 Higher visualization layer: Awareness Cloud of a user of a collaboration environment (user `utntest1@gmail.com`). 52
Figure 5 Middle visualization layer: Detailed view on awareness information focused on context Lavoro A. 56
Figure 6 Lower visualization layer: Awareness space of user `utntest1@gmail.com`, focused on tab Lavoro A. 57
Figure 7 Architecture of a collaboration environment based on the PCP and CONRAD. The architecture is depicted in a simplified form for readability purposes. 59
Figure 8 Awareness information flow concerning user U1. 61
Figure 9 Test results (No filter). 71
Figure 10 Test results (Context filter). 72
Figure 11 Test results (Task filter). 72
Figure 12 Mean values of workload, for each treatment condition. 73
Figure 13 Test results (Number of Errors). 77
Figure 14 Test results (Execution Times). 78

Figure 15 Test results (Number of Errors, Mean Values). 78
Figure 16 Test results (Execution Times, Mean Values). 79
Figure 17 Post-test questionnaire (Mean Values). 80

LIST OF TABLES

Table 1 Treatment conditions. 67
Table 2 Awareness events submitted in the experiment. 69

1 INTRODUCTION

The widespread availability of wireless connectivity, broadband internet connections and mobile devices are some of the main factors that are nowadays pushing either private and corporate users to adopt online services to carry out their activities. People can work on their projects and easily share their results with their collaborators by exploiting what is called the ubiquitous Internet [V.S. Pendyala and S.S.Y. Shim, 2009]:

- Online services, in the form of Web 2.0 applications, emerge: e-mail, Web calendars and social networks such as Twitter [Twitter, 2011] and Facebook [Facebook, 2011] help people to handle their life schedules, keep in touch with each other, share resources and interact with friends and larger communities.

- Web applications are experiencing growing adoption in the enterprise, leading to what can be called as the Enterprise 2.0, as they can support mobile users in the execution of activities and facilitate cooperation in co-located and distributed teams [TeamWox, 2011]. Shared collaboration spaces are proposed to allow users to manipulate artifacts and carry out tasks using heterogeneous business services and application environments [Prinz et al., 2006].

The adoption of online services for carrying out work, home and leisure tasks makes it possible to provide the user with awareness information about personal and shared activities, thanks to the electronic traces of the events generated by applications. As people are starting to use online applications for their personal activities, awareness support is becoming useful everyday. For instance, Grimes and Brush [2008] report that working parents struggle with the integration of different Web calendars, respectively used at home and in the office, in order to holistically manage their work and family schedules.

Awareness support is therefore a fundamental part of a broad range of collaborative activities, but the important role it covers also leads to some major issues:

- The execution of complex tasks typically requires multiple applications offering separate awareness support services. Thus, users receive a fragmented picture of the state of their activities, presented in different workspaces and delivered as parallel information flows which reflect application-dependent viewpoints on what has happened [Erickson et al., 2009, Ardissono et al., 2009]. In such a situation, users are not allowed to holistically manage the awareness information and have to explicitly reconstruct each workspace starting from its fragmented presentation.

- The provision of awareness information is a key factor for keeping users up-to-date with what happens around them; e.g., with the operations performed by their collaborators. However, the delivery of notifications describing the occurred events can interrupt the users' activities, with a possible disruptive effect on their emotional and attentional states.

- At the same time, activity awareness support is a key feature for helping people to resume the state of their tasks when switching from one to the other. However, the awareness spaces offered by most collaboration environments present large event histories which challenge users when searching for information [Mark and Su, 2007, Czerwinski et al., 2004, Iqbal and Horvitz, 2010, Haake et al., 2010].

As a practical example, let's consider Jane, a woman who has one child and that works in a place quite far from her home; to reach her workplace, she participates in a car-sharing initiative with some colleagues. In her work activity, Jane is actually participating in different projects involving distributed teams of people, who keep in touch telematically with each other to carry out the assigned tasks. She has some spare time, during which she takes clarinet lessons; she often happens to skip them, when she is pressed to deal with higher-priority commitments. By managing her personal and work schedules online, Jane constantly receives news from her child's school and from her travel mates using her smartphone. Similarly, she is informed about the tasks that have been assigned to her and about the documents that have been modified by her coworkers. Having said that, how many different notifications is she going to receive, in a mixed order, belonging to such parallel contexts? And are all the notifications sufficiently urgent and important to be delivered, even though they have nothing to do with Janes's current activities? For instance, suppose that our hero is carrying out a shared editing task with a colleague in order to prepare a report for a work project. If the teacher of the clarinet course sends a confirmation note regarding the next lesson, should Jane be notified immediately or later on? Moreover, should she decide to take a holiday and not to check her smartphone for a week, how difficult would it be to tackle all the past messages in the first hours of her comeback to work? With this work, we aim at providing users like Jane with an automated support for:

- Specifying the kind of awareness information that is the most appropriate, from a possibly large amount of information sources, based on the user's current activity context.
- Tailoring the delivery of notifications and the application of filters to individual notification preferences.

- Handling notifications about past activities in an intuitive and effective way.

Our research questions were:

- How can we develop an open framework for supporting awareness in distributed, collaborative workspaces?
- How should context information be formalized for building such support?
- Which is the right solution for supporting the synchronous awareness delivery process?
- Which is the right solution for supporting the asynchronous awareness delivery process?

In this thesis, we describe a holistic awareness support service for open collaboration environments, which can be extended by integrating external applications. For this purpose, we assume that the awareness events describing the actions performed by users in the collaboration environment are classified in activity contexts, so that they can be managed in a structured awareness space reflecting the user's collaborations and private activities.

As a possible solution to the trade-off between informing and interrupting users, we defined two context-dependent notification management policies which support the selection of the notifications to be delivered on the basis of the user's current activities, at different granularity levels: general collaboration context versus task carried out.

In order to help users to easily resume the state of their activity contexts, we offer an incremental access to awareness information, reducing the size of the information space by enabling the user to visualize awareness events from perspectives reflecting different information needs. We propose a model for the visualization of recent information which summarizes the state of the user's activity contexts and from which the details of the occurred events can be easily retrieved. In order to provide the user with a picture of the evolution of her/his collaborations, we implement a specifically shaped tag cloud [Sinclair and Cardew-Hall, 2008].

We evaluate our approach with 2 experiments, involving users of the framework. We use standard tools, such as questionnaires, and measures of mental overload, user performance (speed and precision) and user satisfaction, to show the advantages of our proposed approach.

This thesis is organized as follows:

- In Chapter 2, we present state of the art literature on awareness information, interruption management and context modeling.
- Chapter 3 presents our framework for managing context-based activity awareness; both synchronous notifications and asynchronous, incremental access to awareness information are covered in this chapter.
- Chapter 4 describes the evaluations of our notification management and visualization model implementations.
- Chapter 5 compares our approach with related work and highlights its degree of innovation.
- Chapter 6 concludes the thesis, summing up the main contributions and discussing possible future work.

Our solution for activity awareness management could be interesting for different reasons. It is relevant to the area of collaborative systems, as it describes an open architecture supporting the integration of Web-based, heterogeneous software components into a customized collaboration environment. It offers a good application for notification management techniques, introducing a context-dependent dimension, which can also be complemented with a feature-based selection in order to support the development of finer-grained notification policies. Our visualization model for activity awareness events can be relevant to the area of information retrieval, as well as corporate search engines and enterprise search, as it is aimed at preventing information overload and increasing users' performance.

2

BACKGROUND

This chapter will present definitions, theories and state-of-the-art implementations related to four areas that are strictly connected with the subject of this thesis. They are, respectively: context management (section 2.1), *awareness in general* (section 2.2) *and two specifications of the awareness subject: notification management* (section 2.3) *and awareness spaces* (section 2.4). *Finally, tag clouds are introduced* (section 2.5), *as they are an element that has been exploited in this thesis for awareness purposes.*

2.1 CONTEXT MANAGEMENT

2.1.1 *Definitions of Context*

The research on context modeling and ubiquitous systems provides various definitions of context. In [Dey and Abowd, 2000], a *context* is defined as *any information that can be used to characterize the situation of an entity*. An entity could be a person, a place, or an object that is considered relevant to the interaction between a user and an application, including the user and the application themselves. The authors recognize the importance of four kinds of context: location, identity, activity and time; such categories define primary kinds of contexts. All the information that can be indexed by primary contexts (in other words, the attributes of the primary contexts) are defined as second level contexts.

A different approach for defining context is followed in [Dourish, 2004], where context is presented as an *interactional problem*. Four assumptions are derived:

- Rather than considering context to be information, contextuality is a *relational property* that holds between objects or activities (it may or may not be contextually relevant to some particular activity).
- Rather than considering that context can be delineated and defined in advance, the scope of contextual features is *defined dynamically*.
- Rather than considering that context is stable, it instead argues that context is *particular* to each occasion of activity or action.
- Rather than taking context and content to be two separable entities, it instead argues that context *arises* from the activity.

The author pushes further his conclusions, turning his attention from "context" (as a set of descriptive features of settings) to "practice" (forms of engagement with those settings); the central role is occupied by the meanings of people's actions, in terms of consequences and interpretations of such actions for themselves and the others.

In [Chalmers, 2004], the author expands the aforementioned work of Dourish [2004], focusing his attention on the trade-off between pre-designed formal categorizations of context and socially-constructed, situated actions that are traditionally excluded from these representations. From the latter concept the author derives the conclusion that it is up to *users to create and communicate meaning* and also to *manage contextualization* of activities and objects. As a consequence, users have to find structures that are salient to the context, with all the dynamism, subjectivity and openness that are key characteristics of context. Designers cannot predetermine users' activities, meanings and interpretations, but, at the same time and quite obviously, have no choice but to influence the meaning that users make and the ways that they manage contextualization.

The work in [Bazire and Brézillon, 2005] further highlights the lack of consensus on context definition: the authors collected a set of 150 definitions of context, originating from various disciplines, and analyzed this corpus of definitions through two formal methods. As a result, they built a model of context representing the components of the situation where the context is taken into account and the different relations between those components, with the hypothesis that the reason why definitions diverge is that each definition does not put its focus of attention on the same topics.

The authors of [Zimmermann et al., 2007] try to give a definition that is aimed at the operational use of context. They enumerate five categories for context information:

- *Individuality*, that comprises anything that can be observed about an entity, typically its state. An entity can either be an individual entity or groups of entities that share common aspects of the context.
- *Time*, that subsumes time information like the time zone of the client, the current time or any virtual time.
- *Location*, that classifies the physical or virtual residence of an entity.
- *Activity*, that covers the activities the entity is currently and in future involved in. In most situations an entity is engaged in a *task*, that determines the goals of the performed activities.
- *Relations*, that captures the relations an entity has established to other entities.

In the vision of the authors, context is an operational term: something is context because of the way it is used in interpretation, not due to its inherent properties; therefore, it has to be meant as dynamic, transitional and shareable.

2.1.2 *Context-aware Systems*

In [Dey and Abowd, 2000] a system is defined as *context-aware* if *it uses context to provide relevant information and/or services to the user*, where relevance depends on the user's task. Context-aware applications are classified as (possibly) supporting three features: *presentation* of information and services to a user, *automatic execution* of a service and *tagging* of context to information for later retrieval. The definition of context described in [Dey and Abowd, 2000] is exploited by the same authors in [Dey and Abowd, 2001], where six requirements that designers must fulfill to deal with context are presented:

- *Separation* of concerns, that is, separating how context is acquired from how it is used, so that applications can use contextual information without dealing with how it is acquired.
- Context *interpretation*: applications should not interact with low level contextual information, as it should be interpreted by the architecture before it deals with user programs.
- *Transparent, distributed communications*: if a context is composed of a multitude of interconnected devices, the distribution of the communication between different machines should be transparent to sensors and applications.
- *Constant availability* of context acquisition: components that acquire context should be persistent, available all the time and executing independently from the applications that use them.

- Context *storage*: the architecture, and not the single applications, should support the storage of context details.
- *Resource discovery*: the architecture needs to support a form of resource discovery, in order to find any applicable components and to provide the application with ways to access them, whenever an application requests some kind of contextual information.

Most context-aware systems adapt their behavior to the user's individual context, concerning its physical and social aspects. Consequently, the context representation includes different types of information, such as the user's geographical position and environmental conditions (e.g., see [Cheverst et al., 2000]), the device used to interact with the system (e.g., in [Ardissono et al., 2003]) and the quality of the internet connection (e.g., see [Ding et al., 2001]), as well as the presence of nearby services and objects of interest (e.g., see [Zimmermann and Lorenz, 2008, Cheverst et al., 2000]).

2.1.3 *Context Ambiguity*

As discussed in [Dey and Mankoff, 2005], the recognition of the actual context surrounding the user is often ambiguous regardless of how good are the sensors and interpretation methods employed. For this reason, context-aware applications should explicitly handle ambiguous contexts. *Mediation* techniques are interface elements that help the user to identify and fix system actions that are incorrect, or potentially involve the user in helping the system to avoid making those mistakes in the first place. The targets of this practice are:

- addressing the issues of providing feedback to support users in knowing that the system is attending to them; and

- knowing what the system has done and providing the ability to disambiguate sensed input to avoid the system taking incorrect actions.

The authors state that the information available to context recognizers is often too limited to support the level of certainty necessary to eliminate the need for user feedback. It is therefore interesting enough to underline one of the guidelines expressed in this work: *ambiguity should be retained until mediation is necessary for an application to proceed*.

2.1.4 *Relevant Implementations*

Context aware systems that deal with location data are widespread and the demand for them is growing due to the increasing spread of mobile devices [Baldauf et al., 2007]. Typical examples for location-aware systems are tourist guide projects, that are able to display information on the basis of the current location of the user. Nevertheless, examples of location-aware systems can be found in educational institutions [Burrell and Gay, 2002], workplaces [Kerer et al., 2004] and hospitals [Muñoz et al., 2003].

In [Baldauf et al., 2007] a survey on context-aware systems is presented. The authors describe different design principles and context models for context-aware systems; from the comparison of those systems and frameworks it is evident their similarity in structure, which is most often layered, with a strict division of the context data acquisition and use.

- The Service-Oriented Context-Aware Middleware (SOCAM) project [Gu et al., 2004] is an architecture for building context-aware mobile services, that uses a central server (the context interpreter) which obtains context data through distributed context providers; such data is processed by the central server and then delivered to the clients.

- The Context-Awareness Sub-Structure (CASS) presented in [Fahy and Clarke, 2004] introduced an extensible, centralised middleware approach for context-aware mobile applications. The middleware is composed of an Interpreter, a ContextRetriever, a RuleEngine and a SensorListener. Mobile clients connect to the server over wireless networks.

- The Context Toolkit [Dey and Abowd, 1999] implements a more peer-to-peer oriented architecture, with distributed sensor units (the widgets), interpreters and aggregators, that are registered to a centralized discoverer in order to be found by client applications.

Context-awareness has been implemented within several collaborative systems. In [Rittenbruch, 2002], contextual awareness is defined as composed of three elements:

- *representation* of shared context characteristics;

- *assignment* of events to context; and

- *selection of events*, which is based on the context information.

The author describes three kinds of methods for the representation and assignment of contexts:

- *Active* methods, where the process of representing context and assigning events is actively performed by the user.

- *Structural* methods, where the process of assigning events to contexts is realized with the choice of pre-existing structural context representation.

- *Passive* methods, where both processes are performed automatically on a low contextual level.

Structural and active methods are implemented within the Atmosphere framework as *spheres* and *contextors*, respectively:

- Spheres are user-defined representations of contexts, which contain a set of jointly used artifacts. Activities on such artifacts are grouped by and related to the sphere to which they belong.

- Contextors are user-generated descriptions of activities, which are represented within a particular sphere. Contextors are used in order to model the intentions of the users operating within a context, asking people to explicitly select the contextors which motivate their own actions while they interact with artefatcs (e.g., "final review" of a document).

In [Gross and Prinz, 2003], context is represented using a set of attributes, in order to match events to contexts of origin and to detect the current work context of the user:

- To each context, a unique *name* is given.

- Each context has an *administrator*, that is the person who created and manages the context.

- Users that work in a context and produce events through their actions are *members* of that context.

- *Locations*, where events are produced, can be electronic (e.g., shared workspaces) or physical areas (e.g., meeting rooms).

- The *artifacts* of a context are all objects on which users operate.

- Each context is associated with various single-use and cooperative *applications* (e.g., text editors, programming environments, groupware applications).

- *Events* that are produced in a context are described by their types.
- An *access control list* for a context comprises all the rights that exist for each member of a context.
- Contexts can be *interconnected* with other contexts, and shared elements can be organized in sub-contexts.

This model is applied to the BSCW shared workspace system [Horstmann and Bentley, 1997, OrbiTeam Software GmbH & Co. KG, 2011]. An event-based, client-server framework, called ENI, uses a sensor-based architecture for the capturing of events and various indicators for their presentation. The BSCW server generates for each user action an event that is forwarded to the ENI server via http in XML-based format; ENI server's context module maps incoming events to a context of origin and checks or updates their attributes within the server. A context is then attached to each event. A similar approach is followed for matching the current activity of a user to a specific working context, where information from the operating system about running applications and processed objects is used.

In [Vonrueden and Prinz, 2007], context is defined in terms of documents' metadata. Context information of a document is classified in two levels:

- First level contexts, which can be data regarding users (e.g., names and roles), concepts (e.g., tags and annotations), resources (e.g., related files) and states (e.g., versions and permissions).
- Second level contexts, which extend first level with specific, real time information, such as presence and availability of related users and related documents.

The authors describe the ECOSPACE project, where an application that handles documents metadata works as a Context Provider (the authors propose BSCW as a possible candidate for first level context information). To each document is associated a Context Tag, which is an XML-Structure that is integrated inside the file. The Context Tag contains a single document-id and one or more Context Scopes elements, one for each context element of the document; each Context Scope points to the Context Provider that handles that specific context information. A Context Broker web server is the link between a Context Scope and the Context Provider, managing the retrieval of single context-information and offering representations of a context in form of webservices and user-interfaces.

The authors of [Haake et al., 2010] present a layered context model that could support the provision of self-adaptive services in collaboration environments. The approach is based on a unified formal graph model for context using different representations such as RDF, OWL or SPARQL. The context model is composed of four layers:

- The *knowledge* layer acts as the basis for the context framework, as it contains all relevant conceptual and factual knowledge about the application domain.
- The *state* layer contains information about the current situation including information about physical environment, computing environment, resources and user model.
- The *contextualization* layer contains contextualization rules that define which subset of the state is relevant for a given focus. A focus can be an arbitrary subset of the state, which represents the system's current center of attention.
- In the *adaptation* layer, the relevant rules are identified using the contextualized state.

The adaptation process works translating user activity into state, deriving context for a given focus, and executing adaptation rules on this context (e.g., the presentation of important documents or even hardware adaptations).

2.2 AWARENESS AT A GLANCE

In a collaborative environment, sharing information and knowledge about group and individual activity is central to coordination and successful collaboration. All these factors contribute towards what is commonly referred to as *awareness*. In [Dourish and Bellotti, 1992], awareness is defined as *an understanding of the activities of others, which provides a context for your own activity*. This context is closely related to the group's activity, as it can be a measure to evaluate how relevant to the group is each individual contribution, with respect to the common goals. The context in which the collaboration takes place can be seen as comprised of two elements:

- The *object* of that collaboration, that is, any individual contribution and their content.
- The way in which the object is *produced*, which comprises the character of the contribution and its significance to the group as a whole.

Awareness information can be presented in two modalities:

- In *synchronous* modality, awareness information is presented as it happens, providing participants with feedback of others' current activities.
- In *asynchronous* modality, the workspace presents past activity information, so as to give an individual awareness of the activities of other participants integrated with the work object itself.

However, it is made clear that these are not two different modes of the system, but rather are two facets of a single view of awareness information.

2.2.1 *Workspace Awareness*

Early studies on awareness focus on shared workspaces, which act as a stage for rich person-to-person interaction. In order to synchronize with each other, people need information about their collaborators, the activities carried out, etc., similarly to what naturally happens in co-located collaboration. Collecting information available in and through collaborative workspaces allows people to capture others' locations, activities, and intentions relative to the task and to the space. Such elements are defined in [Gutwin et al., 1996] as *workspace awareness*, which can be seen as *the collection of up-to-the minute knowledge a person holds about the state of another's interaction with the workspace*.

Workspace awareness is seen both as a *product* and a *process*:

- The product is the state of understanding about another person's interaction with the workspace, that allows people to interpret events, anticipate needs, and interact appropriately.
- The process is the continuous cycle of extracting information from the environment, integrating this information with existing knowledge, and using that knowledge to direct further perception.

The problem of maintaining workspace awareness revolves around obtaining useful information, rather than around how people use it [Gutwin and Greenberg, 1999]. Information to be gathered concerns *who* is working in a shared context, *what* they are doing, *where* they are working, *when* various events happen and *how* those events occur. People must be able to determine what to look for in the workspace environment, they must gather perceptual information about others and they have to make sense of the perceptual information in light of what they already know [Gutwin, 1997]. Awareness can serve different purposes for different workspace members, depending on their task and perspective [Simone et al., 1999]; symmetry of workspace awareness information (i.e., others have the same awareness information of my actions as myself) is a key concept for avoiding problems in interaction between users, as they usually behave with the belief that other users will receive specific notification of their actions.

2.2.2 *Activity Awareness*

The concept of workspace awareness can be extended in order to be applied to a set of interrelated activities, each of which is executed within an individual workspace. This concept is called *activity awareness* [Hayashi et al., 1999] and is aimed at generating a collective activity perspective and asynchronous progress notifications. The word activity is used to mean *a human process of a worker to achieve some specific goal* (e.g., writing a specific report); dynamic awareness scope determines the scope of awareness as a set of individual activities related to a collaborative activity (this set may change over time). Awareness *nodes* are introduced to provide perspective on collaborative activities, in the form of viewer-independent viewpoint for activity structures:

- *People* awareness summarizes activities performed by a given individual.
- *Project* awareness summarizes activities that are components of a given project.
- *Place* awareness summarizes activities that interact at a place.

An evolution of the same concept can be found in [Carroll et al., 2003], where activity awareness is defined as *an awareness of other people's plans and understandings*. Complex, long term, coordinated activity cannot succeed without on-going interpretation of current goals, accurate and continuing assessment of the current situation, and analysis and management of resources (including time) that constrain execution of possible plans. Taken alone, the low-level events describing the operations performed on shared artifacts are not enough to help users synchronize. For that purpose, users also need to get a picture of the evolution of their collaboration contexts.

2.2.3 *Relevant Implementations*

Early works, which revolved around the notion of workspace awareness, presented solutions that were appliable to the problems of synchronous collaboration in shared workspaces.

In [Dourish and Bellotti, 1992], a comparative study of three particular collaborative writing systems is presented, that is focused on the mechanisms they provide for sharing awareness information between collaborators:

- The first two analyzed systems, Quilt and PREP, use roles, explicit annotation and directed messaging to provide awareness and coordination information. Information about the content of activities in progress is divorced from the activities themselves, but must be provided separately through some other channel.
- The third system, GROVE, uses audio communication to support informal awareness between participants, but the system's representations of other users' activities are implicit. Parts of the document are presented to each user through a view, which may be public, private or shared. In addition, access control mechanisms can be used at each point in the tree to control who can see, edit or create a text fragment.

All of these systems embody an assumption that a simple awareness of other's activity needs to be augmented with other explicit, or restrictive mechanisms for ensuring an easy collaboration, such as annotations, role assignment, access rights. This approach is problematic for the authors, as they identify three main issues:

- The user who provides the information does not directly benefit; moreover, the price of heightened awareness for the group is clearly restriction in the potential activities of individuals.

- The provided information may not be at the appropriate level of specificity for the receiving individuals, or it may not be relevant to their particular activities at the time.

- Delivery is controlled more by the sender than by the recipient; information is not continually available to be browsed, therefore the recipient is restricted in ways of using the information.

As a solution to these issues, the authors describe *shared feedback* as a way to make information about individual activities apparent to other participants, by presenting feedback on operations within a shared workspace. This approach is proved to lead to low overheads for the providers and recipients of awareness information, to make available information as and when needed as a context for individual activities, and to avoid restrictive prestructuring of group activity.

The ShrEdit synchronous, multi-user text editor is presented as an example of a good implementation of shared feedback. It features a shared window, that presents a view onto a shared document, and a control window, associated with each edit window, that displays the names of the participants in the session. All participants have equal access to the shared document windows and can type at any time: shared windows are indeed locked at the level of text selections (no user can edit text which has been selected by another user).

In [Gutwin et al., 1996], the notion of shared feedback is actually expanded, and *workspace awareness widgets* are presented as an a awareness solution for relaxed-WYSIWIS ("what you see is what I see") collaborative applications:

- *Radar views*. They are a class of widgets based on miniature overviews of an entire workspace, that shows all movement of and changes to artifacts in the workspace, providing information about others' actions.
- *Multiple-WYSIWIS views*. They show a scaled-down duplicate of each person's view of the workspace, giving the group a common, composite view of the whole workspace.
- The *"what you see is what I do"* widget. It provides fullsize details, but shows only a limited part of the other person's view, as it shows only the immediate context around another person's cursor.
- *Workspace teleportals*. They are workspaces that can "teleport" users to another person's location with the pression of a mouse button, returning to their original view when the button is released.

Interlocus, presented in [Hayashi et al., 1999], shifts focus from workspaces to activities, introducing a temporal workspace model that is able to extract activity information from individual workspaces. Activity information is composed of:

- *Threads*, recorded sequences of changes to a user's workspace.
- *Snaps*, the states of a current activity of a given time; sets of snaps form a thread.
- *Anchors*, spatial frames of a workspace that identify a document or a tool relevant to an activity; sets of anchors form a snap.

The awareness scope is defined on the basis of interconnected threads: two or more threads are connected if they share a document, or if an explicit link is stated between them. A *collective activity view* function presents a summary of each collaborative activity, in the form of a recollection of individual actions performed on shared documents. A *what's new* function is an individual view that shows the changes to each interconnected workspace since the last time a user visited it.

A solution for activity awareness in complex, long term activities is presented in [Ganoe et al., 2003], which introduces the Classroom BRIDGE software, designed to support distributed group projects between school classrooms. This system exploits an iconic representation of work artifacts, which is uniformously declined in two different (but interconnected) graphical interfaces:

- A single *large screen display*, which provides awareness information among projects at the classroom level, in order to allow quick comparison across groups and to sense relative progress toward the project goals.
- A *coordinated timeline*, that integrates activity awareness information for planning sake, with individual document timelines that provide a history of the documents in form of icons (one per version of the document).

2.3 NOTIFICATION MANAGEMENT

2.3.1 *The Role of Interruptions*

Interruptions are increasingly common in human-computer interaction. In fact, different agents, such as electronic mailers, Instant Messengers and VOIP calling applications are commonly present and active on many computers and fight to gain the user's attention when something related to them happens. With the advent of the Web 2.0, many new types of software agents, such as shared calendars and shared maps, are becoming sources of notifications and thus of interruptions.

In [Bailey et al., 2001], interruptions are defined as *peripheral, nonessential information, that is helpful or of interest to the user but not necessarily related to the user's current task*, which can be thought as *primary*. The effects of interruptions on people's activities have been thoroughly studied. It has been repeatedly noted that an interruption has a disruptive effect on the user's task performance and emotional state, as well as on the user's capability of recovering the primary task or to perform post-interruption tasks [Cohen, 1980, Bailey et al., 2000, 2001, Cellier and Eyrolle, 1992], [McFarlane and Latorella, 2002]. However, some studies highlight subtler aspects of this phenomenon. For instance, in [Czerwinski et al., 1991b] it is shown that subjects' performance is lower when interrupted by a task that is displayed in a similar way as the primary task. Furthermore, the authors of [Czerwinski et al., 1991a] identify an inverse relation between (primary and interruption) task similarity and people's ability to remember information about the interrupted task. The work reported in [Cutrell et al., 2000] achieves similar results, showing that interruptions that have no relevance to the main task result in longer times to process them and longer task resumption times than relevant messages. Finally, Mark et al. [2008] investigate disruption costs and shows that the context of an interruption does not make a difference on users' performance: people complete the interrupted tasks in less time, with no difference in quality. These findings suggest that people compensate for interruptions by working faster but this makes them experience more stress, higher frustration, time pressure and effort.

2.3.2 *Notifying Awareness*

Interruptions are particularly critical in collaboration environments, which base the awareness support on the delivery of notifications to their users: the main goal of a notification system is in fact to deliver current, important information in an efficient and effective manner without causing unwanted distraction to ongoing tasks [McCrickard et al., 2003].

The existence of a trade-off between interrupting the user and efficiently supporting activity execution is evident. As discussed in [Iqbal and Horvitz, 2010], users acknowledge notifications as disruptive, yet opt for them because of their perceived value in providing awareness. In summary, coworkers are involved in multiple projects and tasks [Mark and Su, 2007, Czerwinski et al., 2004], and this fact increases the potential interruptions from colleagues and automatic agents.

As far as notification management is concerned, McFarlane [2002] discusses the design of systems capable of supporting human interruption management. Four design solutions to coordinate user interruptions are proposed:

- *Immediate*, in which people have no choice but to handle interruptions immediately.
- *Negotiated*, where people can choose independently wether to interrupt their current activity and switch to another one, or not.
- *Mediated*, where it is up to an autonomous agent (the mediator) to decide when to interrupt the user.
- *Scheduled*, where interruptions happen at a premeditated time, so that people can plan their activities accordingly.

Awareness information can also be related to the user's attentional state. In the environments supporting cooperative work, attentional switches are often solicited. The detection of the user's current attentional state, the detection and evaluation of alternative states and their presentation to the user are key factors for handling awareness information in Computer Supported Collaborative Work systems [Roda and Thomas, 2006]. The user's attentional state can be inferred by sensors or by her/his operational context (as current tasks and goals); alternative foci of attention can be evaluated with respect to the appropriateness of the user's current focus, i.e., they must represent a "better value" for achieving goals.

Strategies for presentation can be defined taking into account the interruption from a primary task, the user's reaction to secondary information and the comprehension of information presented in a secondary display [McCrickard and Chewar, 2003]. Content, modality and timing are important parameters to be considered in order to realize secondary displays of awareness information. For instance, the study presented in [Adamczyk and Bailey, 2004] proves that the best times for interruptions correspond to coarse breakpoints in the task execution. Therefore, a hierarchical task model, which identifies coarse events and fine events in a task structure, is mandatory for this purpose.

Bailey et al. [2006] describe a framework for specifying and monitoring user tasks, based on user-defined models of the task structure. Moreover, Iqbal and Bailey [2007] describe a statistical model of the task structure for identifying breakpoints in the execution of generic tasks and the application of deferral policies [Iqbal and Bailey, 2008].

Other works, e.g., [Iqbal and Horvitz, 2006], [Iqbal and Horvitz, 2007] and [Bailey and Iqbal, 2008], further study the impact of interruptions in different phases of the task execution to identify moments where the user can be interrupted in a less disruptive way. Horvitz [1999] bases the management of notifications on evaluating the payoff of interrupting the user, in terms of expected information, by means of a Bayesian inference model.

In the *bounded deferral* approach proposed by Horvitz et al. [2005], low-priority interruptions are deferred, having observed that people are not constant in their attention and few minutes after having started a task they transit to an attentional state which can be interrupted in a less disruptive way.

2.3.3 *Relevant Implementations*

Many groupware and project management tools (e.g, BSCW [Horstmann and Bentley, 1997, OrbiTeam Software GmbH & Co. KG, 2011], Project View IM [Scupelli et al., 2005], Collanos [Collanos, 2008], ActiveCollab [ActiveCollab, 2008], TeamWox [TeamWox, 2011]) support a project-based organization and filtering of notifications. They attempt to reduce the number of messages delivered to the users by enabling them to specify the workspaces from which they wants, or does not want, to receive awareness information. Such notification policies have to be explicitly updated by the user in order to reflect possible changes in interests.

The NESSIE awareness management environment [Prinz, 1999] supports the specification of interest profiles used to filter the notifications generated by the applications. The environment offers an application independent infrastructure and a set of sensors and configurable indicators for event notifications. The contextualization of events is based on BSCW shared workspaces information.

A similar approach is implemented in the AREA framework [Fuchs, 1999], which supports awareness by providing notifications about activities performed by users creating new events describing these activities. For each new event AREA determines which users will be notified about the event by evaluating all user profiles according to their interest specifications and matching them against the privacy requirements of the performing actor; users have to specify an *intensity* value for each kind of event notification to be received. The framework exploits the POLIAwaC [Sohlenkamp et al., 2000] document-sharing environment to handle the provision of notifications:

- In the *lowest* intensity, events are displayed using graphical artifacts such as symbol embellishments and color overlays.
- In the *next higher* intensity level, another graphical artifact, such as icon enlargements, is adopted.
- Events in the *third* intensity level are displayed as a message in an event ticker at the bottom of the main window, which incorporates a browsable list of recent events.
- Events with the *highest* intensity value are displayed in a modal dialogue box, which has to be acknowledged by the user, forcing users to focus on the information presentation explicitly.

In [Wang et al., 2007], the BSCW awareness support is extended by defining dynamic notification preference profiles, which model the user's long-term interests in workspaces; a rule-based inference mechanism tunes the intensity of notification streams, aiming at reducing the information overload.

Other works, such as CASSIUS [Kantor and Redmiles, 2002], take a very different approach: they invest in the usability of the notification management system in order to enable users to easily find the interesting notification sources and to efficiently subscribe for information, even in ad-hoc ways (e.g., performing short-term subscriptions to satisfy very specific information needs).

2.4 AWARENESS SPACES

2.4.1 *Asynchronous Awareness Representations*

Awareness spaces are commonly present in groupware and project management tools as a way to handle awareness information asynchronously. Such spaces must provide users with information on the activities of other users, with respect to the objects within a shared workspace [Appelt, 2001]. Events, that are triggered whenever a user performs an action in a shared workspace (e.g., uploading a new document, downloading or accessing an existing document, renaming a document and so on) are recorded and presented to each user in a structured space.

In [Fuchs et al., 1995] the authors present an organizational context for describing the state of cooperative work, that allows the provision of information concerning synchronous as well as asynchronous situations. Such model is based on three entities:

- *Objects*: work artifacts (e.g., documents and other working resources of any kind), entities that compose the organizational context of work (e.g., groups, departments, organizational roles), users.
- *Relations*: structural relations describe any relationship between objects and an associated organizational context; operational relations describe relations between an actor and an object; semantic relations express any semantic similarity between two entities in the system.
- *Events*: modification events are generated when the state of an object changes due to some action of a user; activities are higher level events, that mark starting and ending of a series of actions.

In [Tam and Greenberg, 2006], *asynchronous change awareness of artifacts* is defined as *the ability of individuals to track the asynchronous changes made to a* *collaborative document or surface by other participants*. Building from theories derived from Gutwin [1997], the authors highlight the fact that people may need to view aspects of the workspace they are working in from different perspectives, of which they enumerate three:

- the artifacts that exist within it (*artifact-based* view);
- the people who work within it (*person-based* view); and
- the workspace itself, as a single one or as collection of related locales (*workspace-based* view).

The specific perspective has effect on both the information that users are interested in and the way that the information is represented. Specifically, within the bounds of the chosen perspective, a person can ask specific questions to probe for further details of changes:

- *Who* questions, whose answers generate *presence* history, *identity* history, *readership* history and *authorship* history.
- *What* questions, whose answers generate *action* history.
- *How* questions, whose answers generate *process* history and *outcome* history.
- *When* questions, whose answers generate *event* history.
- *Why* questions, whose answers generate *cognitive* history and *motivational* history.

In [Brush et al., 2002], asynchronous awareness mechanisms are classified in three categories:

- *Informational* awareness mechanisms, where details of past activity and events are shown or can be queried from a specific workspace area.

- *Subscription*-based awareness mechanisms, where a user can register an interest to receive specific notifications on certain events in a workspace.
- *Peripheral*-based awareness mechanisms, where widgets are applied to provide awareness at a glance outside the workspace.

In [Pankoke-Babatz et al., 2004], the authors focus on the temporal fragmentation of asynchronous awareness: the point in time when an action takes place and the point in time when an observer perceives it are separated. Consequently, the environment should record awareness information once actions take place and present the awareness information in a way that conveys the situation that happened and that is adapted to the observer's situation. This results in different requirements on data processing and presentation:

- For *short-term* usage in which a high level of detail is requested it might be useful to rely on detailed, *text-based* representations.
- For awareness over a *longer period* of time, *overviews* are requested which convey the information at a single glance; in this case symbols turned out to be helpful.

However, considering that in long-term work processes both the situations mentioned above may occur, awareness support must be flexible and adaptable to both the situation in the setting and the particular situation of the observer and potential actor.

2.4.2 *Relevant Implementations*

Standard groupware and project management tools such as BSCW [Horstmann and Bentley, 1997], CANS [Amelung et al., 2007], ActiveCollab [ActiveCollab, 2008] and TeamWox [TeamWox, 2011] organize asynchronous awareness information on the sole basis of its reference workspace/group/directory.

The POLIAwaC environment includes three modalities for asynchronous awareness delivery:

- The *event bar*, that provides a comprehensive event history, plus the possibility for the users to explicitly generate object-related awareness information. The bar shows by default the latest event that is of interest to a user, while a drop-down list can display the full chronological list of events at request.
- The *event dialog*, that is structured like the event bar, but has a history list that is always expanded.
- The *history window*, that displays the history of an object in textual form, which users can personalize adding or removing related events.

In some collaboration environments, the visualization of the degree of activity within a group, or shared workspace, is proposed as a synthetic form of awareness provision. In [Gross et al., 2003], AwarenessMaps are proposed to provide the members of shared workspaces with an overview of users and documents: "the *PeopleMap* shows an array of pictures of active users fading out over time; and the *DocumentMap* provides a schematic overview of the structure of a shared workspace and indicates recent changes."

In [Zhang et al., 2005], the Info-Lotus, a pictorial representation of incoming e-mails, is introduced. This tool uses a graphical metaphor of a growing flower, where emails are divided in groups and sub-groups in order to represent conversation threads.

In [van Dantzich et al., 2002] the Scope, a radar view of awareness events, is proposed. Events are classified on the basis of their originating application; plus, they are represented as elements in a circular radar-like screen, which is divided into sectors that group different kinds of notifications. The more a notification is placed in a central point, the more important that notification is.

Recently, the research about collaboration in online communities has focused on activity awareness in order to inform users about who is active in the topics of interest of the community, which kind of contribution has been provided, and similar. For instance, Vassileva and Sun [2007] propose a "star" view of users, aimed at showing their degree of activity in the community. Moreover, Baishya and Brusilovksy [2009] propose a visualization of activity awareness in CiteULike, which exploits radial time bands to show the time period during which the user/group activity (or the activity on a topic) has occurred.

2.5 TAG CLOUDS

In [Hearst and Rosner, 2008], tag clouds are defined as *visual representations of social tags, displayed in paragraph-style layout, usually in alphabetical order, where the relative size and weight of the font for each tag corresponds to the relative frequency of its use*. Social tags are keywords that participants of tagging services can associate freely with a particular resource [Sinclair and Cardew-Hall, 2008]; such resource may be just about anything, as tagging services are employed to tag such things as photographs, URLs, podcasts, computer games, music and videos. The dataset that emerges from the union of all tags and resources represents a *folksonomy*.

Tag clouds qualities and functionalities have been explored in several studies [Halvey and Keane, 2007] [Rivadeneira et al., 2007].

Bateman et al. [2008] describe a classification of several visual properties of tag clouds, evidencing their relative importance:

- *Font Size*, *Font Weight* and *Intensity* are very important properties for tags in clouds, as they are recognizable patterns for users and they can catch users' attention easily.

- *Number of Pixels*, *Tag Width*, and *Tag Area* have a strong correlation to other important factors, but, taken by themselves, they are not visual features which capture viewers attention.
- *Colour* and *Position* are important features, that can be identified easily by viewers, but they are to be used with care, as misuses can lead to serious drawbacks.

Sinclair and Cardew-Hall [2008] report that:

- Tag clouds are particularly useful for browsing or for non-specific information discovery, as tags are best suited to broad categorization. They are less precise than a focused search, and therefore weak in supporting the seeking of specific information.
- Providing a visual summary of a dataset, they are a good tool for choosing where to begin the information seeking.
- Users tend to rapidly scan tag clouds, rather than reading them thoroughly.

As a data visualization and research tool, tag clouds have both good and bad qualities:

- They have a compact representation; the eye is captured by the largest elements, that presumably are the most important; three dimensions (words, relative importance, order) can be represented simultaneously.
- It is quite difficult to compare tags of similar sizes; length of words contributes to bigger dimensions, thus partially relating the importance of a word to its number of characters; elements with similar meaning may lie far apart, thus not evidencing their likeliness and possibly suggesting false conjunctions with other elements.

Starting from such mixed qualities, Hearst and Rosner [2008] have conducted a study, whose results show that tag clouds are particularly indicated as a visualization tool for collaborative human activity, as opposed to a visualization used for data analysis; tag clouds are good markers of individual or social interaction with the contents of an information collection, and have good functionalities as a *suggestive* device, rather than as a precise depiction of some underlying phenomenon.

3

CONTEXT-BASED ACTIVITY AWARENESS SUPPORT

3.1 A CONTEXT-AWARE FRAMEWORK

3.1.1 *Context Modeling in an Open Environment*

As discussed by Prinz et al. [2006], the chain production model is evolving towards dynamic collaborations among spontaneously assembled groups of people working together. Moreover, *collaborative tasks are often ill-structured at the outset, emerge in the course of the collaborative process, and need to respond flexibly to changing goals or situations* [Haake et al., 2010]. Furthermore, outside working environments, people engage in simple, loosely-coupled collaborations which do not require the complexity of traditional project management but can benefit from some automatic support.

The ubiquitous environment offered by the Internet brings the capability to carry out different activities from any place, thus enabling various kinds of distributed collaborations. The main problem of this approach emerges when considering that this kind of support is currently offered by separate Web 2.0 services, where each one supports only specific types of activities. A major consequence of this scenario consists in the fact that complex activities and projects, which are necessarily to be handled by carrying on many different kinds of operations, require the adoption of multiple applications; each one of the involved applications might offer a different workspace, that enables users to access the relative information streams and to manipulate the necessary resources in its own peculiar way [Ardissono et al., 2009].

It is therefore evident how the lack of integration of Web 2.0 services can seriously prevent the creation of a fruitful collaboration environment, as it leads to a multiplication of coordination spaces, with different access points to each activity context.

Traditional project management tools (e.g., Collanos [Collanos, 2008] and ActiveCollab [ActiveCollab, 2008]), give a solution to this problem, as they offer a great variety of services and integrate them in a single and stable workspace. However, they are suitable for the execution of stable, well-structured projects, and therefore the need for tools supporting lightweight user cooperation and flexible team management remains unsatisfied.

The need for an open environment, that supports the integration of services in a unified workspace, clearly emerges. Our framework, which we called CONRAD [Ardissono and Bosio, 2011b], acts as a mediator between the user and various Web 2.0 services, providing a cross-application perspective on the information they generate. CONRAD models the user's private and shared activity contexts at different granularities in order to take into account the fact that people engage in dynamic collaborations having different levels of complexity. We introduce three types of activity contexts in order to model such collaborations: collaboration groups, activity frames and tasks [Ardissono et al., 2010a,b].

Collaboration groups are created by users and handled by a specific application, the Group Manager, which is also in charge of synchronizing their definitions and properties across the various software components. This way, the creation and management of collaboration groups is centralized, and the risk of replications and incoherences in groups definitions is minimized.

A collaboration group is internally represented as a tuple (gn, U) where:

- gn is the group name; and
- U is the set of users that belong to the group.

The *activity frame* embodies the notion of project, but in a certain way simplifies it in order to adapt it to lightweight and flexible cooperation models. We can distinguish two basic modalities of use for an activity frame:

- In the simplest case, activity frames are used as containers for sets of related users and artifacts. For instance, a user could define a frame to model a thematic group, used to keep track of the communication with a set of people such as the family.
- Activity frames also support the execution of complex activities and the scheduling of operations in order to meet the requirements of structured collaborations. For instance, consider the preparation of a large report which integrates work carried out by different people, or the organization of a holiday oversea. For this purpose, within a frame, users can define tasks which describe actions at finer granularity levels.

An activity frame is internally represented as a tuple (fn, U, O, Oi, T) where:

- fn is the frame name;
- U is the set of users sharing it (possibly a singleton, for private stuff);
- O is the set of objects explicitly associated to the frame;
- Oi is the set of objects associated to it by means of inferences (see below); and
- T is the set of included tasks.

A *task* supports specific activities, during which users can interact and synchronize themselves in order to achieve a goal; a deadline can be associated to a task. Similar to the activity frame definition, a task can also have a set of associated objects, which can be those artifacts that need to be manipulated in order to fulfill the task's goal. A task normally inherits the sharing specifications of the activity frame containing it, whilst they can be modified adding or removing users as needed. The execution of a task can be further specified by defining children tasks and partial order dependencies among them.

A *task* is represented as a tuple $(tn, U, O, Oi, g, P, T, s, d)$ where:

- tn is the task name;
- U is the set of users sharing it (possibly a singleton, for private stuff);
- O is the set of objects explicitly associated to the task;
- Oi is the set of objects associated to it by means of inferences (see below);
- g is the goal;
- P is the set of tasks which must be closed before starting tn;
- T is the set of children tasks;
- s is its state ($enabled$, $disabled$, $closed$); and
- d is its deadline ($null$ by default).

Our framework assumes the presence of a specific tool for the management of the activity contexts, which is accessible and usable in the collaboration environment. For this purpose, we exploit the Collaborative Task Manager service (CTM) presented in [Ardissono et al., 2010a,b], which is an open and interactive Web-based task manager; developed by exploiting the Jalava web-based diagram editor [Jalava, 2010], the CTM enables the user to create activity frames and tasks, relate them by means of order dependencies (linking them in a graph), share them with other people and manage their life cycle (see Figure 1).

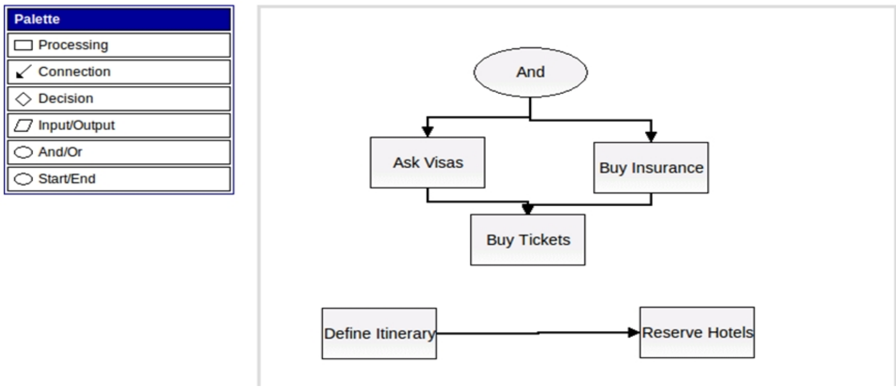


Figure 1: Portion of the User Interface offered by the CTM for the specification and management of activity contexts.

Specifically, the CTM offers a Web page for each activity context c showing the related workspace, with the links to the associated objects (documents, e-mails, etc.) and to the users sharing that context (collaborators). Importantly enough, the CTM can be integrated with business services to support the execution of operations from its User Interface, as it aims at representing a unified access point to the user's workspaces, from which (s)he can handle personal and shared activities:

- The page of the CTM associated to c enables the user to send messages to the involved collaborators, to drag and drop the links to the documents that have to be manipulated within c, and to launch business services in order to create and manipulate objects which are automatically associated to c.

- When the user creates or accesses an object o from the User Interface of an activity context c, the CTM associates o to c by adding the object identifier to the O element of c's tuple. This way, the object is included in the related workspace and can be accessed from it at any time.

The CTM is therefore a powerful tool for the recognition of the user's high-level activities: by using the CTM, the user provides explicit information about the context (s)he is working at. Nevertheless, a user might manipulate an object o by invoking business services without using the CTM. If o has not yet been associated to any activity contexts, the CTM attempts to infer this association by reasoning on the user's collaborations: in this case, if o is shared with the exact same users sharing one (or more) of the user's activity frames and tasks, the CTM associates o to such contexts by updating the Oi element of their tuples.

3.1.2 *Contextualization of Events*

Several business services offer APIs for receiving information about the events that occur during their execution. For instance, if a client application authenticates on Google Documents [Google, 2010b] with a user U's credentials, it can poll the service to receive events describing who shares U's documents and how such documents are manipulated by other users. Such events specify, e.g., that U, or another user, has uploaded/shared/removed a document at a certain time.

Most applications manage resource sharing and collaborative operations by listing the involved users (e.g., the targets of an e-mail message). However, this information is not enough to identify the context in which the operations are performed. For instance, if the users that share a document participate in two collaborations, the document and its manipulation events could belong to any of them; the reference activity context of the actions is therefore ambiguous.

This issue can be addressed by offering an automatic support for specifying the context of the users' operations across applications. In our case, we exploit the Collaborative Task Manager to support the execution of business services within specific contexts (activity frames and tasks), and thus use it to classify awareness events accordingly.

In our framework, we assume that each business service that is integrated in the collaboration environment is wrapped by an *adapter* (see Section 3.4.1), which collects the events that are generated by the service and translates them into a standard representation format, that is used within the collaboration environment. This format is basically a list, composed of <feature, value> pairs, that specifies the event data. For instance, when a user uploads a document, an event that comprises the following list of pairs is generated:

- the unique identifier of the event (<ID, $identifier$>);
- the application that originated the event (<application, GoogleDocs>);
- the operation that has been performed (<operation, document-upload>);
- the actor that performed the operation (<actor, $user-account$>);
- the name of the artifact that has been manipulated by the operation (<document, $document-URL$ >);
- a time stamp of the event <time, $time-stamp$>.

Given such a representation, the CTM *contextualizes* the event, associating it to one or more activity contexts, by adding a suitable <feature, value> pair to its description. Specifically, each event can be associated to a set of activity contexts explicitly or by means of inferences on the performed operation and on its objects.

The *explicit* association of an event to a set of contexts ($contextList$) is specified by adding a <contexts, $contextList$> pair to the event description. This type of association occurs in the following situations:

- The event has been generated by an application which handles contexts and classifies events. For instance, workflow management systems support the specification of processes and hierarchical tasks that can be mapped to CONRAD activity contexts.

- The application does not handle contexts but the event concerns an object explicitly associated to an activity context (i.e., the object is referenced in the O component of the context tuple). For instance, if a document doc is explicitly associated to a context c, the document-update events concerning doc can be explicitly associated to c, as well.

- Users can provide feedback about the context of an operation by correcting the classification of the awareness event they receive as notification (see Section 3.2). Even though this kind of information becomes available only after the system has notified the user, it can be exploited to classify the objects involved in the event, thus refining the specification of the user's activity contexts. In turn, this improves the system's capability to contextualize the following events.

The *implicit* association of an event to a set of activity contexts is specified by adding an <inferred-contexts, contextList> pair to the event description. In this type of association inferences on the user's operations have to be made, leading to an uncertain classification. The CTM can implicitly associate events to contexts in various situations:

- If an object o is implicitly associated to a context c (i.e., it is referenced in the Oi component of the tuple describing c), the events concerning o can only be implicitly associated to c.

- If an event involves a list of users L (e.g., the targets of an e-mail message), it can be implicitly associated to all the activity contexts involving the same set of users, similarly to what is done for the classification of objects.

Notice that, if an event is explicitly (implicitly) associated to a task which is part of an activity frame/task hierarchy, "contexts" ("inferred-contexts") includes multiple items corresponding to the path between the root of the hierarchy and the current task.

3.1.3 *Recognizing the Current Context*

A context-dependent awareness framework must, at each instant of time, be able to recognize the particular context (or set of contexts) which the user is focusing on.

We define the user's *current focus of attention* as the list of activity contexts (s)he is handling while (s)he performs an operation; an operation is a generic action, performed by a user interacting with a generic business service, that generates an awareness event.

In an open collaboration environment, it is not possible to assume that business services track the user's focus of attention: most services do not even model contexts. Thus, in CONRAD, this type of information is inferred by analyzing the awareness events generated by the user's operations and contextualized by the Collaborative Task Manager.

The current focus of attention (*CF*) may include zero, one or more activity contexts, depending on the classification of the observed awareness events. For example, CF might include more than one element because the user is focusing on a task (in which case CF reflects the task/activity frame hierarchy) or because the user's focus of attention is ambiguous.

The current focus CF is handled as follows:

- $CF = \{\}$ when the user starts a session in the collaboration environment because there is no information about what (s)he is doing.
- *CF* evolves on the basis of the occurrence of awareness events whose actor is the user, reflecting the most recent behavior. Specifically, for

```
updateCF(ev) {

  //CASE A: ev is explicitly associated to some activity contexts
  if (ev.contexts != null) {
    CF = ev.contexts;
    HCF = null; // reset temporary focus shift hypothesis
    }

  //CASE B: ev is implicitly associated to some activity contexts
  if (ev.contexts == null && ev.inferred-contexts != null) {
    in = intersection (CF, ev.inferred-contexts);

    if (in.size()>0); // ev is consistent with CF -> CF is not updated

    if (in.size()==0) { // otherwise a focus shift is hypothesized
      if (HCF == null)
        HCF = in; // store focus shift hypothesis for later consideration
      else {          // compare ev.inferred-contexts to HCF
                      // and update CF only if the focus shift is consistent
                      // w.r.t. the one hypothesized for the previous event
        focusTransition = intersection (HCF, ev.inferred-contexts);
        if (focusTransition.size()>0) {  // ev supports hypothesis
          CF = focusTransition;
          HCF = null;
        }
        else HCF = ev.inferred-contexts; // ev does not support hypothesis:
                  // -> update temporary focus shift hypothesis again
      }
    }
  }

  //CASE C: if contexts and inferred-contexts are null, CF is not updated
}
```

Figure 2: Algorithm for updating the user's current focus of attention. The algorithm is written in pseudo-code and the dot notation is used to refer to the event features.

each awareness event *ev* generated by a user's operation, the CF of the actor is updated according to the algorithm shown in Figure 2.

The evolution of the current focus of a users can happen in three different ways, whose modalities are determined by the presence (or by the absence) of a contextualization pair in the last performed event's description.

- If *ev* is explicitly associated to some contexts ("contexts" = $\{c1, ..., cn\}$), then CF is set to $\{c1, ..., cn\}$ because the event classification provides strong evidence about a focus shift.
- If *ev* is implicitly associated to some contexts ("inferred-contexts"= $\{c1, ..., cn\}$), we can distinguish two situations:
  - If $\{c1, ..., cn\} \cap CF \neq \emptyset$, CF is not updated with the new information because the event classification, although uncertain, is compatible with the user's recent behavior.
  - Otherwise, the event classification can only be explained hypothesizing a focus shift. However, as the hypothesis is uncertain, CF is revised only if the following user's event supports the same inference. This way, the fluctuations of CF and of the system's notification behavior are reduced.
    In order to support a revision of CF with "one-event delay", the hypothesized focus shift is stored in a temporary HCF variable for subsequent comparison. When the next event is received (HCF is not empty), the algorithm is repeated: if the new event carries explicit information about the context of the user's operations, or it confirms the previous focus shift hypothesis, CF is updated accordingly. Otherwise CF is not modified but the temporary focus shift hypothesis is updated to reflect the new evidence about the user's operations. See Figure 2 for details.

- If ev is not associated to any activity context, CF is not modified because of the user's recent behavior is too ambiguous to support any hypothesis.

3.2 SYNCHRONOUS AWARENESS INFORMATION MANAGEMENT

3.2.1 *Notification Policies*

The CONRAD framework supports synchronous awareness information management with real-time, synchronous notifications about the events occurred in the user's activity contexts. Such notifications can be filtered according to different policies to meet individual user's preferences. The user can select one policy as a default in order to apply it to all of the activity contexts. Moreover, (s)he can override the default on specific contexts to specify personalized notification management rules.

The first two policies are standard ones and enable the user to decide which events (s)he wants to be notified about in absolute terms; the third and fourth policies, specific of our work, support the context-dependent filtering of notifications on the basis of the user's operations:

- With the *Total filter* policy, all the notifications from the specified activity context are filtered out.
- With the *No filter* policy, all the notifications from the specified activity context are submitted as soon as they are generated by the services.
- With the *Context filter* policy, only the notifications from the activity frames in the user's current focus of attention are submitted. When the focus changes, a summary notification is generated: the message shows the number of deferred notifications and a link to the awareness space, where they can be inspected.

- With the *Task filter* policy, only the notifications from the tasks in the user's focus of attention are submitted. When the focus of attention changes, a summary notification is generated which reports the number of deferred notifications and a link to the awareness space. If the current focus of a user does not contain any task, the policy behavior changes to context filter.

If an awareness event is associated with a set of hierarchically related activity contexts, the policies are applied to the most specific one in order to focus on the information concerning the user's current task rather than the high-level activity (s)he is carrying out.

Given the user's focus of attention CF, the "Context filter" policy is applied to an awareness event as follows (the "Task filter" policy is analogous but is applied to the tasks in CF):

- If the current focus is empty ($CF = \{\}$), the event is filtered out, because of lack of information about the user's activities, that makes it impossible to identify a context to which the event classification could be compared. This is a rather conservative strategy, but considering that the user has explicitly chosen a filter policy ("Context" or "Task"), it might be the safer choice.

- If the current focus is non-empty, CF is compared to the event classification. If $CF \cap$ "contexts" $\neq \emptyset$ or $CF \cap$ "inferred-contexts" $\neq \emptyset$, then the event is submitted as a notification. Otherwise, it is filtered out.
  The rationale behind considering the intersection between the user's current focus and the reference contexts of the event is that the event might be relevant to the user's activities in one or more of such contexts.

- If the event cannot be associated to any context (i.e., "contexts" = "inferred-contexts" = {}) it is filtered out, unless the user has selected the

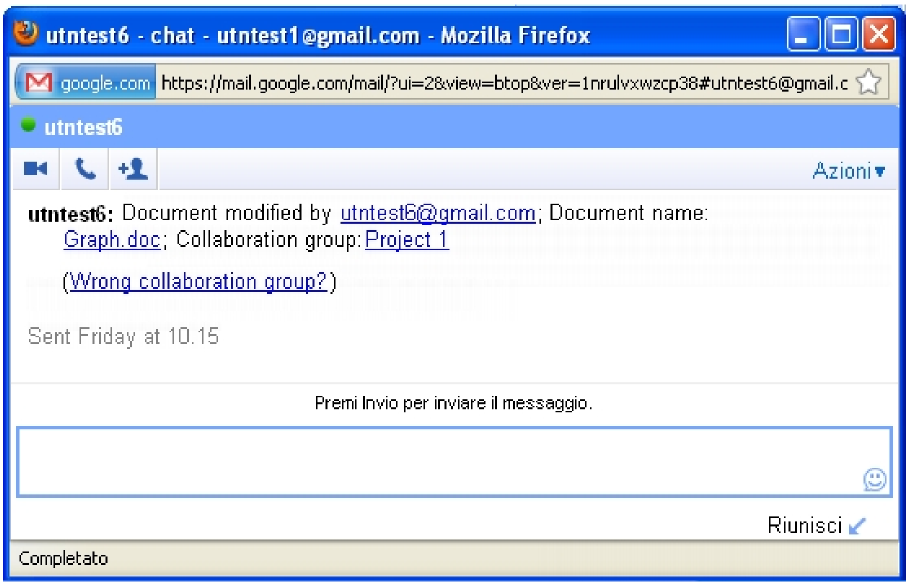


Figure 3: A notification message.

"no filter" policy as a default, because there is no available information about the context of the event.

3.2.2 *Presentation of Awareness Information*

In Section 3.1.2 we have presented how the descriptor of an awareness event stores information about the application which generated the event, the performed operation, its actor, parameters (if any), time-tag and context information. Starting from the event descriptors, CONRAD generates their external format, that is to be used for presentation as notifications and in the awareness space (see Section 3.3).

Figure 3 shows a notification generated by CONRAD to inform the user about a Google Documents event. The content of the notifications has been defined according to the principles described below:

- *Integration with the collaboration environment:* the User Interface of the awareness support tool should enable the user to interact with collaborators, access artifacts and open the workspace(s) that are related to the activity context(s) which the notification is about, by means of a click. Thus, if the application generating the awareness event provides references to the involved entities, they should be included in the notification as hypertextual links. For instance, in Figure 3, both the actor (`utntest6@gmail.com`) and the edited document (`Graph.doc`) are linked and directly accessible from the notification window. This solution has the disadvantage of requiring an extra effort from the user, which has to click on the relative objects in order to fully explore the content of the notification; conversely, it has the advantage of keeping the notification area compact and easy to read, leaving to the user the choice to access related artifacts when necessary while allowing him to see the content of the event at a glance.

- *Transparency:* in order to make the user aware of the system's behavior and inferences, each notification must show the reference activity context(s) of the described event (see the "Collaboration group" part of Figure 3). It is worth to note that, when the "inferred-contexts" feature is used to manage an awareness event, the event can be misclassified and incorrectly handled by the awareness support tool. The presentation of this type of information in the notification message is aimed at supporting the recognition of this type of mistake.

- *Collection of user's feedback (mediation of ambiguity):* if the user believes that the system has misclassified some awareness events (either generated by her/his actions, or by actions that have been performed by other actors), (s)he can correct the system by clicking on the `Wrong [context type]?` link of the notification. The user's feedback is taken into consideration to refine the specification of the involved activity context(s), which are the basis for the classification of the following events.

The notifications are normally handled as Instant Messages and presented in pop-up windows. Other presentation forms could be adopted, e.g., to deliver notifications as e-mail messages, depending on the user's preferences.

3.3 ASYNCHRONOUS AWARENESS INFORMATION MANAGEMENT

The CONRAD framework supports asynchronous awareness information management with a view on awareness information, which can be focused on the recent past and on specific information needs. For this purpose, we designed a three-layered awareness presentation model [Ardissono et al., 2011a,b].

- The *higher layer* visualization gives the users a general overview of their collaborations, designed as a tag cloud. Users can inspect each element that is present in the cloud by cliking on them, and therefore accessing to the middle layer.
- The *middle layer* visualization enables the user to view the detailed awareness information from the perspective of a specific contextual element (collaboration group, activity frame, task, user; see Section 3.1.1) and to possibly browse the hierarchy of contexts; this visualization always contains a button, that permits users to access to the lower layer.
- The *lower layer* visualization presents the long term history of all the user's activity contexts.

Figure 4: Higher visualization layer: Awareness Cloud of a user of a collaboration environment (user `utntest1@gmail.com`).

3.3.1 *Higher Layer: Awareness Cloud*

The higher presentation layer is aimed at providing the users with an overview on the state of their activity contexts, possibly focused on a specific time interval (e.g., during the last two days or after the last *catch up* with such contexts). The design of this layer has been driven by two main requirements:

- Providing a synthetic presentation of the state of the users activity contexts in order to reduce the overloading effect. In fact, if users engage in many different activities, the number of contexts whose state is visualized can be rather large.
- Helping the users to quickly identify the contexts deserving attention.

Concerning the first requirement, we decided to synthesize the state of each activity context as the relative number of awareness events occurred in the time interval selected by the user. We chose this type of information as a starting point because it supports the identification of the contexts that evolved the most in the recent past, and thus might deserve the user's attention.

The second requirement led us to select the tag cloud visualization model for the presentation of information at this level; in fact, we know from literature that, especially for open-ended searches, tags are known for immediately evidencing the most relevant contents, thanks to the large visual differences of such contents from other elements [Sinclair and Cardew-Hall, 2008, Bateman et al., 2008].

We thus designed this layer as an Awareness Cloud showing the degree of activity occurred in the user's private and collaboration contexts. The nodes of the cloud represent activity contexts and actors in order to enable users to monitor the state of their collaborations from different viewpoints:

- how many events occurred in a specific activity context; and
- how active was an actor in the collaborations shared with the user.

As shown in Figure 4, the Awareness Cloud for a user U (`utntest1@gmail.com`) is presented in a Web page. The top of the page enables the user to configure the cloud:

- The form at the right (Month, Day, Year) can be used to specify the starting date of the Awareness Cloud. If such form is filled in, the event history starting from the selected date until the current time is used to generate the cloud.
- The `CATCH UP!` button at the left enables the user to refresh the cloud by setting the starting time to the current time. If the user does not specify the starting date, the cloud is generated using the time of the last catch up.

The lower portion of the page displays information about the user's activity contexts in the selected time interval.

- The nodes represent four types of entities: *user* nodes are associated to U's collaborators; *collaboration group*, *activity frame* and *task* nodes are associated to the user's collaboration groups, activity frames and tasks, respectively. For instance, in the sample Awareness Cloud of Figure 4, CLAUDIO is a user node and LAVORO A (work A) represents a collaboration sphere. In order to help the user to distinguish user nodes from context nodes, the former are in *italics*. In contrast, all types of context nodes have the same font style because the cloud abstracts from hierarchical details concerning the user's activity contexts (collaboration groups can include activity frames, which in turn can include tasks).
- The relative size of each node in the cloud represents the degree of activity in the selected time interval and depends on the number of associated awareness events that have been collected in the collaboration environment. Notice that user nodes visualize the degree of activity of the represented users *within U's activity contexts* because the operations performed by users in other contexts, to which U has no access, must not be disclosed.

For each user, a dynamic awareness cloud is generated, which reflects the activity contexts (s)he engages in and the selected time interval.

In order to enable the user to quickly inspect the details (s)he is interested in, the nodes of the cloud are direct access points to the awareness information presented in the middle layer. Each node is linked to a view on awareness information which shows the related awareness events; e.g., all the events concerning PRENOTARE ALBERGO (Book hotel, see Appendix A) since the last catch up.

The Awareness Cloud includes a maximum number of 40 elements to be visualized at each time because, as discussed in [Bateman et al., 2008], a cloud with too many tags can be puzzling and hard to read. Should more than 40 elements be eligible for visualization, those with least elements (the "smallest" ones) would be dropped. Users can however personalize the cloud by suppressing nodes they are not interested in. We also plan to extend the cloud generation model in order to allow the user to specify high-priority nodes, associated to users and/or contexts to be monitored with particular attention. Such nodes will not be dropped from the cloud and will be depicted in a different color for easy identification (See Chapter 6).

3.3.2 *Middle Layer: Hierarchical, Browsable Views on Awareness Information*

The middle layer of our visualization model enables the user to view the detailed awareness information from different perspectives. Each view is a Web page generated by taking into account a specific context, which refers to its source node in the Awareness Cloud, and the time interval selected for the generation of the cloud. Specifically:

- A view linked from a user node displays the events describing the actions performed by the represented user in the activity contexts shared with the current user.
- A view linked from a context node displays the events occurred in that context.

Figure 5 shows a view presenting the events that are associated to node Lavoro A of the Awareness Cloud: events can be sorted by date, activity context and task, or by actor (if such information is provided by business services).

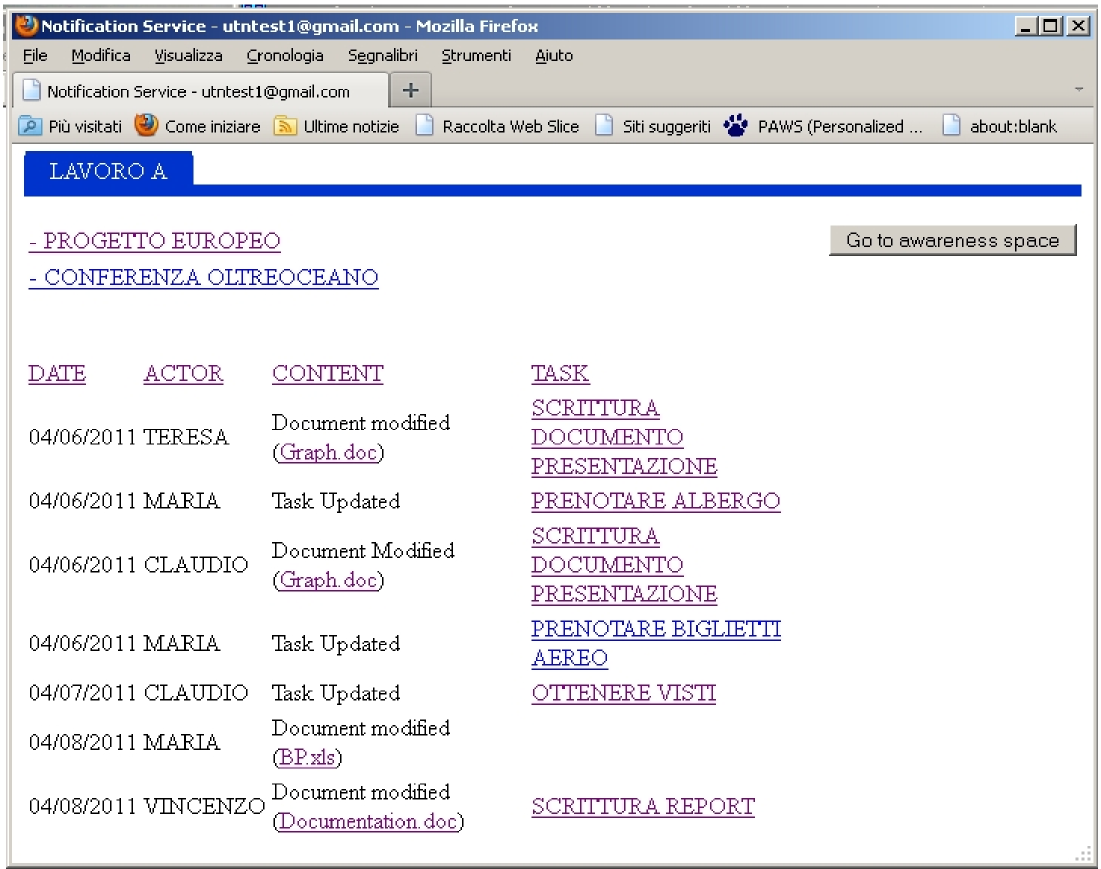
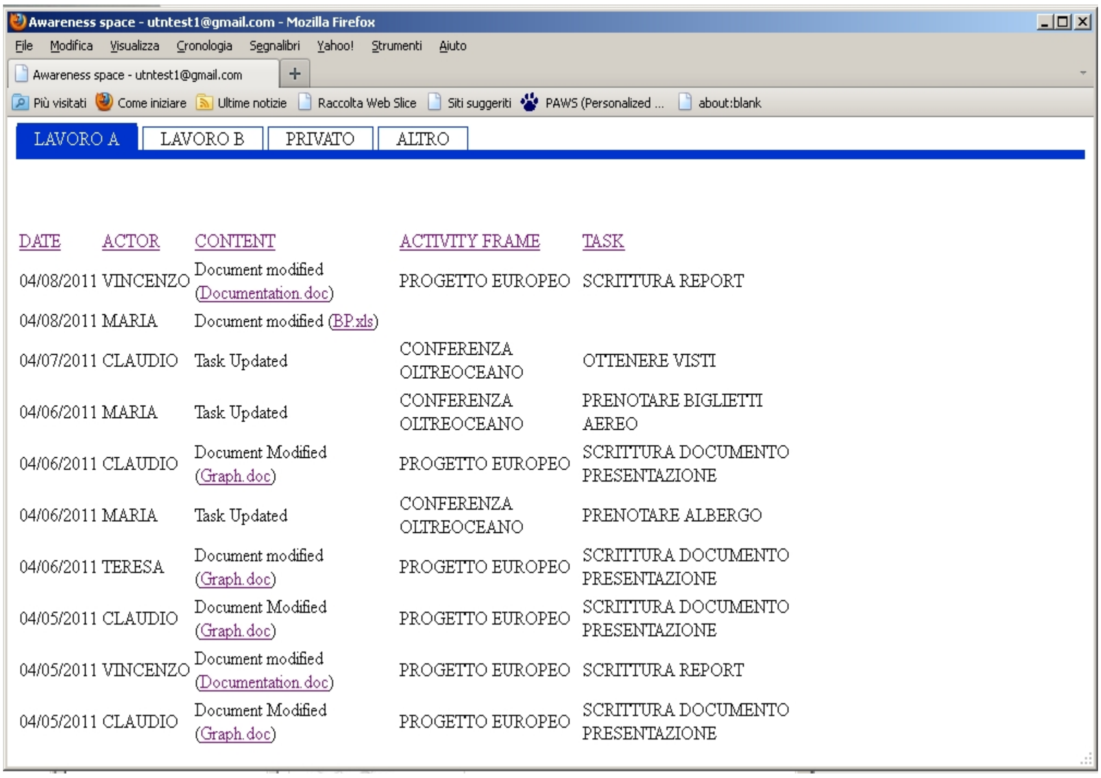


Figure 5: Middle visualization layer: Detailed view on awareness information focused on context Lavoro A.

As activity contexts can be nested, the middle visualization layer supports a hierarchical navigation of awareness information, that enables the user to visualize details about activity contexts at different granularity levels. For instance, as the Lavoro A context includes the Progetto Europeo and Conferenza Oltreoceano activity frames, its view includes two links (located in the top-left portion of the page) pointing to other middle layer views relative to such contexts.

The views of the middle visualization layer also include a `GO TO AWARENESS SPACE` button (see the top-right portion of Figure 5) which can be used to



DATE	ACTOR	CONTENT	ACTIVITY FRAME	TASK
04/08/2011	VINCENZO	Document modified (Documentation.doc)	PROGETTO EUROPEO	SCRITTURA REPORT
04/08/2011	MARIA	Document modified (BP.xls)		
04/07/2011	CLAUDIO	Task Updated	CONFERENZA OLTREOCEANO	OTTENERE VISTI
04/06/2011	MARIA	Task Updated	CONFERENZA OLTREOCEANO	PRENOTARE BIGLIETTI AEREO
04/06/2011	CLAUDIO	Document Modified (Graph.doc)	PROGETTO EUROPEO	SCRITTURA DOCUMENTO PRESENTAZIONE
04/06/2011	MARIA	Task Updated	CONFERENZA OLTREOCEANO	PRENOTARE ALBERGO
04/06/2011	TERESA	Document modified (Graph.doc)	PROGETTO EUROPEO	SCRITTURA DOCUMENTO PRESENTAZIONE
04/05/2011	CLAUDIO	Document Modified (Graph.doc)	PROGETTO EUROPEO	SCRITTURA DOCUMENTO PRESENTAZIONE
04/05/2011	VINCENZO	Document modified (Documentation.doc)	PROGETTO EUROPEO	SCRITTURA REPORT
04/05/2011	CLAUDIO	Document Modified (Graph.doc)	PROGETTO EUROPEO	SCRITTURA DOCUMENTO PRESENTAZIONE

Figure 6: Lower visualization layer: Awareness space of user `utntest1@gmail.com`, focused on tab Lavoro A.

access the awareness space of the collaboration environment (lower visualization level).

3.3.3 *Lower Layer: Context-dependent Awareness Space*

The lower level of our model is the awareness space of the collaboration environment, which presents the long-term event history concerning the users activity contexts. In order to help the users in the navigation of such information, this space offers a different tab for each collaboration group; within a tab, events can be sorted by date, actor, content and activity context. Figure 6 shows a tab of the awareness space for the Lavoro A collaboration sphere of user `utntest1@gmail.com`. It is evident how the event list shown in this page is longer than the one reported in Figure 5: this is due to the fact that the awareness space is not subject to time constraints.

3.4 TECHNICAL SPECIFICATIONS

3.4.1 *Architecture*

Our awareness framework is based on the Personal Cloud Platform (PCP) [Ardissono et al., 2009], which supports the integration of heterogeneous software components (such as Web Services, Web applications and legacy software) in a unified environment, offering single sign-on and a cross-application management of shared activities. The collaboration support that the PCP offers is based on the following core components:

- For each user registered in the environment, a Group Manager instance supports the creation and management of collaboration groups. Such specifications are injected in the software components that are integrated in the environment (if technically possible), in order to synchronize them without human intervention.
- For each user, a Collaborative Task Manager instance supports the specification and management of activity frames and tasks.

While different Group Manager (or CTM) instances synchronize with each other with respect to the overall set of collaboration groups (activity contexts) defined in the environment, each instance provides its user with a personal view on them, reflecting her/his workspaces and access rights.

The PCP supports a loosely-coupled interaction among software components based on the Publish and Subscribe protocol [Wikipedia, 2010]. The

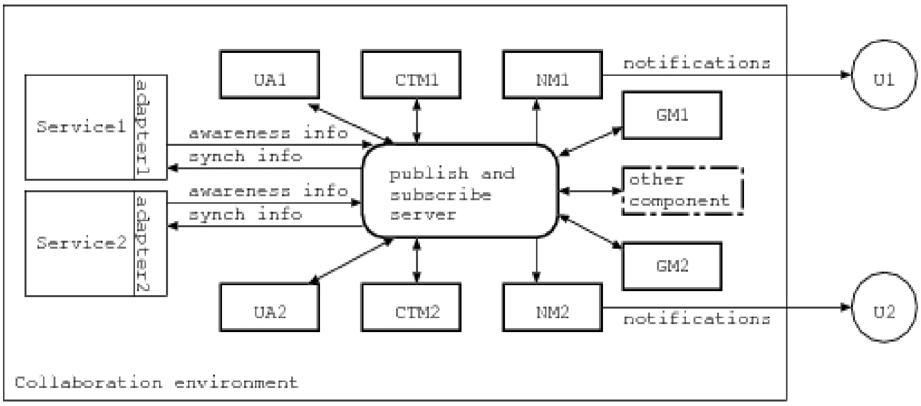


Figure 7: Architecture of a collaboration environment based on the PCP and CONRAD. The architecture is depicted in a simplified form for readability purposes.

propagation of information from one software component to another is decoupled as follows:

- The component which generates the information publishes it by invoking a Publish and Subscribe server, which acts as a hub among software components.
- The components that are interested in receiving a certain type of information subscribe to it. The subscription consists in specifying the pattern of features characterizing the interesting information items; e.g., all the events generated by application $myApp$ which are directed to user U.
- When the hub receives a new piece of information, it notifies the subscribed components.

In order to interact with the hub, each software component must be capable of performing subscriptions, publications, and to be notified about events. For this purpose, an *adapter* is demanded to wrap the software component; the adapter is in charge of:

- Subscribing the component for relevant types of events.
- Intercepting the events and messages that have been generated by the component (or polling it in order to retrieve them) and publishing such information using the APIs of the hub. Notice that, as events have to be published in a common format (as lists of <feature, value> pairs, see Section 3.1.2), the adapter also has to translate events and messages to such a format.
- Receiving notifications from the hub and, depending on the type of information, invoking the APIs of the wrapped software component in order to perform the appropriate operations.

The adapter depends on the execution model and communication protocol of the software component. Thus, in order to integrate an external application in a collaboration environment based on the PCP, the human administrator has to develop a specific one.

The CONRAD framework extends the PCP with awareness information support. For each user registered in the environment:

- A *User Agent* instance stores the user's notification preferences and tracks her/his focus of attention.
- A *Notification Manager* instance collects the awareness information directed to the user and manages it, in order for it to be presented in the awareness space and as notifications.

CONRAD exploits the specification of collaboration groups and activity frames as access control lists to constrain the propagation of information in the collaboration environment. Each instance of the awareness management components subscribes for the pieces of information that concern its user. For instance, a user U's Notification Manager only receives the awareness information concerning U and U's collaborations. Moreover, U's User Agent only

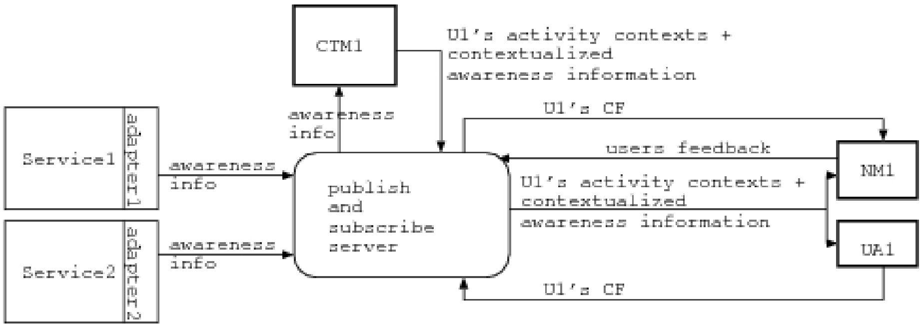


Figure 8: Awareness information flow concerning user U1.

receives the awareness information concerning U's operations on business services.

Figure 7 depicts the architecture of a collaboration environment integrating two external services: `Service1` and `Service2`. The components of the PCP supporting collaboration and awareness are represented as thick rectangles and their names are abbreviated: `CTM` for Collaborative Task Manager; `GM` for Group Manager; `UA` for User Agent; `NM` for Notification Manager. The PCP architecture includes other components, that have not been reported for simplicity.

3.4.2 *Awareness Information Management*

As described in Section 3.1.2, the Collaborative Task Manager service (CTM) classifies the awareness events that are generated by the integrated services in the user's activity contexts and extends events descriptions with the "contexts" and "inferred-contexts" features. Figure 8 shows the data flow among the PCP core components during the management of the awareness information for a user U1 (U1's instances of CTM, Notification Manager and User Agent are denoted as `CTM1`, `NM1`, `UA1`):

- The CTM (CTM1) is subscribed to the awareness information generated by business services. When it is notified from the publish and subscribe hub, it classifies the events in the user's activity contexts; then, it publishes the contextualized awareness information in order to make it available to the other components. The CTM also publishes events which provide evidence about the user's focus of attention; e.g., the fact that the user opens a CTM window to work on a particular task.
- The User Agent (UA1) is subscribed to the awareness events extended with the "contexts"/"inferred-contexts" features and uses them to track the user's current focus of attention CF. Each time the User Agent updates CF, it publishes a "current-focus($user$, CF)" event in order to make this information available to the other components.
- The Notification Manager (NM1) is subscribed to the "current-focus($user$, CF)" events and to the contextualized awareness information. It uses such data to organize the awareness space and to deliver notifications.
- As described in Section 3.2.2, the user is allowed to correct the classification of the awareness events it receives as notifications by clicking on their `Wrong [context type]?` links. If the user makes such a correction, the Notification Manager publishes a "user feedback" event. The CTM is subscribed to this type of event and uses it to correct the specification of the involved activity contexts.

3.4.3 *Implementation Details*

The PCP and the CONRAD prototype are developed in Java by exploiting the Google Web Toolkit (GWT, [Google, 2010c]) to build the User Interface of the applications. The current dashboard is built as an iGoogle page, allowing users to collect the Gadgets of their favorite apps, as well as those offered by the collaboration environment. The PCP components employ the Google Authentication service [Google, 2010a] as a sign-on service. This authentication method is also used to access the business data concerning the registered users; e.g., calendar data. As a publish and subscribe hub, we exploited GigaSpaces [GigaSpaces, 2008], which provides a scalable environment where clients can publish information and subscribe for notification on events (e.g., creation, update, deletion).

Within the PCP, the propagation of information among software components is controlled in the following ways:

- At the network transport level, the transmission of information is secured using the *Secure Socket Layer* communication protocol.

- Within the PCP, the propagation of awareness information is controlled by associating separate instances of the PCP components to each user. Each instance only subscribes for the events directed to its user. Thus, if the business services integrated in the collaboration environment do not violate the privacy of their users, the PCP components only receive information which their users can be informed about.

4

EXPERIMENTAL EVALUATION

4.1 EXPERIMENTAL SETTINGS

We conducted two experiments in order to test our awareness framework: the first one aims at evaluating our synchronous awareness management approach and the second is directed at evaluating our asynchronous awareness management approach.[1].

In order to evaluate our framework, we formulated three hypothesis:

- The first two hypotheses aim at validating the benefits of our synchronous awareness management solution, which we claim can effectively reduce users mental workload when dealing with notifications in a collaborative environment.
- The third hypothesis regards our asynchronous awareness solution, which we think can significantly improve the level of performance of the users when interacting with an activity awareness space.

4.1.1 *Workload Reduction*

Mental workload has been frequently used as a measure for calculating the effects of interruptions: high levels of workload represent a negative

1 The main reason behind the choice of conducting two separate experiments was the need to optimize working times: our prototype is in continuous (and still active) development, therefore we decided to start the testing phase as soon as we implemented the synchronous awareness management system, when the asynchronous part was still in development phase

experience for users [Iqbal and Bailey, 2005]. We tackle the problem of interruptions with context-dependent notification policies (the Context filter and the Task filter): such filters, in our intuition, will significantly reduce the levels of mental workload in our collaborative environment. We also think that the Task filter policy, further reducing the number of notifications and limiting such amount to those that directly impact the current task of the user, should significantly generate less workload than the Context filter policy.

Hypothesis (H1): The introduction of context-dependent notification policies (Context filter and Task filter) will have a positive effect on users' workload on a computer-based, collaborative writing task.

Hypothesis (H2): A stricter interruption reducing policy (the Task filter) will have a positive effect on users' workload on a computer-based, collaborative writing task, compared to a more permissive policy (the Context filter).

4.1.2 *Performance improvement*

We hypothesize a causal relationship between the introduction of the Awareness Cloud on top of an awareness space structured on the basis of the user's activity contexts (henceforth, context-aware awareness space) and people's performance during a task. We decide to measure users' performance by calculating the time they take to solve some information seeking tasks and the number of errors they commit during the same tasks.

Hypothesis (H3): The introduction of an incremental access to a context-dependent activity awareness space will improve users' performance on an awareness information seeking task, in terms of execution times and number of errors.

4.1.3 *Apparatus*

We evaluated the context-aware provision of awareness information offered by CONRAD using a prototype collaboration environment; this prototype was based on the Personal Cloud Platform and included the following business services:

- a document sharing service (Google Documents);
- a calendar management application;
- an e-mail service;
- an instant messaging service (Google Talk);
- a workflow management tool (jBPM [JBoss Community, 2010]); and
- a service supporting the scheduling of meetings [Bosio, 2010].

The prototype was configured for use on a PC laptop (a Dell Latitude E6400 equipped with Windows XP), with the built-in monitor used as the display (14.1-in LCD, 1280x800 pixel resolution). Experimental sessions were recorded with a desktop capture program and clocked by the experimenter. The experimenter annotated all interesting actions and comments by the users while sitting at some distance from them.

4.2 FIRST EXPERIMENT

4.2.1 *Subjects*

Twenty-four volunteers were involved as participants in this experiment (15 men and 9 women). They had a median age of 26 (M = 27.5, minimum

# TREATMENT	1ST CONDITION	2ND CONDITION	3RD CONDITION
Group 1	1	2	3
Group 2	1	3	2
Group 3	2	1	3
Group 4	2	3	1
Group 5	3	1	2
Group 6	3	2	1

Table 1: Treatment conditions.

25, maximum 34) and were students or staff members of the University of Torino. They performed the test for free, without any reward.

4.2.2 *Experimental Design*

The experiment had a single-factor, within-subjects design. Three treatments were applied - two experimental and a base-case control treatment. The experimental treatments consisted in the Context filter and Task filter notification policies while the No filter policy was treated as the base-case.

Each treatment condition was considered as an independent variable. Participants' workload was considered as a dependent variable. Each participant received the three treatments but the order was counterbalanced in order to minimize the effect of practice and fatigue: each participant was assigned to one of six groups that defined the counterbalanced ordering of treatments; see Table 1.

4.2.3 *Procedure*

The experimental task was designed as a document reading, comprehension and elaboration one. The user had to produce a text and integrate it in a document with graphs created by another user, simulating a computer supported collaborative work situation. At the end of each task, the participant was given a five minutes break for resting; the next task started only when (s)he confirmed to be in good mental and physical conditions.

For each treatment, the user was given a two-page electronic document that explained some analytical results. The documents were very similar in structure and they were written in the form of newspaper articles, having a colloquial writing style, but with a rigorous presentation of data (we chose this presentation style in order to limit comprehension difficulties). Even though the documents dealt with same topic, i.e., the health effects of smoking on people, each of them focused on a different aspect:

- risks of cardiac illnesses on young diabetic people;
- risks of cardiac and vascular illnesses on healthy people;
- risks of cigars and pipe smoke compared to cigarette smoke.

The user was told to read and understand each document in order to summarize such results in an abstract of about 20 lines. (S)he was told that, at the same time, a colleague was producing a histogram representing the data analyzed in the document: after this was completed, the user had to include the graph in her/his own document, while ensuring that it was consistent and that it fitted the text (s)he had written.

Reading, comprehension and writing operations were the user's main attentional focus. The user was also told that, when the colleague had completed the histogram, (s)he would have been notified by the system by

# event	act. frame	notification policy
`Document created`	Project 2	No filter
`Document modified`	Project 2	No filter
`Group member removed`	Project 2	No filter
`Confirmed lesson`	English c.	No filter
`Group member added`	Project 1	Context, No filters
`Document modified`	Project 1	Context, No filters
`Document created`	Project 1	Context, No filters
`Confirmed meeting`	Project 1	Task, Context, No filters
`Document modified`	Project 1	Task, Context, No filters

Table 2: Awareness events submitted in the experiment.

means of an Instant Message. Depending on the actual treatment and on the tested policy the user could also receive other notifications concerning her/his area(s) of interest within her/his collaboration groups. Test users were in fact members of three collaboration groups representing different activity frames: two projects at work (Project 1 and Project 2) and an English course.

The document editing task was part of Project 1. Nine events, i.e., operations executed by other people within the user's collaboration contexts, were simulated by the system; the timing and order of events were randomized by the event simulator.

Test users were told that, based on their current notification policy, they would have received some notifications about other users' operations on their activity contexts. They were also told to act naturally and freely, paying attention to these notifications coherently both with their attitudes and with their current status and needs during the experiment: they could therefore choose to deal with notifications the way they thought was the most proficient for achieving their final target. Pop-up notifications were selected as a modality that could assure that interruptions would not go unnoticed, allowing the most direct and complete notification [Adamczyk and Bailey, 2004].

During the No filter treatment, nine notifications, for all nine events, were sent to the user. Five events originated notifications during the Context filter treatment (events related to Project 1 only). Two events originated notifications during the Task filter treatment (events related to user's current task); see Table 2 for details. As described above, notifications of events related to the current task were delivered to the users in all three treatments. The `Document modified` event was indeed the completion of the histogram which the user needed to close her/his task; the `Confirmed meeting` event was the confirmation of a project meeting where the document (s)he was working at would have been discussed.

At the end of the task, the user was asked to perform another operation on a different activity frame: defining a meeting for Project 2. This allowed the system to track the focus change and deliver a notification message which signaled the presence of unread events, if any. The message contained a link to the Web-based awareness space which displays the full list of events, grouped by collaboration group (see Chapter 3.3).

One participant at a time performed this test. The experimenter had a script in order to manage consistently each participant's treatments. Participants were given a booklet of written instructions that contained pictures of the business services and described the treatment conditions. Such instructions

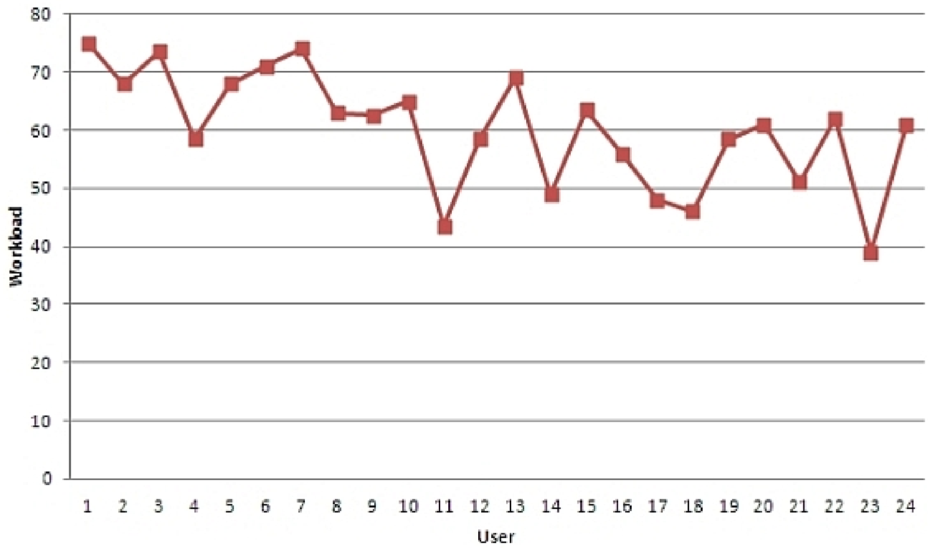


Figure 9: Test results (No filter).

were available as a reference throughout the experiment. Each participant was engaged in the testing activity for a period of about one hour.

4.2.4 *Measures*

Participants' workload was calculated with a modified version of the NASA-TLX [Hart and Stateland, 1988] questionnaire[2] at the end of each task.

4.2.5 *Results*

Figures 9, 10 and 11 show the results of the questionnaire we proposed to the test participants for No filter, Context filter and Task filter, respectively.

2 NASA-TLX questionnaires were chosen as a standard evaluator for users' workload. However, the submitted questionnaire was modified to omit physical effort evaluation, which the experimenter did not consider as relevant in the selected scenario. See Appendix A.

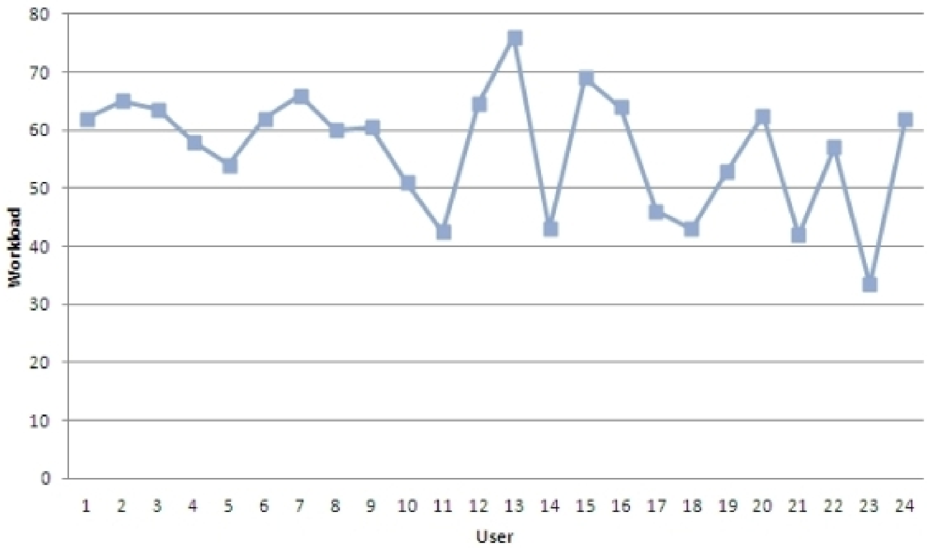


Figure 10: Test results (Context filter).

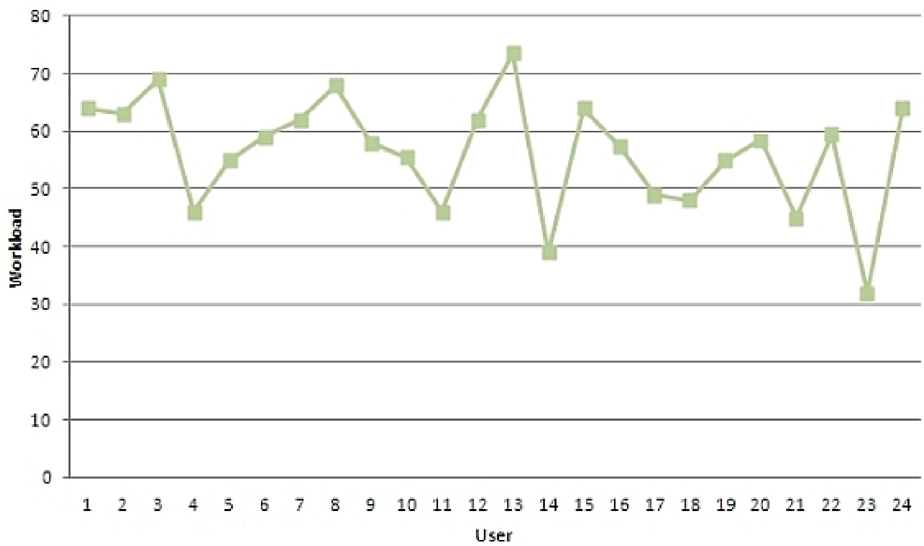


Figure 11: Test results (Task filter).

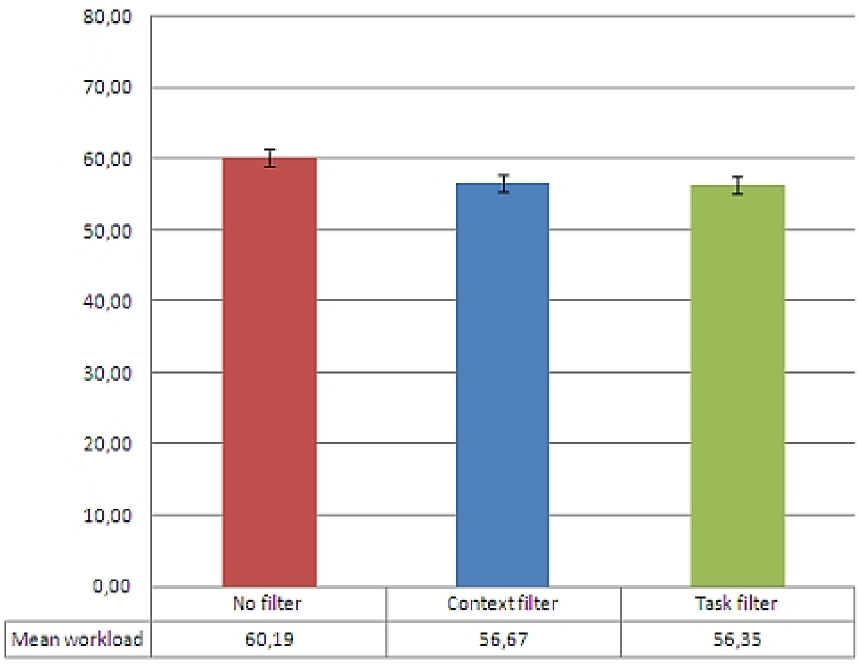


Figure 12: Mean values of workload, for each treatment condition.

Each user is defined on the x-axis with a number ranging from 1 to 24, while the expressed workload is represented in the y-axis.

Figure 12 shows mean values for each treatment condition.

We used Bartlett test to assure homogeneity of variances and Mauchly test to assure sphericity. We selected within-subjects, one-way ANOVA to analyze the collected data and Bonferroni pairwise correction test for post-hoc comparison analysis, using an alpha level of 0.05 to make decisions of significance.

- Bartlett's p-value was $0.97 > 0.05$, so we assume that variances are equal.

- Mauchly test did not show any violations of sphericity against Filter ($p = 0.21 > 0.05$).

- With one-way repeated-measure ANOVA, we found a significant effect of Filter on Workload ($F(2,23) = 7.74$, $p < 0.01$).
- Bonferroni pairwise comparison revealed a significant difference between No filter and Context filter ($p < 0.05$), and between No filter and Task filter ($p < 0.05$). No significant difference was found between Context filter and Task filter ($p = 1.00$).

4.3 SECOND EXPERIMENT

4.3.1 *Subjects*

Sixteen volunteers participated in this experiment (10 men and 6 women). They were students or staff of the University of Torino and performed the test for free, without any reward. Participants had a median age of 26 ($M = 26.5$, minimum 25, maximum 28).

4.3.2 *Experimental Design*

The experiment had a single-factor, between-subjects design. Two treatments were applied - an experimental and a base-case control one:

- The experimental treatment consisted in a context-dependent activity awareness space enhanced with an Awareness Cloud and the middle visualization layer.
- The context-dependent awareness space (lower visualization layer) was considered as the base-case.

Each treatment condition was considered as an independent variable. Participants' performance was considered as a dependent variable and was calculated considering two objective measures: number of committed errors and time needed to complete the assigned task.

Participants were divided into two groups of eight people and each group received a single treatment in order to prevent side-effects such as practice and fatigue.

Users were also given a post-test questionnaire after task completion, which aimed at evaluating qualitatively their opinion on the User Interface solution they had experimented during the test (see Appendix A).

4.3.3 *Procedure*

The experimental task was designed as an information recovering and comprehension one, simulating a typical, asynchronous exploration of awareness information in a collaboration environment. All users were briefed about their scenario before the beginning of the task: as participants of three different collaboration groups, and after a brief period of absence, they had received awareness information regarding other users' activities, that was still to be read.

Such information was collected in a structured list (the awareness space), where each event-related element was organized on the basis of its originating collaboration group.

Each user was given information about the nature of their collaborations, such as names of collaboration groups, activity frames, tasks and involved users. Such instructions were available to participants as a reference during the experiment.

Users were then asked to answer six questions, whose answers could be found by navigating the awareness information events:

1. Which user completed the largest number of elementary operations regarding collaboration group LAVORO A?

2. How many tasks progressed (in terms of completed operations) within collaboration group Lavoro B?

3. Which user completed the last operation on task Scrittura Documento Presentazione in activity frame Progetto Europeo, collaboration group Lavoro A?

4. In which context had the largest number of operations been carried out?

5. Please write contexts, projects and tasks where user Maria completed one or more operations.

6. Which users carried out one or more operations regarding the activity frame Organizzazione Cena, which belongs to collaboration group Privato?

While questions 1, 2 and 4 are general, quantitative oriented, questions 3, 5 and 6 are more specific and require some more exploration effort.

Participants belonging to the control group used a more traditional form of activity awareness space, which is embodied by our lower layer (see chapter 3.3.3). The rationale behind this choice is the fact that this layer, taken in isolation, is a good example of a standard awareness space. Participants who received the experimental treatment could instead exploit our full asynchronous awareness support, with the presence of the higher and middle visualization layers: users of the experimental group could therefore access *projections* on the awareness space by clicking on the nodes of the Awareness Cloud.

Each participant was engaged in testing activity for a period of about 15 minutes.

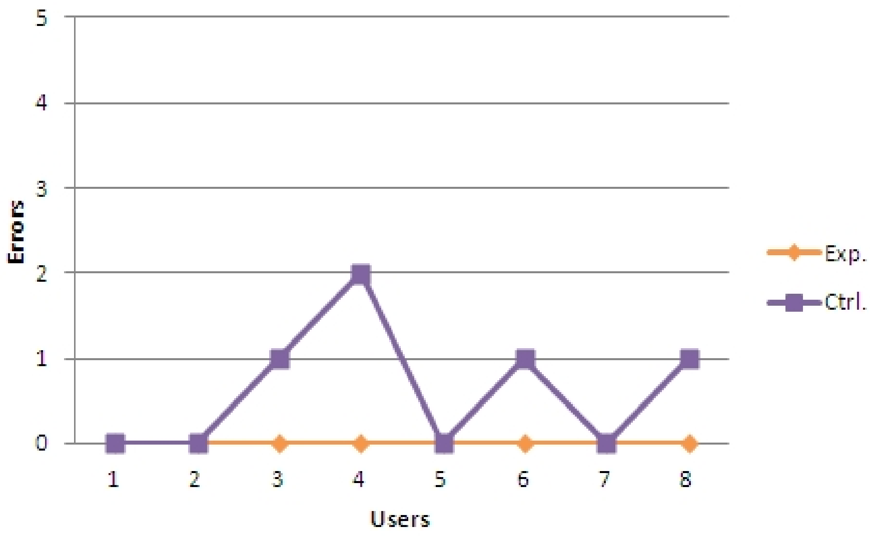


Figure 13: Test results (Number of Errors).

4.3.4 *Results*

Figure 13, 14, 15 and 16 show the results of the user tests.

We used an unpaired Welch's t-test (which does not assume equal variances) to analyze collected data. An alpha level of 0.05 was used to make decisions of significance. We found a significant effect for both number of errors ($t = -2.38$, $p = 0.049 < 0.05$) and execution times ($t = -3.15$, $p = 0.011$).

In the post-test questionnaire we asked users to evaluate their own experience with the User Interface they operated with. Users expressed their opinion relatively to four aspects:

- efficacy;
- efficiency;
- simplicity of use; and
- pleasantness of use.

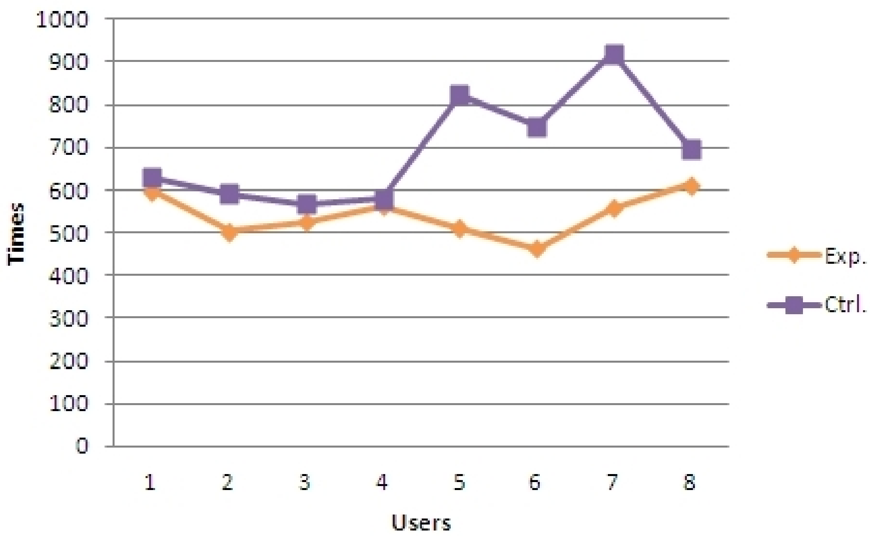


Figure 14: Test results (Execution Times).

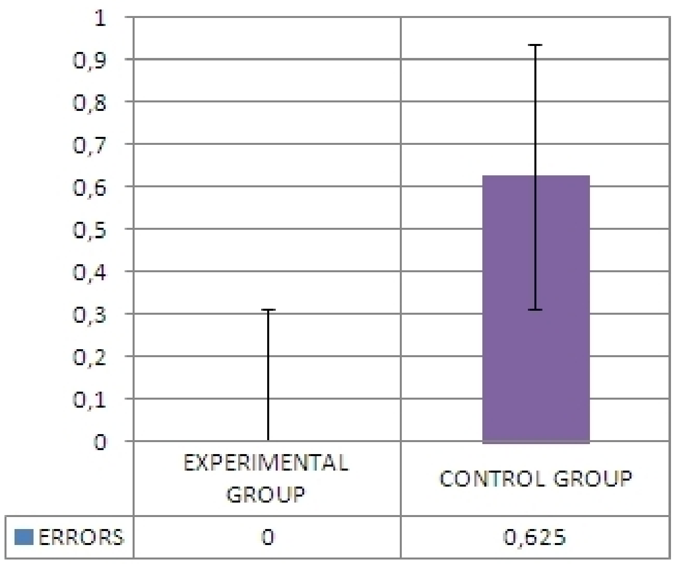


Figure 15: Test results (Number of Errors, Mean Values).

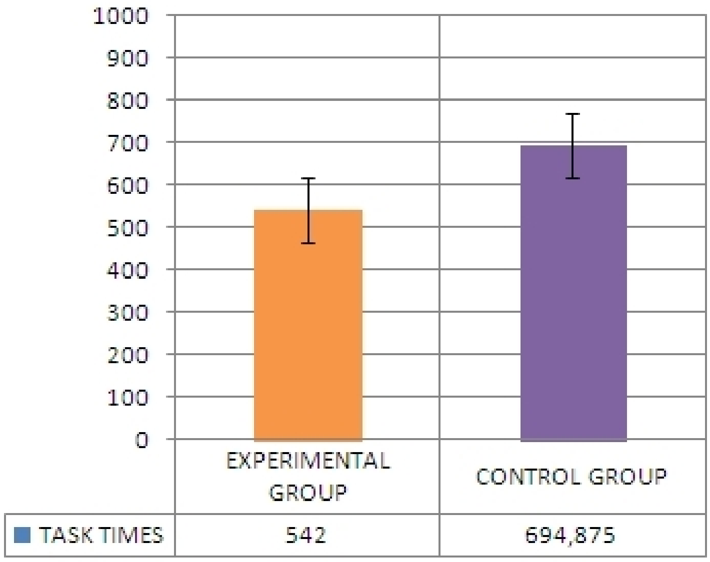


Figure 16: Test results (Execution Times, Mean Values).

Each user could choose between seven non-decreasing levels of satisfaction, ranging from 1 to 7[3]. Figure 17 shows the results of the questionnaire.

Again, we used an unpaired Welch's t-test to analyze collected data, with an alpha level of 0.05 to make decisions of significance. We found a significant difference in two qualities: simplicity (t = -2.376, p = 0.03613) and pleasantness (t = -2.5668, p = 0.03373). No difference were econuntered for efficacy (t = 0.6732, p = 0.5183) and efficiency (t = -0.2125, p = 0.8352).

4.4 ANALYSIS

4.4.1 *Hypothesis H1 and H2*

The results of our first experiment revealed that the context and task filter reduced users' workload with respect to the non filtered condition, thus evi-

3 The full questionnaire is reported in Appendix A.

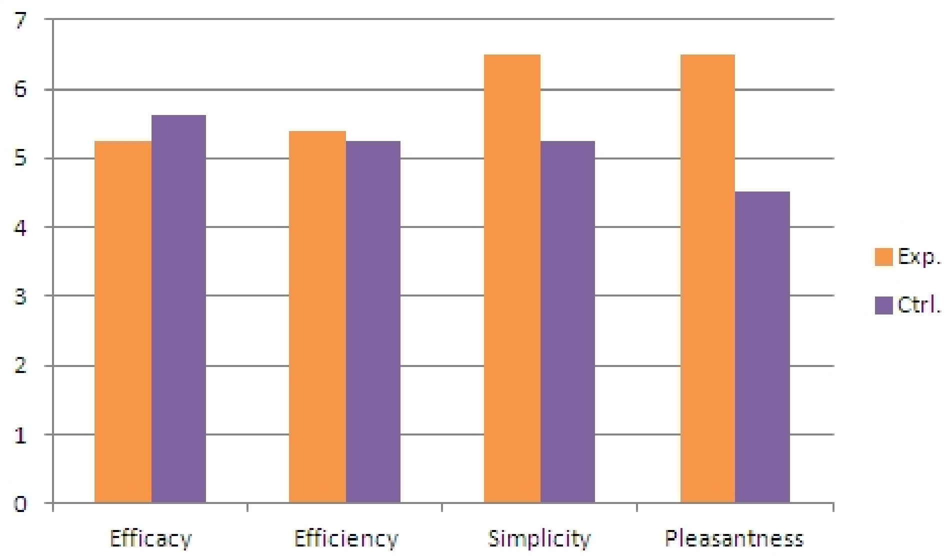


Figure 17: Post-test questionnaire (Mean Values).

dencing how context-dependent filtering could be helpful to users. Therefore, our hypothesis H1 can be confirmed.

Nevertheless we have to discard our hypothesis H2, as none of the two policies performed significantly better than the other. This fact highlights that, with some people, a further contextualized reduction of interruptions, such as the task filter, may not be an immediate solution for obtaining better workload results.

One point that could be argued is that both the context filter and the task filter are some kind of "best" solutions. We know from the literature that individuals may differ in how they control and process information, as well as they handle attention between different tasks.

First-hand observations of participants' behavior confirmed the presence of individualities in task performance: although participants were given the same instructions and had similar backgrounds, differences were observed in their strategies for executing the required tasks. These observations could be useful for the purpose of this analysis because, even though they do not allow the formulation of other statistical claims, they give some interesting insights.

Specifically, we must consider that, as it could be evinced from the description of the experiment, only the histogram completion in PROJECT 1, of all the delivered notifications, had a direct impact on the user's task, as it was necessary to complete it. When faced with the no filter and context filter policies, which delivered respectively nine and five notifications, users enacted different strategies for handling interruptions:

- Some users immediately switched their attention from the task to the notification, read it and then switched back to the main task.
- Under the No filter condition, some users did not pay attention to any notification: they reduced each opened window to the desktop bar and continued with the main task. They read the notifications later on, searching for the "completed histogram" message.
- Under the No filter condition, some users switched from the first strategy we described above to the second one, probably frustrated by the lack of immediate utility of the early notifications they received.

Having observed such different types of behavior, filtering notifications with respect to the current task seems to be a desired approach especially because, when the deferral was not system-enacted, users tried some strategies to defer attention switches. We believe that this aspect might assume great importance in real world scenarios, characterized by larger activity contexts and significant numbers of notifications, possibly hardly related to the user's current activities.

4.4.2 *Hypothesis H3*

The results of the second experiment revealed that the incremental access to awareness information significantly improved users' performance, in terms of execution times and number of errors. Our hypothesis H3 is therefore confirmed. First-hand observations of participants behavior led us to grasp two aspects that may explain these results:

- Our higher layer, the Awareness Cloud, proved itself as very easy to understand and to use, and showed a good level of integration with the awareness space. Indeed, the users of the experimental group were left free to choose arbitrarily whether to adopt it or not, but every one of them (even those who did not know what a tag cloud was) opted for its use since the first question.
- The Awareness Cloud also allowed users to express fast and precise queries by clicking on the desired nodes and accessing the corresponding middle layer.
- Navigating into the lower layer of the awareness space in isolation did not prove itself as immediate and error-free: users of the control group who did not commit errors spent more time doing their tasks, probably due to the need of verifying their answers with more accuracy.

Results coming from the proposed questionnaire give us some other insight about the qualitative improvement that are brought by our approach:

- Users seemed satisfied by both tested solutions, in terms of efficacy and efficiency (i.e., the capability to complete the assigned tasks with the given resources and with a minimal amount of effort). This might be explained by the fact that, although some of the users who used the lower layer alone committed some errors, or took a longer time to complete their task, they did not find particular difficulties in achieving their results.

- Simplicity and pleasantness of use, instead, emerge as distinctive qualities of our layered approach: some users, during some informal talking, described our solution as "practical, good and interesting", and the results we got from the questionnaires seem to confirm this opinion. Users looked in favorable way at the Awareness Cloud, stating that it could give a good and intuitive representation of the underlying activity awareness elements.

4.5 threats to validity

A few threats could be identified with regards to our empirical evaluation processes. The following discussion addresses them and describes the limitations of our work.

4.5.1 *First Experiment*

The within-subject design of the experiment might in principle lead to the comparability of tasks across the three conditions. However, we think that counterbalancing the order of treatments has significantly reduced the problem.

Some users might have ignored their collaborators during the experiment by sequentially completing each piece of work. In order to prevent this type of behavior, we planned the timing of the events in such a way that, in several cases, the interruptions occurred when the test user was reading his material (comprehension stage).

An open issue, to be addressed in our future work, is the fact that individual differences and presentation methods could influence the workload. We know from the literature that individuals may differ in how they control and process information, and in how they handle attention between tasks. Individual differences can therefore have a relevant impact on users' behavior when performing an interrupted task [Latorella, 1999]. Nevertheless, at this development stage we tried to carry out a user test that could be as general as possible.

As discussed in literature ([Bartram et al., 2003]), the introduction of different presentation modalities, coupled with the recognition of the user's focus of attention and its relation with the notification, might improve the user's experience leading to different workload values. In our future work, we will investigate whether differences in users' cognitive style, self-efficacy, desire for control, information processing and attention can cause different preferences in notification modalities.

Finally, it should be noticed that, in order to measure our dependent variable, we selected the expressed workload, which is subjective and self-evaluated. Although we consider such variable as a valuable measure for our goal, in our future research we plan to also consider other, objective measures, such as the time users need to complete a task and the number of occurred errors.

4.5.2 *Second Experiment*

The experiment is based on a simple procedure of six questions, that might not cover the wide range of possibilities in which past events are processed by users within an asynchronous awareness support. The data we collected confirms our hypothesis for the tested situations, although we recognize that there might be some other important elements still missing from the scope of our current evaluation process. As an example, it could be interesting to test how the information that the user collects is exploited within his operational contexts: in other words, we plan to measure how the asynchronous awareness space improves future operations of the users.

It is also difficult to cover the wide, possibly infinite variety of modalities in which past events can accumulate within an awareness space. This experiment featured a single cloud, but it is possible that users might react differently to differently shaped clouds, as these are formed by events whose quantity and nature are constantly subject to modifications as the workspace activities take form. A field test, possibly covering a sufficiently long time span, might give more precise answers to these questions, with respect to the scenario based experiment we have realized this far.

5 COMPARISON WITH RELATED WORK

5.1 MOTIVATIONS

While projecting and developing CONRAD, our main focus has been the provision of awareness for flexible types of collaboration, as this feature is poorly provided by groupware and project management tools such as Collanos, ActiveCollab and TeamWox [TeamWox, 2011]. The notion of project modeled by such tools is closely related to traditional project management, while our attention is devoted to the dynamic types of collaboration that are emerging in modern working scenarios [Prinz et al., 2006].

Our approach is far more suitable for dealing with ordinary collaborations in informal environments, such as the establishment of ephemeral collaboration groups; such kind of groups are largely aimed at achieving short-lived tasks (e.g., organizing a party at home next Saturday).

At the same time, the integrated awareness information workspace offered by CONRAD provides users with a *coordinating representation* of shared activities which, as discussed in [Introne and Alterman, 2006], has a central role in preventing mistakes and supporting user coordination in any groupware environment.

5.2 CONTEXT

The informational definition of context presented by Dey and Abowd [2000] and the relational one described in [Dourish, 2004] have both been taken into account in our work. In fact, we recognize that information such as people, objects, location, identity, activity and time are fundamental for describing contexts, as they are all part of our concept of activity frame; moreover, an activity frame is an evolving element, which is defined and modelled by users on the basis of their relations and targets, and that in turns defines relations between people, objects and activities.

We exploit informational and relational aspects of context also when context information is to be presented to users in form of awareness information:

- In synchronous presentation modality, the presentation of awareness information is subject to policies which consider the relations between the originating context and the current focus of operativity of the user.

- In asynchronous presentation modality, awareness information is presented in a structured way, whose organization is derived by the relations between elements that have been described by users within activity frames and tasks.

It could be interesting to compare our architectural approach in defining a context aware system with the guidelines expressed in [Dey and Abowd, 2001].

- *Separation of concerns* is respected, as Group Manager or Collaborative Task Managers instances synchronize with each other on the basis of contextual information, while each CTM instance provide its user with a personal view of her/his workspaces.

- The architecture deals with *context interpretation* with a loosely-coupled interaction among software components based on the Publish and Subscribe protocol. Each software component is wrapped by an adapter, which subscribes the component for relevant types of events, translates events and messages to such a component specific format and invokes the APIs of the wrapped software component in order to perform the appropriate operations. User programs, therefore, do not directly deal with low level contextual information.

- Communication between components is *transparent*, as it is in charge of the Publish and Subscribe server, which acts as a hub among software components; the components that are interested in receiving a certain type of information subscribe to it, and when the hub receives new information, it notifies the subscribed components.
- The Publish and Subscribe server is *constantly available* to subscribed components; therefore, it can receive new information from components and notify them of its existence at any time.
- The loosely-coupled modality of interaction between components of our architecture is partially in contrast with the centered, *architectural storage* of context information: context information "flows" from the components to the hub, and then back to each subscribed component (more precisely, its adapter).
- *Resource discovery* is fully supported, thanks to the Publish and Subscribe server subscription mechanism.

Most groupware environments (e.g., Project Vie IM [Scupelli et al., 2005], Collanos [Collanos, 2008], Feng Office [Feng Office, 2010] and ActiveCollab [ActiveCollab, 2008]) are based on closed architectures; as such environments cannot be extended with external business services, they expose the users to a fragmented view of their contexts, considering that non-native application cannot be included in their scope.

Contextual information produced by external services is unlikely to be integrated within a common management structure, as it is with our approach. In fact, our framework overcomes such a limitation because it is based on an open architecture, supporting the integration of heterogeneous software components into a customized collaboration environment and holistically handling the contextual information that is produced by each integrated service.

Similar to our approach, Atmosphere [Rittenbruch, 2002] introduces the concept of *contextual awareness* to explicitly relate objects and awareness events to contexts. The notion of Sphere, which is a user-defined representations of a context, containing a set of jointly used artifacts, is closely related to our concept of activity frame. However, the authors introduce the notion of *contextor*, which is a description that models the intentions of the users operating within a context; users are therefore asked to explicitly select the contextors which motivate their own actions while they interact with business services. The user is therefore involved in both representing contexts and assigning events to them. In comparison, CONRAD does not impose any extra-efforts on the user's activities: it bases the representation of the intentions behind actions by relying on task management. We know from literature how task management has been recognized as an important feature for the organization of various types of activities (e.g., distributed collaborative writing, in [Tran et al., 2006]); task management is also applied as an optimal strategy to "get things done" in most project management tools, as well as in the recent task manager applications based on the GTD model [Allen, 2003]; e.g., DoIt [DoIt.im, 2011] and Things [Cultured Code, 2011].

In the context modeling method proposed by Gross and Prinz [2003], the attributes of the awareness events are matched against the context descriptions to identify the closest one and manage events accordingly: this behavior is similar to what we have implemented for the recognition of the Current Focus of attention of a user. Anyway, we can find some differencies in the way context information is handled by the system:

- CONRAD can merge the context information provided by integrated business services (e.g., contact lists, etc.) to determine to which existing activity context the events can be associated (see section 3.1.2).
- Following the guidelines in [Dey and Mankoff, 2005], CONRAD involves the user in the loop by accepting corrections when events are misclassified. This way, the system can correct the occurred mistakes.

5.3 AWARENESS

5.3.1 *Synchronous Awareness Management*

As for what regards notifications handling, currently we do not deal with breakpoints in tasks [Bailey et al., 2006, Iqbal and Bailey, 2007, 2008], as this aspect involves mapping event types to breakpoint evidence, which can very difficult for the administrator of an open collaboration environment and thus hardly applicable in realistic use cases.

The notification management policies offered by our framework belong to the mediated category of McFarlane [2002], with the Notification Manager playing the mediator role. CONRAD improves notification management by introducing a context-dependent dimension: the user is allowed to filter out notifications depending on her/his current activities and thus on the kind of information which (s)he might need at each time.

The NESSIE awareness management environment [Prinz, 1999] supports the specification of interest profiles used to filter the notifications generated by the applications. However, such profiles are based on the specification of event patterns, similar to the e-mail filters offered by current mail clients, and do not explicitly model the user's activity contexts.

In the *bounded deferral* approach proposed by Horvitz et al. [2005], the priority of the interruptions is computed on the basis of a set of features that have been specified by the user; e.g., if the sender of an e-mail is my boss, then the message has high priority; these features lack the contextuality that is central to our approach. Our work, indeed, treats the contextual information behind notifications as the main factor for determining its possible relevance for a user. A feature based selection could be a great improvement for our context based filtering, in order to grant finer grained notification policies for the users that want to accurately tuning their flow of notifications (e.g., if my boss is also a friend of mine, I might wish to specify that the notifications concerning his activities at work should be immediately submitted, while other notifications might be safely delayed).

The CASSIUS framework [Kantor and Redmiles, 2002] exploits a subscription based awareness support: this approach is similar to ours, as it implies event driven notifications and a subscrition mechanism from awareness sources. The presentation of awareness information is although very simple and a-contextual, as it is up to the users to browse and organize the various sources, in order to create ad hoc subscriptions. However, the framework is clearly addressed to more open ended, generic fields of use than ours, with the possibility to realize simple and on-the-fly subscriptions to a large number of diverse awareness sources.

In [Wang et al., 2007], the BSCW awareness support is extended by defining dynamic notification preference profiles and by tuning the *intensity* of notification streams accordingly. This functionality is complementary to the one offered by CONRAD, which focuses on informing the user about the events concerning the particular activities (s)he is carrying out.

Intelligent and adaptive systems [Harrer et al., 2006] can be seen as closely related to our approach of goal decomposition, which comprises the creation of tasks and subtask hierarchies. However, such systems remain inherently domain-specific and data-driven; our approach, instead, makes the user responsible for defining tasks and offers basic support to user coordination, while it does not require any specific domain knowledge (see section 3.1.1).

5.3.2 *Asynchronous Awareness Management*

Our approach in asynchronous awareness management is in line with the elements described in [Fuchs et al., 1995], as it supports the provision of information based on:

- Objects: work artifacts are included in the definition of activity frames, while groups and roles are supported by the group manager.
- Relations: structural relations between objects and context are supported by activity frames definitions.
- Events: users actions are structured on the basis of a hierarchical structure, as described by the Collaborative Task Manager. Moreover, the incremental access to awareness information differentiates our work from standard groupware and project management tools, which organize information on the sole basis of its reference workspace/group/directory.

Our incremental access model represents a valid answer to the problem of temporal fragmentation of asynchronous awareness [Pankoke-Babatz et al., 2004], as it mixes different characteristics from two different visualization modalities: text-based representations and tag clouds.

- The middle and lower layers of our model give a highly detailed description of past activity, ideal for short-term usage.

- The higher layer of our model (the Awareness Cloud) is aimed at giving as much information as possible at a single glance, exploiting a visual metaphor as a general overview for the awareness information.

As in our approach, AwarenessMaps [Gross et al., 2003], Info-Lotus [Zhang et al., 2005] and Scope [van Dantzich et al., 2002] are synthetic forms of awareness provision that exploit the degree of activity within a group or a shared workspace. Our proposal makes a step forward in this direction by visualizing awareness information at different granularity and abstraction levels:

- The *granularity* aspect [Collins et al., 1996] concerns the generality of the activity context to be considered and is motivated by the fact that users engage in different types of collaborations, such as thematic groups (e.g., small or large virtual communities), more or less structured projects, and specific tasks.

- The *abstraction* aspect enables the users to receive a summary of the state of their activity contexts, from which they can select the contexts to be inspected in detail.

The activity degree of a user has also been taken as an information measure for online collaborative communities [Vassileva and Sun, 2007, Baishya and Brusilovksy, 2009]. Our proposal differs from such works because, besides modeling individual users and groups, we model the user's activity contexts. Therefore, the visualization we propose enables the user to assess the state of general collaborations or to focus on peculiar aspects, such as particular tasks. This feature makes our visualization model suitable for integration in collaboration environments, where users can engage in shared activities having different complexity levels.

6 CONCLUSIONS

In this thesis, we have introduced the idea of a context-based awareness support service for open collaboration environments. Open environments rely on web applications, as they are offered by the Web 2.0; complex activities and projects might require the adoption of multiple applications for being carried out in such environments, with each one offering a separate workspace.

We proposed the CONRAD framework, which has the role to mediate between users and heterogeneous Web 2.0 services, providing a cross-application perspective on the information they generate:

- The framework models users activity contexts at different granularities: collaboration groups, activity frames and tasks.
- The context of users operations is specified exploiting the Collaborative Task Manager, which classifies events within specific contexts accordingly.
- The framework is able to recognize, at each instant of time, the particular context (or set of contexts) which the user is focusing on (the Current Focus of attention).

We support awareness information management with two complementary modalities: synchronous and asynchronous delivery. Synchronous awareness information management aims at maximizing the benefits from the trade-off between informing and interrupting users. It consists in real-time notifications about the events occurred in the users activity contexts; such notifications can be filtered according to different policies to meet individual user preferences:

- The Total filter policy, which blocks all the notifications from a specified activity context.
- The No filter policy, which allows all the notifications from a specified activity context to be submitted as soon as they are generated.
- The Context filter policy, which allows the notifications from the activity frames in the users current focus of attention to be submitted.
- The Task filter policy, which allows the notifications from the tasks in the users focus of attention to be submitted.

Users can select one policy as a default in order to apply it to all of the activity contexts. Notifications are handled as Instant Messages and presented in pop-up windows.

Asynchronous awareness information management is designed to facilitate the users in resuming the state of their activity contexts, enabling them to visualize awareness events from the perspective that better reflects their information needs. We proposed a three-layered awareness presentation model:

- The higher layer visualization exploits a tag cloud visualization in order to show the degree of activity occurred in the user's contexts; the aim of this visualization is to give the user a general overview of his collaborations.
- The middle layer visualization enables the user to view the detailed awareness information from the perspective of a specific context.

- The lower layer visualization presents the long term history of all the user's activity contexts.

The main contributions of this thesis can be summarized as follows:

- The open architecture, based on subscription model, that allows integration of streams from various collaboration and notification services.
- The integrated synchronous and asynchronous awareness approach, composed of both notifications and context awareness.
- The hierarchical policies for filtering notifications based on the user's current activity.
- The tag-cloud based hierarchical, asynchronous awareness visualization tool.

These findings could be applied in computer supported collaboration settings, which use web 2.0 services, as long as these services have the necessary characteristics for being integrated into the framework. A major limitations of the approach is represented by the centralized architecture of the framework, which requires an administrator to be set up. The use of the Collaborative Task Manager, which is used to practically represent context and activity knowledge, is also mandatory.

Our results also gave us interesting insights for future research. One important field that is worth investigating is the identification of *alternative presentation modalities* for displaying the delivered notifications. This feature is particularly relevant considering that users showed some preoccupation, during the experiment, as they could not immediately detect the reference activity context of the notifications they were receiving, and thus, they could not quickly evaluate their importance at first glance. This aspect could indeed be connected with the fact that all the notifications were presented with the same modality (pop-up windows, see Figure 3), regardless of the fact that they were relevant to the user's current activities or not. Therefore, the user had only one solution, that is, to read them thoroughly in order to identify their context and their eventual relevancy with his current actions.

As reported in chapter 4, this lead some users to immediately minimize all pop-up windows as they appeared, without reading them, or after reading just a couple of notifications that happened to be not relevant to their current task.

We plan to overcome this issues, improving the presentation modalities in order to support an easy identification of the context of the incoming notifications. For instance, notifications that are unrelated to the user's current focus might be delivered with a less intrusive modality (for example, as automatically fading balloons). In contrast, the Instant Message pop-up might be reserved for notifications concerning the user's current tasks and thus deserving immediate attention and a direct point of access to manage them.

The artificial setting of the experiment, which led the users to operate in an unfamiliar context, can possibly have led them to treat every notification as noise, considering that, apart from a few, they did not have an effect on a real-life situation. This could make users highly positive to any filter that reduces the number of notifications; at this stage, our results confute such an hypothesis, as the stricter Task filter did not have a different impact from the broader Context filter.

On the contrary, highest priority notifications, that can seriously impact on the user's situation ("The power line is shutting down now!"), cannot be easily discarded within a filtering policy; anyway, the artificial setting made once again very hard to correctly handle such notifications, that were therefore discarded by the experimenter: a thorough field test will be necessary to further explore such themes.

The second experiment revealed that the incremental access to awareness information significantly improved users' performance, in terms of times of execution and number of errors. Users also expressed satisfaction regarding the simplicity and pleasantness of use of our incremental model, with the higher layer that received the most favorable opinions. Nevertheless, it is worth noting that users indicated as a major drawback of the Awareness Cloud the fact that it made hard to spot nodes with a very low density of events: while it could be much faster to identify high density elements (specifically, collaboration groups and contexts with high levels of activity), those written with the smallest font (such as low-activity tasks) might get lost among the crowd.

This aspect is typical of a tag cloud [Bateman et al., 2008] but could be addressed by supporting a personalized configuration of the cloud, based on the user's interests. We plan to enable the user to configure the Awareness Cloud by specifying which elements (s)he wants to monitor with most attention. When the cloud is generated, such elements will then be displayed with a different color (e.g., red instead of traditional light blue) and would never be omitted whenever the cloud is too large.

Another approach that is worth exploring for personalizing the configuration of the cloud is adaptivity. One possible improvement in this area could be represented by an observation of the past activity of a user in order to extract helpful information for parameterizing the elements visualized in the cloud. Our context-aware architecture offers some useful sources of information about users behavior, such as the Current Focus log and the Event log; an example of a possible adaptation rule could be that contexts in which the user has been involved the most (in terms of produced events) are presented in the cloud with a distinctive and progressive visual element (e.g., a lighter to darker color nuance).

With this regard, Kleanthous and Dimitrova [2010] propose a method for providing personalized support to virtual communities, depending on what is known about knowledge sharing activities in such communities; they use psychological models for automatically detecting problematic areas, enabling targeted notifications for different community members, with the aim of improving the functioning of the community as a whole. Martinez et al. [2011] improve observation of group activity by automatically distinguishing between collaborative, non-collaborative or somewhat collaborative activities within the group, by exploiting log traces along with video and audio recordings.

Other improvements to the Awareness Cloud might originate from the introduction of a multi-faceted search capability [Tunkelang, 2009]. Contextual features (collaboration group, activity frame and task), alongside with more traditional elements (author, date, etc.) could be used as a refinement for obtaining a more precise, ad-hoc visualization of the Awareness Cloud.

Faceted search could also be a solution to an issue of the actual tag cloud that we might call "duplication": very active authors appear as very large, alongside with the contexts in which they acted; therefore, a single very active author could severely alter the cloud's shape. This issue could be solved with the implementation of a visualization modality that calculates contexts' dimension by considering the number of authors that contributed to each context, instead of the number of single actions.

A

APPENDIX

A.1 Experiment I: NASA-TLX Questionnaire

Indicate the value that better defines your perception of the below indicated elements, relatively to the experiment you have completed. Use values between 0 and 10, with a 0,5 scale (example: 7,5).

- Mental Demand:
- Temporal demand:
- Performance:
- Effort:
- Frustration:

For each of the following points, cross the element that has contributed the most to the workload during the experiment.

1. Performance vs. Temporal demand
2. Mental demand vs. Effort
3. Frustration vs. Mental demand
4. Temporal demand vs. Frustration
5. Frustration vs. Effort

6. Performance vs Mental demand
7. Performance vs. Frustration
8. Effort vs. Performance
9. Temporal demand vs. Mental demand
10. Temporal demand vs. Effort

A.2 EXPERIMENT II: POST-TEST QUESTIONNAIRE

Express your evaluation about the tools you have utilized during the experiment. Use values between 1 and 7, with a 1 scale (example: 5).

- Efficacy (how well the tool has satisfied your necessities):
- Efficiency (how rapidly, and with the minimum waste of resources, you reached your goals using the tool):
- Simplicity of use:
- Pleasantness of use:

A.3 TRANSLATION OF ITALIAN TERMS IN FIGURES

Figures contain the following italian terms, which are translated for better comprehension:

Prenotare albergo = Book hotel.
Raccolta articoli bibliografia = Retrieve bibliography papers.
Scrittura documento presentazione = Write presentation document.
Lavoro B = Work B.
Articolo conferenza europea = European conference paper.

Mandare inviti = Send invitations.

Scrittura business plan = Write business plan.

Organizzazione vacanza = Plan holidays.

Scrittura report = Write report.

Organizzazione cena = Organize dinner.

Prenotare biglietti treno = Book train tickets.

Prenotare biglietti aereo = Book airplaine tickets.

Privato = Private.

Progetto europeo = European project.

Conferenza oltreoceano = Overseas conference.

Scrittura abstract = Write abstract.

Ottenere visti = Get visas.

Lavoro A = Work A.

BIBLIOGRAPHY

ActiveCollab. Project management & collaboration. http://www.activecollab.com/, 2008.

P.D. Adamczyk and B.P. Bailey. If not now, when?: The effects of interruption at different moments within task execution. In *Proceedings of the SIGCHI conference on Human factors in computing systems (CHI '04)*, pages 271–278. ACM, 2004.

D. Allen. *Getting Things Done: the art of stress-free productivity*. Penguin, 2003.

C. Amelung, J. Laffey, and P. Turner. Supporting collaborative learning in online higher education through activity awareness. In *Proc. 8th Int. Conf. on Computer Supported Collaborative Learning (CSCL '07)*, pages 75–77, New Brunswick, New Jersey, 2007.

W. Appelt. What groupware functionality do users really use? analysis of the usage of the bscw system. *Parallel, Distributed, and Network-Based Processing, Euromicro Conference on*, 0:337, 2001. ISSN 1066-6192.

L. Ardissono and G. Bosio. Context-dependent awareness support in open collaboration environments. *User Modeling and User-Adapted Interaction*, pages 1–32, 2011b. ISSN 0924-1868.

L. Ardissono, A. Goy, G. Petrone, M. Segnan, and P. Torasso. INTRIGUE: personalized recommendation of tourist attractions for desktop and handset devices. *Applied Artificial Intelligence, Special Issue on Artificial Intelligence for Cultural Heritage and Digital Libraries*, 17(8-9):687–714, 2003.

L. Ardissono, A. Goy, G. Petrone, and M. Segnan. From service clouds to user-centric personal clouds. In *Proceedings of IEEE 2009 International Conference on Cloud Computing (CLOUD-II 2009)*, pages 1–8, Bangalore, India, 2009. IEEE.

L. Ardissono, G. Bosio, A. Goy, G. Petrone, and M. Segnan. Open, collaborative task management in Web 2.0. In *Proceedings of MCCIS 2010 IADIS multiconference on computer science and information systems*, pages 20–27, Freibug, Germany, 2010a. IADIS Press.

L. Ardissono, G. Bosio, A. Goy, G. Petrone, M. Segnan, and F. Torretta. Collaborative service clouds. *International Journal of Information Technology and Web Engineering*, 5(4):23–39, 2010b.

L. Ardissono, G. Bosio, and M. Segnan. An activity awareness visualization approach supporting context resumption in collaboration environments. In *Proceedings of the International Workshop on Adaptive Support for Team Collaboration 2011 (ASTC 2011) - CEUR Workshop Proceedings, Vol. 743*, pages 15–25, Girona, Spain, 2011a. CEUR. URL http://sunsite.informatik.rwth-aachen.de/Publications/CEUR-WS/Vol-743/.

L. Ardissono, G. Bosio, and M. Segnan. A visualization model supporting an efficient context resumption in collaboration environments. *Advances in User Modeling: selected papers from UMAP 2011 workshops.*, Lecture Notes in Computer Science(7138):(to appear), 2011b.

B.P. Bailey and S.T. Iqbal. Understanding changes in mental workload during execution of goal-directed tasks and its application for interruption management. *ACM transactions on Computer-Human Interaction*, 14(4):1–28, 2008.

B.P. Bailey, J.A. Konstan, and J.V. Carlis. Measuring the effects of interruptions on task performance in the user interface. In *Proceedings of the IEEE*

International Conference on Systems, Man, and Cybernetics, pages 757–762, Nashville, TN, USA, 2000. IEEE Press.

B.P. Bailey, J.A. Konstan, and J.V. Carlis. The effects of interruptions on task performance, annoyance, and anxiety in the user interface. In *Proceedings of Human-Computer Interaction - INTERACT'01*, pages 593–601, Tokyo, 2001. IOS Press.

B.P. Bailey, P.D. Adamczyk, T.Y. Chang, and N.A. Chilson. A framework for specifying and monitoring user tasks. *Computers in Human Behavior*, 22(4): 709–732, 2006.

D. Baishya and P. Brusilovksy. CiteAware: visual group awareness for a reference sharing system. In *Proc. of workshop on visual interfaces to the social and semantic web (VISSW2009)*, pages 153–156, Sanibel Island, Florida, 2009.

M. Baldauf, S. Dustdar, and F. Rosenberg. A survey on context-aware systems. *International Journal of ad hoc and ubiquitous computing*, 2(4):263–277, 2007.

L. Bartram, C. Ware, and T. Calvert. Moticons: detection, distraction and task. *International Journal of Human-Computer Studies*, 58(5):515–545, 2003.

S. Bateman, C. Gutwin, and M. Nacenta. Seeing things in the clouds: the effect of visual features on tag cloud selections. In *Proc. of 19th ACM Conference on Hypertext and Hypermedia (HT'08)*, pages 193–202, Pittsburgh, PA, USA, 2008.

M. Bazire and P. Brézillon. Understanding context before using it. In A. Dey, B. Kokinov, D. Leake, and R. Turner, editors, *LNCS 3554, Modeling and Using Context*, pages 29–40, Berlin Heidelberg, 2005. Springer-Verlag.

G. Bosio. A user perspective on cloud computing. In *Proceedings of The 3rd International Conference on Advances in Human-oriented and Personalized*

Mechanisms, Technologies, and Services (CENTRIC 2010), pages 1–4, Nice, France, 2010. IEEE.

A.J.B. Brush, D. Bargeron, J. Grudin, and A. Gupta. Notification for shared annotation of digital documents. In *Proc. SIGCHI conference on Human factors in computing systems (CHI '02)*, pages 89–96, Minneapolis, USA, 2002.

J. Burrell and G. K. Gay. E-graffiti: Evaluating real-world use of a context-aware system. *Interacting with Computers*, 14(4):301–312, 2002.

J.M. Carroll, D.C. Neale, P.L. Isenhour, M.B. Rosson, and D. Scott McCrickard. Notification and awareness: synchronizing task-oriented collaborative activity. *International Journal of Human-Computer Studies*, 58(5):605–632, 2003.

J.M. Cellier and H. Eyrolle. Interference between switched tasks. *Ergonomics*, 35(1):25–36, 1992.

M. Chalmers. A historical view of context. *Computer Supported Cooperative Work: the Journal of Collaborative Computing*, 13:223–247, 2004.

K. Cheverst, N. Davies, K. Mitchell, and P. Smith. Providing tailored (context-aware) information to city visitors. In P. Brusilovsky, O. Stock, and C. Strapparava, editors, *LNCS n. 1892: Adaptive Hypermedia and Adaptive Web-Based Systems, International Conference (AH 2000)*, pages 73–85. Springer, Berlin Heidelberg New York, 2000.

S. Cohen. Aftereffects of stress on human performance and social behavior: a review of research and theory. *Psychological Bulletin*, 88(1):82–108, 1980.

Collanos. Products overview: Team enabling professionals. http://www.collanos.com/en/products, 2008.

Jason Collins, Jim Greer, and Sherman Huang. Adaptive assessment using granularity hierarchies and bayesian nets. In Claude Frasson, Gilles Gauthier, and Alan Lesgold, editors, *Intelligent Tutoring Systems*, volume 1086 of *Lecture Notes in Computer Science*, pages 569–577. Springer Berlin / Heidelberg, 1996. ISBN 978-3-540-61327-5.

Cultured Code. Things Mac. http://culturedcode.com/things/, 2011.

E.B.. Cutrell, M. Czerwinski, and E. Horvitz. Effects of instant messaging interruptions on computing tasks. In *Proceedings of CHI '00 extended abstracts on Human factors in computing systems*, pages 99–100, The Hague, Netherlands, 2000. ACM.

M. Czerwinski, S. Christman, and M. Rudisill. Interruptions in multitasking situations: The effects of similarity and warning. Technical Report JSC-24757, NASA Johnson Space Center, Houston, Texas, 1991a.

M. Czerwinski, S. Christman, and B. Schumacher. The effects of warnings and display similarity on interruption in multitasking environments. *ACM SIGCHI Bulletin*, 23(4):38–39, 1991b.

M. Czerwinski, E. Horvitz, and S. Wilhite. A diary study of task switching and interruptions. In *Proceedings of CHI 2004 Conference on Human Factors in Computing Systems*, pages 175–182, Vienna, 2004. ACM Press.

A. K. Dey and G. D. Abowd. The context toolkit: Aiding the development of context-aware applications. pages 434–441. ACM Press, 1999.

A.K. Dey and D. Abowd. 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, 2000.

A.K. Dey and D. Abowd. A conceptual framework and a toolkit for supporting the rapid prototyping of context-aware applications. *Human-Computer Interaction: special issue on Context-Aware computing*, 16(2):97–166, 2001.

A.K. Dey and J. Mankoff. Designing mediation for context-aware applications. *ACM transactions on Computer-Human Interaction*, 12(1):53–80, 2005.

Y. Ding, R. Malaka, C. Kray, and M. Schillo. RAJA - a resource-adaptive Java agent infrastructure. In *Proceedings of 5th International Conference on Autonomous Agents (Agents '01)*, pages 332–339, Montreal, CA, 2001.

DoIt.im. Doit anywhere, any time! http://www.doit.im/, 2011.

P. Dourish. What we talk about when we talk about context. *Personal and Ubiquitous Computing*, 8(1):19–30, 2004.

P. Dourish and V. Bellotti. Awareness and coordination in shared workspaces. In *Proceedings of the 1992 ACM conference on Computer-Supported Cooperative Work (CSCW '92)*, pages 107–114, Toronto, Canada, 1992. ACM.

J.S. Erickson, S. Spence, M. Rhodes, D. Banks, J. Rutherford, E. Simpson, G. Belrose, and R. Perry. Content-centered collaboration spaces in the Cloud. *IEEE Internet Computing*, 13(5):34–42, 2009.

Facebook. Facebook helps you connect and share with the people in your life. http://www.facebook.com/, 2011.

P. Fahy and S. Clarke. *CASS-middleware for mobile context-aware applications*, pages 304–308. Ieee, 2004.

Feng Office. Feng Office: Your world wide office. http://www.fengoffice.com, 2010.

L. Fuchs. Area: A cross-application notification service for groupware. In *In Proceedings of the Sixth European Conference on Computer-supported Cooperative Work*, pages 61–80. Kluwer Academic Publishers, 1999.

Ludwin Fuchs, Uta Pankoke-Babatz, and Wolfgang Prinz. Supporting cooperative awareness with local event mechanisms: the groupdesk system. In *Proceedings of the fourth conference on European Conference on Computer-Supported Cooperative Work*, ECSCW'95, pages 247–262, Norwell, MA, USA, 1995. Kluwer Academic Publishers. ISBN 0-7923-3697-6. URL http://dl.acm.org/citation.cfm?id=1241958.1241974.

C. H. Ganoe, J. P. Somervell, D. C. Neale, P. L. Isenhour, J. M. Carroll, M. B. Rosson, and D. S. Mccrickard. Classroom bridge: using collaborative public and desktop timelines to support activity awareness. In *In Proceedings of Sixteenth Annual ACM Symposium on User Interface Software and Technology*, pages 21–30, 2003.

GigaSpaces. GigaSpaces SBA. http://www.gigaspaces.com/pr_overview.html, 2008.

Google. Google authentication. http://code.google.com/intl/it-IT/apis/accounts/docs/AuthForWebApps.html, 2010a.

Google. Google documents. http://www.google.com/google-d-s/tour1.html, 2010b.

Google. Google Web Toolkit. http://code.google.com/intl/it-IT/webtoolkit/, 2010c.

A. Grimes and A.J. Brush. Life scheduling to support multiple social roles. In *Proceedings of CHI 2008*, pages 821–824, Florence, Italy, 2008. ACM.

T. Gross and W. Prinz. Awareness in context: A light-weight approach. In *Proc. of the Eighth European Conference on Computer-Supported Cooperative*

Work (ECSCW 2003), Kluwer Academic Publishers, Dortrecht, NL (2003, pages 295–314. Kluwer Academic Publishers, 2003.

T. Gross, W. Wirsam, and W. Gräther. AwarenessMaps: Visualising awareness in shared workspaces. In *Extended Abstracts of the Conference on Human Factors in Computing Systems - CHI 2003*, pages 784–785. ACM, New York, 2003.

Tao Gu, Hung Keng Pung, and Da Qing Zhang. A middleware for building context-aware mobile services. *2004 IEEE 59th Veicular Tecnology Conference VTC 2004Spring IEEE Cat No04CH37514*, 5(1):2656–2660, 2004.

C. Gutwin. *Workspace Awareness in Real-Time Distributed Groupware*. PhD thesis, Department of Computer Science, University of Calgary, 1997.

C. Gutwin and S. Greenberg. The effects of workspace awareness support on the usability of real-time distributed groupware. *ACM Transactions on Computer-Human Interaction, Special Issue on HCI in the new Millennium*, 6 (7):243–281, 1999.

C. Gutwin, S. Greenberg, and M. Roseman. Workspace awareness in real-time distributed groupware: Framework, widgets, and evaluation. In *Proceedings of HCI on People and Computers XI*, pages 281–298, London, 1996.

J.M. Haake, T. Hussein, B. Joop, S. Lukosch, D. Veiel, and J. Ziegler. Modeling and exploiting context for adaptive collaboration. *International Journal of Cooperative Information Systems*, 19(1-2):71–120, 2010.

M. J. Halvey and M. T. Keane. An assessment of tag presentation techniques. In *WWW '07: Proceedings of the 16th international conference on World Wide Web*, pages 1313–1314, New York, NY, USA, 2007. ACM Press. ISBN 9781595936547.

A. Harrer, B. McLaren, E. Walker, L. Bollen, and J. Sewall. Creating cognitive tutors for collaborative learning: steps toward realization. *User Modeling and User-Adapted Interaction*, 16(3):175–209, 2006. 10.1007/s11257-006-9007-4.

S.G. Hart and L.E. Stateland. Development of NASA-TLX (Task Load Index): results of empirical and theoretical research. In P.A. Hancock and N. Meshkati, editors, *Human Mental Workload*, pages 239–250. North-Holland, Amsterdam, 1988.

K. Hayashi, T. Hazama, T. Nomura, T. Yamada, and S. Gudmundson. Activity awareness: a framework for sharing knowledge of people, projects, and places. In *Proceedings of the sixth conference on European Conference on Computer Supported Cooperative Work*, ECSCW'99, pages 99–118, Norwell, MA, USA, 1999. Kluwer Academic Publishers. ISBN 0-7923-5948-X.

M. A. Hearst and D. Rosner. Tag clouds: Data analysis tool or social signaller? *Hawaii International Conference on System Sciences*, 2008. ISSN 1530-1605.

T. Horstmann and R. Bentley. Distributed authoring on the Web with the BSCW shared workspace system. *StandardView*, 5(1):9–16, 1997.

E. Horvitz. Principles of mixed-initiative user interfaces. In *Proc. of ACM Conference on Human Factors in Computing Systems (CHI'99)*, pages 159–166, Pittsburgh, PA, 1999.

E. Horvitz, J. Apacible, and M. Subramani. Balancing awareness and interruption: Investigation of notification deferral policies. In *Lecture Notes in Artificial Intelligence n. 3538:Proceedings of 10th International Conference on User Modeling*, pages 433–437, Edinburgh, UK, 2005. Springer-Verlag.

J. Introne and R. Alterman. Using shared representations to improve coordination and intent inference. *User Modeling and User-Adapted Interaction*, 3-4 (4):249–280, 2006.

S. T. Iqbal and B. P. Bailey. Investigating the effectiveness of mental workload as a predictor of opportune moments for interruption. In *CHI '05 extended abstracts on Human factors in computing systems*, CHI EA '05, pages 1489–1492, New York, NY, USA, 2005. ACM. ISBN 1-59593-002-7.

S.T. Iqbal and B.P. Bailey. Understanding and developing models for detecting and differentiating breakpoints during interactive tasks. In *Proceedings of SIGCHI conference on Human factors in computing systems*, pages 697–706, San Jose, California, 2007. ACM.

S.T. Iqbal and B.P. Bailey. Effects of intelligent notification management on users and their tasks. In *Proceedings of 26th annual SIGCHI conference on Human factors in computing systems*, pages 93–102, Firenze, Italy, 2008.

S.T. Iqbal and E. Horvitz. Leveraging characteristics of task structure to predict the cost of interruption. In *Proc. of Conference on Human Factors in Computing Systems*, pages 741–750, Montreal, Canada, 2006. ACM.

S.T. Iqbal and E. Horvitz. Disruption and recovery of computing tasks: field study, analysis, and directions. In *Proc. of Computer/Human Interaction 2007 (CHI 2007)*, pages 677–686, San Jose, CA, 2007.

S.T. Iqbal and E. Horvitz. Notifications and awareness: a field study of alert usage and preferences. In *Proceedings of the 2010 ACM conference on Computer Supported Cooperative Work (CSCW 2010)*, pages 27–30, Savannah. Georgia, 2010. ACM.

Jalava. Online diagram editor. draw flowcharts, shapes, process diagrams, and more on the web., 2010.

JBoss Community. jBPM makes your workflow. http://www.jboss.org/jbpm, 2010.

M. Kantor and D. Redmiles. CASSIUS: designing dynamic subscription and awareness services. In *Proceedings of CSCW Workshop on ad hoc communications and collaboration in ubiquitous computing environments*, New Orleans, LA, 2002.

C. Kerer, S. Dustdar, M. Jazayeri, D. Gomes, A. Szego, and J. A. Burgos Caja. Presence-aware infrastructure using web services and rfid technologies. In *Proceedings of the 2nd European Workshop on Object Orientation and Web Services*, 2004.

Styliani Kleanthous and Vania Dimitrova. Analyzing community knowledge sharing behavior. In Paul De Bra, Alfred Kobsa, and David Chin, editors, *User Modeling, Adaptation, and Personalization*, volume 6075 of *Lecture Notes in Computer Science*, pages 231–242. Springer Berlin / Heidelberg, 2010. ISBN 978-3-642-13469-2.

K. Latorella. Investigating interruptions: Implications for flightdeck performance. Technical Report TM-1999-209707, NASA, 1999.

G. Mark and N. Su. Considering Web 2.0 technologies within an ecology of collaborations. In *Proc. of SociUM: Adaptation and Personalisation in Social Systems: Groups, Teams, Communities*, pages 50–59, Corfu, Greece, 2007.

G. Mark, D. Gudith, and U. Klocke. The cost of interrupted work: more speed and stress. In *Proceedings of CHI 2008*, pages 107–110, Firenze, Italy, 2008. ACM.

Roberto Martinez, James R. Wallace, Judy Kay, and Kalina Yacef. Modelling and identifying collaborative situations in a collocated multi-display groupware setting. In *Proceedings of the 15th international conference on Artificial* *intelligence in education*, AIED'11, pages 196–204, Berlin, Heidelberg, 2011. Springer-Verlag. ISBN 978-3-642-21868-2.

D. S. McCrickard and C. M. Chewar. Attuning notification design to user goals and attention costs. *Communications of the ACM*, 46:67–72, 2003.

D. S. McCrickard, M. Czerwinski, and L. Bartram. Introduction: Design and evaluation of notification user interfaces. *International Journal of Human-Computer Studies*, 8:509–514, 2003.

D.C. McFarlane. Comparison of four primary methods for coordinating the interruption of people in Human-Computer Interaction. *Human-Computer Interaction*, 17(1):63–139, 2002.

D.C. McFarlane and K.A. Latorella. The scope and importance of human interruption in human-computer interaction design. *Human-Computer Interaction*, 17(1):1–61, 2002.

M. A. Muñoz, V. M. Gonzalez, M. Rodríguez, and J. Favela. Supporting Context-Aware Collaboration in a Hospital: An Ethnographic Informed Design. 2003.

OrbiTeam Software GmbH & Co. KG. Be smart - cooperative, worldwide. http://public.bscw.de/en/about.html, 2011.

U. Pankoke-Babatz, W. Prinz, and L. Schäfer. Stories about asynchronous awareness. In *COOP'04*, pages 23–38, 2004.

W. Prinz. NESSIE: an awareness environment for cooperative settings. In *Proc. of 6th European Conference on Computer Supported Cooperative Work (ECSCW '99)*, pages 391–410, Pittsburgh, PA, 1999. Springer.

W. Prinz, H. Löh, M. Pallot, H. Schaffers, A. Skarmeta, and S. Decker. ECOSPACE - towards an integrated collaboration space for eProfessionals.

In *Proceedings of 2nd International Conference on Collaborative Computing: networking, applications and worksharing*, pages 39–45, Atlanta, Georgia, 2006. IEEE.

M. Rittenbruch. Atmosphere: a framework for contextual awareness. *International Journal of Human-Computer Interaction*, 14(2):159–180, 2002.

A. W. Rivadeneira, D. M. Gruen, M. J. Muller, and D. R. Millen. Getting our head in the clouds: toward evaluation studies of tagclouds. In *CHI '07: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 995–998, New York, NY, USA, 2007. ACM. ISBN 978-1-59593-593-9.

C. Roda and J. Thomas. Attention aware systems: Theories, applications, and research agenda. *Computers in Human Behavior*, 22(4):557–587, 2006.

P. Scupelli, S. Kiesler, S.R. Fussell, and C. Chen. Project View IM: a tool for juggling multiple projects and teams. In *Proceedings of CHI'2005: Human factors in computing systems*, pages 1773–1776, Portland, Oregon, 2005. ACM.

Carla Simone, Gloria Mark, and Dario Giubbilei. Interoperability as a means of articulation work. In W. Prinz and A. Wolf, editors, *Proceedings of ACM Conference on Work Activities Coordination and Collaboration*, 1999.

J. Sinclair and M. Cardew-Hall. The folksonomy tag cloud: when is it useful? *Journal of Information Science*, 34(1):15–29, 2008.

M. Sohlenkamp, W. Prinz, and L. Fuchs. Poliawac: Design and evaluation of an awareness-enhanced groupware client. *AI Soc.*, 14:31–47, April 2000. ISSN 0951-5666.

J. Tam and S. Greenberg. A framework for asynchronous change awareness in collaborative documents and workspaces. *International Journal of Human Computer Studies, (In*, 64:583–598, 2006.

TeamWox. TeamWox GroupWare. http://www.teamwox.com, 2011.

M.H. Tran, G.K. Raikundalia, and Y. Yang. Using an experimental study to develop group awareness support for real-time distributed collaborative writing. *Information and Software Technology*, 48(11):1006–1024, 2006.

D. Tunkelang. *Faceted Search*. Synthesis lectures on information concepts, retrieval, and services. Morgan and Claypool Publishers, 2009. ISBN 9781598299991.

Twitter. Twitter is without a doubt the best way to share and discover what is happening right now. http://twitter.com/twitter, 2011.

M. van Dantzich, D. Robbins, E. Horvitz, and M. Czerwinski. Scope: providing awareness of multiple notifications at a glance. In *Proceedings of the Working Conference on Advanced Visual Interfaces*, New York, 2002.

J. Vassileva and L. Sun. Using community visualization to stimulate participation in online communities. *e-Service Journal*, 6(1):3–39, 2007.

M. Vonrueden and W. Prinz. Distributed document contexts in cooperation systems. In *CONTEXT*, pages 507–516, 2007.

V.S. Pendyala and S.S.Y. Shim. The Web as the ubiquitous computer. *Computing now*, (September):90–92, 2009.

Y. Wang, W. Gräther, and W. Prinz. Suitable notification intensity: the dynamic awareness system. In *GROUP'07: Proceedings of the 2007 International ACM Conference on supporting group work*, pages 99–106, New York, NY, 2007. ACM.

Wikipedia. Publish/subscribe. http://en.wikipedia.org/wiki/Publish/subscribe, 2010.

L. Zhang, N.T., and D. Vronay. Info-Lotus: a peripheral visualization for email notification. In *Proc. of CHI 2005*, pages 1901–1904, Portland, Oregon, 2005.

A. Zimmermann and A. Lorenz. LISTEN: a user-adaptive audio-augmented museum guide. *User Modeling and User-Adapted Interaction*, 18(5):389–416, 2008.

A. Zimmermann, A. Lorenz, and R. Oppermann. An operational definition of context. In *CONTEXT 2007, Lecture Notes in Artificial Intelligence n. 4635*, pages 558–6571. Springer-Verlag, Berlin Heidelberg, 2007.