ORIGINAL PAPER

Context-dependent awareness support in open collaboration environments

Liliana Ardissono · Gianni Bosio

Abstract   The widespread adoption of online services for performing work, home
and leisure tasks enables users to operate in the ubiquitous environment provided
by the Internet by managing a possibly high number of parallel (private and shared)
activity contexts. 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 emo-
tional and attentional states. 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. These policies are offered by the COntext depeNdent
awaReness informAtion Delivery (CONRAD) framework. We tested such policies
with users by applying them in a collaboration environment that includes a set of
largely used Web 2.0 services. The experiments show that our policies reduce the
levels of workload on users while supporting an up-to-the-moment understanding of
the interaction with their shared contexts. The present paper presents the CONRAD
framework and the techniques underlying the proposed notification policies.

Keywords   Personalized awareness information support · Notification
management policies · Interruption management · Collaboration environments ·
Context awareness · Web 2.0

L. Ardissono (✉) · G. Bosio
Dipartimento di Informatica, Università di Torino, Turin, Italy
e-mail: liliana.ardissono@di.unito.it

G. Bosio
e-mail: gianni.bosio@di.unito.it

1 Introduction

With the large availability of wireless connectivity, broadband internet connections and mobile devices, we witness a growing adoption of online services that enable private and corporate users to carry out activities and manage collaborations by exploiting the ubiquitous environment offered by the Internet (Pendyala and Shim 2009):

– People are increasingly using Web 2.0 applications to handle their life schedules, keep in touch with each other and share resources. For instance, consider the success of e-mail, Web calendars and of services such as Twitter (2011) and Facebook (2011) in supporting the interaction among distributed family members, friends and larger communities.
– The Enterprise 2.0 is envisioned, which employs Web applications for resource management to support mobile users in the execution of activities and to facilitate the cooperation in co-located and distributed teams (TeamWox 2011). Moreover, 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. For instance, some services support social awareness (Dix 1997) by displaying presence information about the user's contacts. Other services support workspace awareness (Gutwin et al. 1996) by reporting events about the operations performed on their User Interfaces (e.g., object manipulation). However, the execution of complex tasks typically requires multiple applications offering separate awareness support services. Thus, the user receives a fragmented picture of the state of her/his activities, presented in different workspaces and delivered as parallel notification flows which reflect application-dependent viewpoints on what has happened (Erickson et al. 2009; Ardissono et al. 2009c). In such a situation, the user is not allowed to holistically manage the awareness information and has to explicitly reconstruct each workspace starting from its fragmented presentation.

We aim at developing a holistic awareness support service for open collaboration environments, which can be extended by integrating external applications. As a first step toward such a goal, we present the COntext depeNdent awaReness informAtion Delivery (CONRAD) framework. CONRAD acts as a mediator between the user and business services providing a cross-application perspective on the information they generate. The framework offers a User Interface for the management of awareness information, which is presented (i) as synchronous notifications and (ii) in an asynchronous Web-based awareness space structured on the basis of the user's collaboration contexts.

As discussed in several works (e.g., Horvitz et al. 2005; Bailey et al. 2001), there is a trade-off between informing the user about what is happening and interrupting her/him with notifications about irrelevant events or with information that could be conveniently conveyed later on. Thus, information streams cannot simply be merged: they have to be managed in order to avoid overloading or disturbing the user during

the execution of activities. Our framework attempts to address the trade-off between informing and interrupting by applying filters which reduce the number of delivered notifications. The framework offers two context-dependent policies supporting the selection of notifications on the basis of the activities carried out by the user when the awareness events are generated:

1. The *context filter* informs the user about the events concerning the collaboration contexts (s)he is working at and filters out the other ones.
2. The *task filter* is more selective and filters notifications on the basis of the user's current task.

This paper describes the awareness information support offered by CONRAD focusing on its notification policies. We were interested in evaluating whether, in an open collaboration environment, the adoption of context-dependent notification policies such as the context and task filters reduces the detrimental effects of interruption, while giving users useful information for an up-to-the-moment understanding of the interaction with their shared contexts. For this purpose, we tested such policies with users by applying them in a collaboration environment that integrates a set of largely-used Web 2.0 services supporting the execution of shared activities. The experiments show that our context-dependent policies improve the levels of workload on users and provide useful information for catching up with the events occurred in their collaboration contexts.

The rest of this paper is organized as follows: Section 2 presents a usage scenario and a background on awareness information, interruption management and context modeling. Section 3 describes the awareness support offered by CONRAD, focusing on notification management. Section 4 describes the framework architecture and the techniques applied to manage the awareness information. Section 5 describes our evaluation of the context filter and task filter policies. Section 6 positions our contribution in the related work and Sect. 7 closes the paper.

This work extends the preliminary work presented in Ardissono et al. (2009a,b), which describe the context filter notification policy and a preliminary version of the algorithm for tracking the user's focus of attention across collaboration contexts.

2 Background

2.1 Scenario

The provision of awareness information is very important at work, where employees engage in several parallel collaborations. In fact, they need to be kept updated about the occurring events; moreover, they have to resume the state of such collaborations when switching among them (Mark and Su 2007; Czerwinski et al. 2004; Iqbal and Horvitz 2010; Haake et al. 2010). However, as people are starting to use online applications for their personal activities, awareness support is becoming useful everyday. For instance, Grimes and Brush (2008) reports 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.

The online management of activities has obvious advantages, such as the ubiquitous access to resources and the possibility of interacting with co-workers, friends, etc., by means of synchronous and asynchronous communication tools such as Instant Messaging and e-mail. However, at the current stage, it exposes users to a fragmented presentation of awareness information, which has to be explicitly reconstructed in order to understand what is happening. As a practical example, let's consider Mary, a woman with two children who works in a place out of town and participates in a car-sharing initiative with her colleagues. At work, Mary is part of different projects involving distributed teams of people who keep in touch with each other to carry out the assigned tasks. Moreover, in her (practically non-existent) spare time, she takes English lessons, which she often skips in order to deal with high-priority commitments. By managing her personal and work schedules online, Mary can receive news from her children's school and from her travel mates using a smart phone. Similarly, she can be informed about the tasks assigned to her and about which documents have been modified. However, how many different notifications is she going to receive, in a mixed order, belonging to such parallel contexts? Moreover, are all the notifications sufficiently urgent and important to be delivered, even though they have nothing to do with Mary's current activities? For instance, suppose that Mary 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 English course sends a confirmation note regarding the next lesson, should Mary be notified immediately or later on? Our work aims at providing users like Mary with an automated support for:

– Specifying the kind of awareness information that is the most appropriate, based on the user's current activity context.
– Tailoring the delivery of notifications and the application of filters to individual notification preferences.

2.2 Interruption management

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.

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, Czerwinski et al. (1991b) shows that subjects' performance is lower when interrupted by a task that is displayed in a similar way as the primary task. Furthermore, Czerwinski et al. (1991a) identifies an inverse relation between (primary and interruption) task similarity and people's ability to remember information about the interrupted task. Cutrell et al.

(2000) achieves similar results, showing that interruptions irrelevant to the main task result in longer times to process them and longer task resumption times than relevant messages. Finally, Mark et al. (2008) investigates 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. To conclude, all the reported works seem to converge toward the idea that interruptions have to be limited, as far as possible, for the user's well being.

2.3 Awareness support

Dourish and Bellotti (1992) define awareness as the understanding of the other users' activities, which provides a context for (one's) own activity. 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 (Gutwin et al. 1996). Gutwin and Greenberg (1999) defines the concept of workspace awareness as the up-to-the-moment understanding of another person's interaction with the shared workspace. According to the authors, the problem of maintaining workspace awareness in groupware revolves around obtaining useful information, rather than around how people use it. 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. Finally, Carroll et al. (2003) introduces the concept of *activity awareness* to represent the awareness of project work that supports group performance in complex tasks. 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.

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 (Scott 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.

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 (Scott 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, 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. Other works, e.g., Iqbal and Horvitz (2006, 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.

2.4 Context modeling

The research on context modeling and ubiquitous systems provides various definitions of context. Dey and Abowd (2000, 2001) define context as "any information that can be used to characterize the situation of an entity. An entity is a person, place, or object that is considered relevant to the interaction between a user and an application, including the user and application themselves". Zimmerman et al. (2007) refines Dey and Abowd's definition in order to specify possible categories of context information, such as individuality, activity, location, time and relations, and suggests an operational approach to their modeling. Bazire and Brézillon (2005) provides a formal specification of context, Abowd and Mynatt (2000) discusses context modeling issues and perspectives in ubiquitous systems and Baldauf et al. (2007) presents a survey on context-aware systems.

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).

We are interested in two of the dimensions introduced in Zimmerman et al. (2007): the first one is the activity context, i.e., what the user is doing and how this relates to her/his objectives and commitments. The second one is the specification of the relational contexts in which the user engages and of the artifacts and people associated to such contexts. These dimensions are the basis for the specification of notification management policies which take the user's behavior into account. However, they might be complemented with other features in order to enrich the adaptation capabilities of the awareness support tool. For instance, aspects of the individual context, such as time pressure and user device, could be taken into account in the presentation and filtering of notifications.

Several collaboration environments manage workspaces that group the entities and applications used to perform activities. Usually, a workspace is mapped to the structure of a project (e.g., in project management tools such as Collanos (2008) and ActiveCollab (2008)) or to a shared folder storing the documents to be manipulated (e.g., in groupware environments such as BSCW (Horstmann and Bentley 1997; OrbiTeam Software GmbH & Co. KG 2011)). However, more complex representations have been introduced to support the adaptation of the collaboration environment to its users' needs. For instance, Haake et al. (2010) presents a layered context model for the specification of domain knowledge, focal context information and adaptation rules supporting the provision of self-adaptive services. In our work, we start from a lightweight context representation, focused on the relations among users and artifacts, in order to offer context-dependent notification management in a scalable environment. However, this context representation could be extended to support other types of adaptation, such as the context-dependent configuration of applications and communication channels, depending on the users' activities; see Veiel et al. (2010).

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. Moreover, as not all ambiguity can be resolved using automatic techniques, a correct handling of ambiguous contexts will often need to collect feedback from the user, by means of a *mediation* dialog. In our work, we adhere to these guidelines: as described in the following sections, CONRAD supports both the specification of ambiguous contexts and the retrieval of user feedback aimed at correcting the system's recognition mistakes.

3 Awareness information support in the CONRAD framework

CONRAD supports awareness in open collaboration environments by allowing a user-centered visualization and management of the events concerning the user's activities. The core elements are (i) the explicit management of the user's activity contexts, which are the basis for the presentation of workspaces on the User Interface of the collaboration environment, and (ii) the tracking of the user's focus of attention across such contexts. We describe these aspects before focusing on notification policies.

3.1 Management of the user's activity contexts

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. In fact, as discussed in 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. Thus, besides traditional project management tools (e.g., Collanos 2008; ActiveCollab 2008), which are suitable for the execution of stable, well-structured projects, there is a need for tools supporting lightweight user cooperation and flexible team management.

In Ardissono et al. (2010a,b), we introduced two types of activity contexts in order to model such collaborations:

– The *activity frame* simplifies the notion of project in order to support lightweight and flexible cooperation.
  – 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.

  As discussed below, the state and content of an activity frame is displayed to the user in a workspace from which (s)he can manipulate artifacts and interact with collaborators. In order to provide such unified access point for the execution of activities, 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), T is the set of included tasks.

– The *task* supports interaction and synchronization aimed at achieving a goal. A user can define a task to specify and carry out the execution of a personal or a shared activity; e.g., writing a section of a report, or reserve the hotels for a holiday. A task may include artifacts to be manipulated and can have a deadline; moreover, it can be related to other tasks in partial order relations for scheduling purposes. 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 $(tn, U, O, Oi, g, P, T, s, d)$ tuple where: tn is the task name, U, O and Oi are the users sharing it and the associated objects (as above), 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 that the collaboration environment offers a tool supporting the user in the creation and management of activity contexts internally represented by means of the above described tuples. For this purpose, we exploit the Collaborative Task Manager service (CTM) presented in Ardissono et al. (2010a,b). The CTM is an open, interactive Web-based task manager. Similar to standard task managers (e.g., DoIt (DoIt.im 2011) and Things (Cultured Code 2011)), the CTM offers a User Interface which enables the user to create activity frames and tasks, relate them by means of
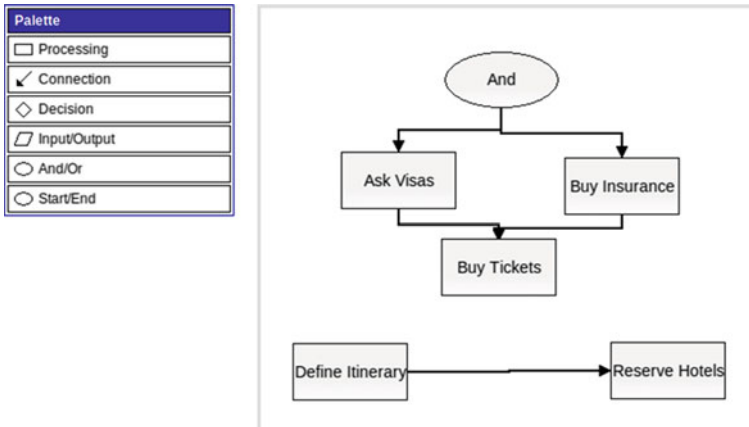
Fig. 1 Portion of the User Interface offered by the CTM for the specification and management of activity contexts

order dependencies (linking them in a graph), share them with other people and manage their life cycle; e.g., see the tour preparation plan in Fig. 1.[1] However, the CTM can also be integrated with business services to support the execution of operations from its User Interface. 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).

– The page of the CTM associated to c enables the user to send messages to the involved collaborators. Moreover, it enables her/him to drag and drop the links to the documents that have to be manipulated within c, or to launch business services in order to create/manipulate objects which are automatically associated to c.
– When the user creates/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. In this way, the object is included in the related workspace and can be accessed from it at any time.

As such, the CTM represents a unified access point to the user's workspaces, from which (s)he can handle personal and shared activities. Moreover, it is 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.

Indeed, 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: if o is shared with the exact same users sharing one (or more) of the user's activity frames/tasks, the CTM associates o to such contexts by updating the Oi element of their tuples.

---

[1] The UI of the CTM has been developed by exploiting the Jalava web-based diagram editor (http://jalava.buildyourownapps.com/), which supports the specification of diagrams including the typical constructs used in Workflow Nets and thus suitable for the specification of standard workflows (van der Aalst 1998).

3.2 Classification of awareness 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 shared 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 sharing a document participate in two collaborations, the document and its manipulation events could belong to any of them and the reference activity context of the actions is ambiguous. This issue can be addressed by offering an automatic support for specifying the context of the users' operations across applications. As previously described, we exploit the CTM to support the execution of business services within specific contexts. The CTM is thus used to classify awareness events accordingly.

We assume that each business service integrated in the collaboration environment is wrapped by an adapter (see Sect. 4.1) which collects the events generated by the service and translates them to a standard representation format used within the collaboration environment. Events are represented as lists of ⟨feature, value⟩ pairs specifying the event identifier, the originating application, the performed operation, its actor, arguments, and so forth. For instance, in the document upload example, the following type of event is generated[2]:

⟨⟨ID, *identifier*⟩, ⟨application, GoogleDocs⟩, ⟨operation, document-upload⟩, ⟨actor, $user-account$⟩, ⟨document, $document-URL$⟩, ⟨time, $time-stamp$⟩⟩.

Given such a representation, the CTM associates an event to one or more activity contexts (i.e., contextualizes the event) 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.

---

[2] In the example variables are emphasized. The internal representation is XML-based but we do not show it for readability purposes.

- As described later on in Sect. 3.5, the user can provide feedback about the context of an operation by correcting the classification of the awareness event (s)he receives as notification. 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 actions have to be made, leading to an uncertain classification.

  The CTM can implicitly associate events to contexts in various situations. For instance, 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. Moreover, 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 a frame/task hierarchy, "contexts" ("inferred-contexts") includes multiple items corresponding to the path between the root of the hierarchy and the current task.

3.3 Recognizing the user's current focus of attention

In order to apply notification management policies that take the user's behavior into account, at each instant of time the awareness support tool must 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; e.g., while editing a document. 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 actions and contextualized by the CTM.

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. Then, CF evolves on the basis of the occurrence of awareness events whose actor is the user, reflecting the most recent behavior. Specifically, for each awareness event ev generated by a user action, the CF of the actor is updated according to the algorithm shown in Fig. 2:

```
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
}
```

Fig. 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

A: If *ev* is explicitly associated to some contexts ("contexts" = $\{c1, \ldots, cn\}$), then *CF* is set to $\{c1, \ldots, cn\}$ because the event classification provides strong evidence about a focus shift.

B: If *ev* is implicitly associated to some contexts ("inferred-contexts" = $\{c1, \ldots, cn\}$):

– If $\{c1, \ldots, cn\} \cap CF \neq \emptyset$, *CF* is not updated with the new information because the event classification, uncertain, is compatible with the user's recent behavior.

– Otherwise, the event classification can only be explained by hypothesizing a focus shift. However, as the hypothesis is uncertain, *CF* is revised only if the following user event supports the same inference. In 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 actions, 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 actions. See Fig. 2 for details.

C: 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.

3.4 Notification management policies

CONRAD offers two types of awareness support:

– 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 preferences.
– An asynchronous awareness space, structured on the basis of the user's activity contexts, which presents the occurred events in Web pages supporting a multi-faceted search for information.

We skip the description of the awareness space, which is out of the scope of this paper. Concerning notification management, CONRAD offers four policies, described below. 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 actions:

– *Total filter*: all the notifications from the specified activity context are filtered out.
– *No filter*: all the notifications from the specified activity context are submitted as soon as they are generated by the services.
– *Context filter*: 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.
– *Task filter*: 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.

Notice that the "Context filter" policy is applied instead of the "Task filter" one when the user's current focus does not contain any task. Moreover, 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 $CF = \{\}$, the event is filtered out, because of lack of information about the user's activities.
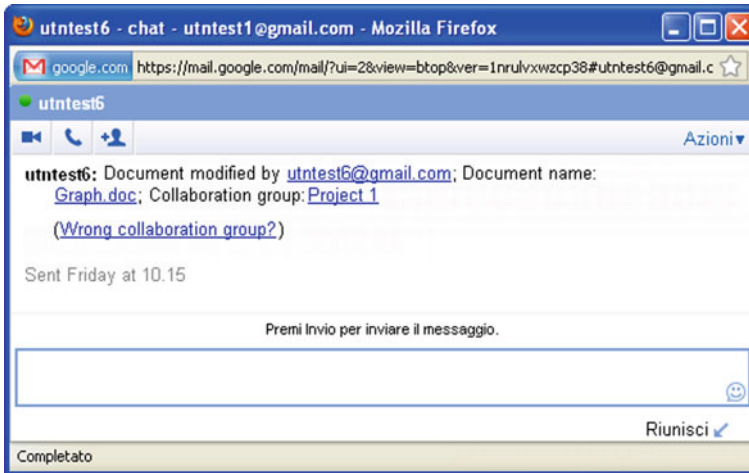
Fig. 3 A notification message

– Otherwise *CF* is compared to the event classification. If *CF* ∩ "contexts" ≠ ∅ or *CF* ∩ "inferred-contexts" ≠ ∅, 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 "no filter" policy as a default, because there is no available information about the context of the event.

3.5 Presentation of awareness events

As specified in Sect. 3.2, 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, to be used for presentation as notifications and/or in the awareness space.

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

– *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 of the activity context 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 Fig. 3, both the actor (utntest6@gmail.com)

and the edited document (Graph1.doc) are linked and directly accessible from the notification window.

– *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 Fig. 3). When the "inferred-contexts" feature is used to manage an awareness event, it 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 performed by other actors), (s)he can correct the system by clicking on the "Wrong collaboration group?" link of the notification. The user's feedback is taken into consideration to refine the specification of the involved activity contexts, which are the basis for the classification of the following events.

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

4 Technical details

4.1 Architecture

The CONRAD framework is based on the Personal Cloud Platform (PCP) (Ardissono et al. 2009c), 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 offered by the PCP 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 integrated in the environment (if technically possible) in order synchronize them without human intervention.

– For each user, a CTM 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). Different from traditional point-to-point interaction, this protocol prescribes that the 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 interested in receiving a certain type of information subscribe to it. The subscription consists of 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, it must be wrapped by an adapter which:

– Subscribes the component for relevant types of events; e.g., specification of collaboration groups.
– Intercepts the events and/or messages generated by the component (or polls it in order to retrieve them) and publishes 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 Sect. 3.2), the adapter also has to translate events and messages to such a format.
– Receives notifications from the hub and, depending on the type of information, invokes the APIs of the wrapped software component in order to perform the appropriate operations; e.g., injection of collaboration groups as contact lists.

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.[3]

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 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 components supporting awareness subscribes for the pieces of information concerning 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 receives the awareness information concerning U's operations on business services.

Figure 4 depicts the architecture of a collaboration environment integrating two external services: Service 1 and Service 2. 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, not shown for simplicity.

---

[3] Software libraries are available to support the interaction between the adapter and the hub, as well as content translation, which is based on XML Transformations (XSLT). Therefore, the portion of code to be developed for each application only concerns the interaction between adapter and software component.
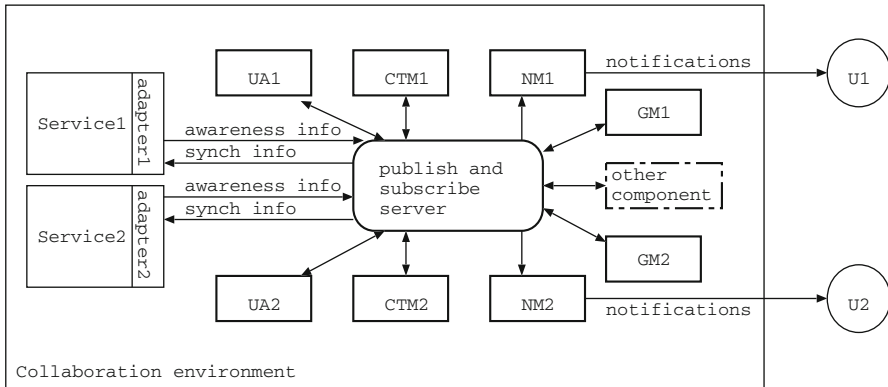
Fig. 4 Architecture of a collaboration environment based on the PCP and CONRAD. The architecture is depicted in a simplified form for readability purposes
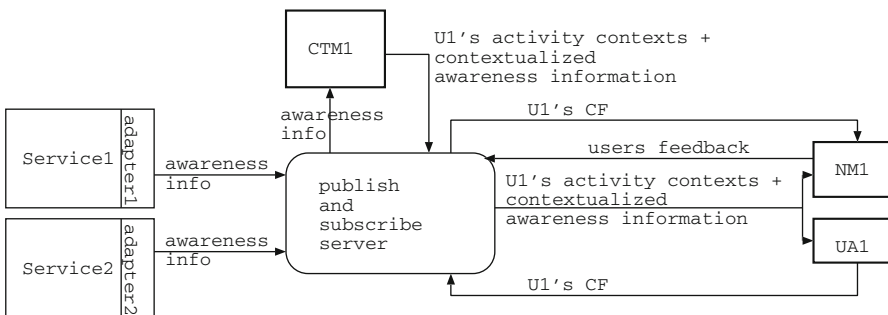
Fig. 5 Awareness information flow concerning user $U1$

4.2 Awareness information management

As described in Sect. 3.2, the CTM service classifies the awareness events generated by the integrated services in the user's activity contexts and extends events with the "contexts" and "inferred-contexts" features. Figure 5 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 Sect. 3.5, the user is allowed to correct the classification of the awareness events it receives as notifications by clicking on their "Wrong collaboration group?" links. If the user makes this 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.

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 (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.

5 Evaluation of the proposed notification management policies

We evaluated the notification management policies offered by CONRAD in a collaboration environment based on the Personal Cloud Platform and including 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).

While the total filter and no filter notification policies are standard for groupware environments and project management tools we were interested in analyzing the usefulness of the third and fourth policies, which are specific of our work.

Table 1 Treatment conditions

# Treatment	Condition order: first	Cond. order: second	Cond. order: third
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

5.1 The experiment

Three of the previously described notification policies (No filter, Context filter and Task filter) were empirically compared. The experiment was conducted to test a hypothesized causal relationship between these alternative UI design solutions and people's mental workload during a task.

Our research question was: "Does a different, context-dependent notification policy modify the level of workload on the users? If the answer is positive, which policy (or policies) can give best results?"

Hypothesis (Ha) 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.

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

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 and was calculated with a modified version of the NASA-TLX (Hart and Stateland 1988) questionnaire[4] at the end of each task. 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.

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

---

[4] 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.

situation. At the end of each task, the participant was given a 5 min 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 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 (simulated) collaboration groups.

Test users were 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 pay attention to these notifications coherently both with their attitudes and with their current status and needs during the experiment. 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

Table 2 Awareness events submitted in the experiment

# Event	Collaboration group	Notification policy
Document created	Project 2	No filter
Document modified	Project 2	No filter
Group member removed	Project 2	No filter
Confirmed lesson	English course	No filter
Group member added	Project 1	Context filter, no filter
Document modified	Project 1	Context filter, no filter
Document created	Project 1	Context filter, no filter
Confirmed meeting	Project 1	Task filter, context filter, no filter
Document modified	Project 1	Task filter, context filter, no filter

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 activity context.

All participants used the CONRAD prototype integrated in the PCP-based collaboration environment. They performed the task on a PC laptop (a Dell Latitude E6400 equipped with Windows XP) with the built-in monitor as the display (14.1-in LCD, 1280×800 pixel resolution). Sessions were recorded with a desktop capture program. The experimenter noted all interesting comments by the users while sitting at some distance from them.

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 were available as a reference throughout the experiment. Each participant was engaged in the testing activity for a period of about 1 h.

5.2 Results

Figure 6 shows the results of the questionnaire we proposed to the test participants. 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. Three values are reported for each user, one for each treatment condition. Figure 7 shows mean values for each treatment condition and reports standard deviation values.

We used Bartlett test to assure homogeneity of variances and Mauchly's 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.
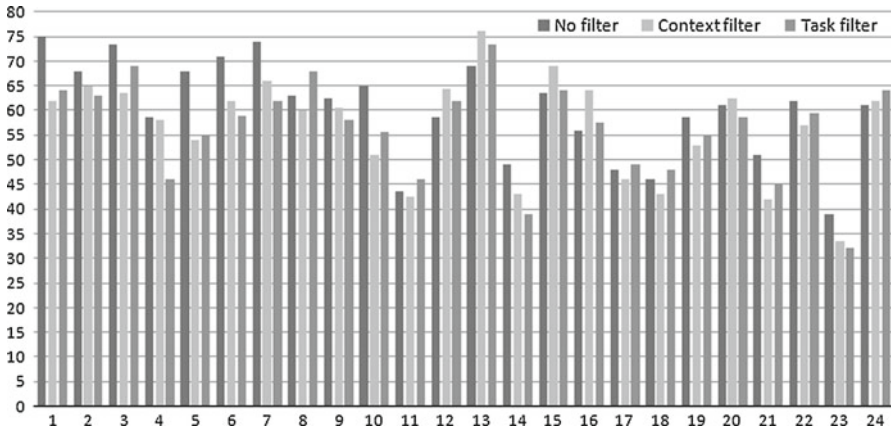
Fig. 6 Workload per user: the x-axis represents users and the y-axis shows the values of the expressed workload

	No filter	Context filter	Task filter
Mean workload	60,19	56,67	56,35
Standard deviation	9,72	13,84	9,92
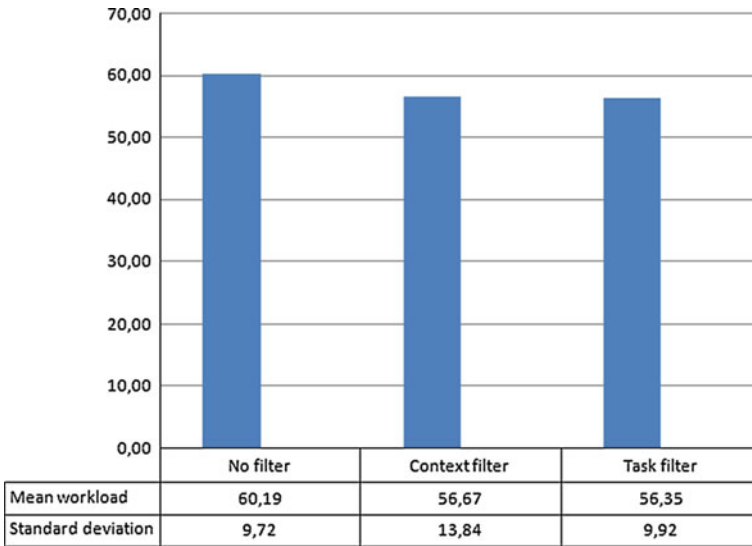
Fig. 7 Mean values and standard deviation of workload, for each treatment condition

– Bartlett's p-value was $0.97 > 0.05$, so we assume that variances are equal.
– Mauchly's 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 the significant differences 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 ($p = 1.00$).

5.3 Discussion

The results of our experiment revealed that the context and task filter reduced users' workload with respect to the non filtered condition, thus evidencing how context-dependent filtering could be helpful to users. Nevertheless, 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 also 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 discussion because, even though they do not allow the formulation of other statistical claims, they give some interesting insights. Specifically, it is evident from the description of the experiment that, of all the delivered notifications, only the histogram completion in Project 1 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 to the second one, probably frustrated by the lack of immediate utility of the first 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 it 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.

The results of the experiment also gave us interesting insights for future research, among which the identification of alternative presentation modalities for displaying the notifications delivered to the user. During the experiment, users seemed puzzled because they could not immediately detect the reference activity context of the notifications and thus they could not quickly evaluate their importance. Indeed, all the notifications were presented in pop-up windows (see Fig. 3), regardless of the fact that they were relevant to the user's current activities or not. Therefore, the user had to read them thoroughly (and in particular the "Collaboration group" element) in order to

identify their context. As stated before, under the "no-filter" condition, this lead some users to minimize such pop-up windows when they appeared, without reading them or after reading just a couple of them, which were not related with their current task.

We plan to improve the presentation modalities in order to support an easy identification of the context of the incoming notifications. For instance, notifications unrelated to the user's current focus might be delivered as automatically fading balloons, in order to be less intrusive. 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.

5.4 Threats to validity

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

– 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.

6 Related work

Most groupware environments are based on closed architectures and support awareness limited to their native applications; e.g., Project Vie IM (Scupelli et al. 2005),

Collanos (2008), Feng Office (2010) and ActiveCollab (2008). As such environments cannot be extended with external business services, they expose the user to a fragmented view of her/his workspaces, for each non-native application. 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.

In the development of CONRAD, we focused on 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 reflects traditional project management and can be hardly applied to the dynamic types of collaboration emerging in modern working scenarios (Prinz et al. 2006). Moreover, it is unsuitable for dealing with ordinary collaborations in informal environments, such as the establishment of ephemeral collaboration groups 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.

The importance of developing open, generic awareness support infrastructures is advocated for in other projects. For instance, 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 order to address this issue, Gross and Prinz (2004) proposes a method for modeling the contexts in which users operate. In that work, the attributes of the awareness events (actor, application, etc.) are matched against the context descriptions to identify the closest one and manage events accordingly. Our work differs in two ways: first, CONRAD merges the context information provided by business services (e.g., contact lists, etc.) to determine the reference activity contexts of the events. Second, following the guidelines in Dey and Mankoff (2005), CONRAD involves the user in the loop by accepting corrections when events are misclassified. In this way, the system can repair the occurred mistakes.

Similar to our approach, Atmosphere (Rittenbruch 2002) introduces the concept of *contextual awareness* to explicitly relate objects and awareness events to contexts. However, it introduces an abstract notion of *contextor* 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 business services (e.g., "final review" of a document). 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, which (i) 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)), and (ii) is 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).

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", "negotiated", "mediated", and "scheduled". In the mediated solution an autonomous agent decides when to interrupt the user. The notification management policies offered by our framework belong to the mediated category, with the Notification Manager playing the mediator role.

Other approaches, based on the mediation of notifications, attempt to address the trade-off between interruptions and awareness:

– Many groupware and project management tools (e.g, BSCW (Horstmann and Bentley 1997; OrbiTeam Software GmbH & Co. KG 2011), Project Vie IM, Collanos, ActiveCollab, TeamWox) support a project-based organization and filtering of notifications. They attempt to reduce the number of messages delivered to the user by enabling her/him to specify the workspaces from which (s)he wants, or does not want, to receive awareness information. Such notification policies do not take the user's behavior into account for reducing the information overload; moreover, they have to be explicitly updated by the user in order to reflect possible changes in interests. In comparison, 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.
– In Wang et al. (2007), the BSCW awareness support is extended by (i) defining dynamic notification preference profiles which model the user's long-term interests in workspaces and (ii) 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.
– 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. Moreover, 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. In such works, the priority of the interruptions is computed on the basis of a set of features a-contextually specified by the user; e.g., if the sender of an e-mail is my boss, then the message has high priority. Our work handles notifications on the basis of their context, as this is a main factor for determining the possible relevance of a notification. For example, 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. Of course, the context-based filtering of notifications might be complemented with a feature-based selection to support the development of finer-grained notification policies.

– Bailey et al. (2006) describes a framework for specifying and monitoring user tasks, based on user-defined models of the task structure. Moreover, Iqbal and Bailey (2007) describes 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). Currently, we do not deal with this aspect because mapping event types to breakpoint evidence is very difficult for the administrator of an open collaboration environment and thus hardly applicable in realistic use cases. Anyway, the loosely-coupled model applied for integrating business services, based on the development of adapters, supports this kind of extension.

– 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). While CONRAD takes a similar approach with respect to the integration of heterogeneous notification sources (in our case, the business services of the collaboration environment), so far we have not investigated the other issues addressed in CASSIUS.

Collaboration environments also are in the scope of intelligent and adaptive systems. Intelligent tutors (Harrer et al. 2006) are applied to collaborative, single-task domain, learning environments: conceptual understanding, visual organization, task coordination, task coherence and task selection are identified as elements for supporting tutoring with abstraction from directly captured data. These elements can be seen as closely related to our approach of goal decomposition, with the consequent creation of tasks and subtask hierarchies. However, they remain inherently domain-specific and data-driven. Instead, as our approach makes the user responsible for defining tasks and offers basic support to user coordination, it does not require any specific domain knowledge.

7 Conclusions

The provision of awareness information about users' activities has to face a trade-off between informing people about the state and evolution of their collaboration contexts and interrupting them with notifications which might have a disruptive effect on their attentional and emotional states. The CONRAD awareness support framework attempts to address this issue by providing a holistic management of awareness information in open collaboration environments.

CONRAD offers a Web-based awareness space and the possibility to filter the notifications submitted to the user by means of context-dependent policies. This support is based on an explicit management of the user's collaboration contexts, on the contextualization of awareness information and on the recognition of the user's focus of attention across workspaces.

We proposed two context-dependent notification policies, the Context filter and the Task filter, which select the notifications to be delivered on the basis of the user's current focus of attention at different granularity levels. We did an empirical evaluation of these policies, confronting them with a non-filtered situation. The experiment

revealed that our policies reduce users' workload with respect to the baseline condition. Moreover, they provide users with useful information for catching up with the events occurred in their collaboration contexts.

In our future work we plan to investigate the adoption of different presentation modalities for notifications in order to enhance the usability of our notification system. For instance, we plan to present peripheral information in less intrusive communication channels with the aim of further reducing the disruptive effect of notifications on the user. Moreover, we want to investigate whether individual differences in users can lead to different preferences in notification modalities. Furthermore, we plan to investigate models for visualizing the state and evolution of the user's activity contexts. This is a complementary feature with respect to notification management, as it supports users in resuming the state of the collaborations they have ignored for some time.

In our future work we also intend to investigate the user's privacy preservation issue (Kobsa 2007) in order to support the management of individual privacy protection policies. In fact, at the current development stage, our awareness framework only applies standard group-based restrictions on the propagation of personal information.

References

Abowd, G., Mynatt, E.: Charting past, present and future research in ubiquitous computing. ACM Trans. Comput. Hum. Interact. Special Issue on HCI in the new Millennium **7**(1), 29–58 (2000)

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

Adamczyk, P., Bailey, B.: 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), pp. 271–278. ACM (2004)

Allen, D.: Getting Things Done: The Art of Stress-Free Productivity. Penguin, Camberwell (2003)

Ardissono, L., Bosio, G., Goy, A., Petrone, G., Segnan, M.: Context-aware notification management in an integrated collaborative environment. In: Proceedings of UMAP Workshop "Adaptation and Personalization for Web 2.0", pp. 21–30. Trento, Italy (2009a)

Ardissono, L., Bosio, G., Goy, A., Petrone, G., Segnan, M.: Managing context-dependent workspace awareness in an e-collaboration environment. In: Proceedings of WI/IAT09 Workshop "Intelligent Web Interaction" (IWI 2009), pp. 42–45. IEEE, Milano (2009b)

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

Ardissono, L., Bosio, G., Goy, A., Petrone, G., Segnan, M.: Open, collaborative task management in Web 2.0. In: Proceedings of MCCIS 2010 IADIS Multiconference on Computer Science and Information Systems, pp. 20–27. IADIS Press, Freiburg (2010a)

Ardissono, L., Bosio, G., Goy, A., Petrone, G., Segnan, M., Torretta, F.: Collaborative service clouds. Int. J. Inf. Technol. Web Eng. **5**(4), 23–39 (2010b)

Ardissono, L., Goy, A., Petrone, G., Segnan, M., Torasso, P.: INTRIGUE: personalized recommendation of tourist attractions for desktop and handset devices. Appl. Artif. Intell. Special Issue Artif. Intell. Cult. Heritage Digit. Libr. **17**(8–9), 687–714 (2003)

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

Bailey, B., Iqbal, S.: Understanding changes in mental workload during execution of goal-directed tasks and its application for interruption management. ACM Trans. Comput. Hum. Interact. **14**(4), 1–28 (2008)

Bailey, B., Konstan, J., Carlis, J.: Measuring the effects of interruptions on task performance in the user interface. In: Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics, pp. 757–762. IEEE Press, Nashville (2000)

Bailey, B., Konstan, J., and Carlis, J.: The effects of interruptions on task performance, annoyance, and anxiety in the user interface. In: Proceedings of Human-Computer Interaction—INTERACT'01, pp. 593–601. IOS Press, Tokyo (2001)

Baldauf, M., Dustdar, S., Rosenberg, F.: A survey on context-aware systems. Int. J. Ad Hoc Ubiquitous Comput. **2**(4), 263–277 (2007)

Bartram, L., Ware, C., Calvert, T.: Moticons: detection, distraction and task. Int. J. Hum. Comput. Stud. **58**(5), 515–545 (2003)

Bazire, M., Brézillon, P.: Understanding context before using it. In: Dey, A., Kokinov, B., Leake, D., Turner, R. (eds.) LNCS 3554, Modeling and Using Context, pp. 29–40. Springer-Verlag, Berlin (2005)

Bosio, G.: 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), pp. 1–4. IEEE, Nice (2010)

Carroll, J., Neale, D., Isenhour, P., Rosson, M., Scott McCrickard, D.: Notification and awareness: synchronizing task-oriented collaborative activity. Int. J. Hum. Comput. Stud. **58**(5), 605–632 (2003)

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

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

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

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

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

Cutrell, E., Czerwinski, M., Horvitz, E.: Effects of instant messaging interruptions on computing tasks. In: Proceedings of CHI '00 Extended Abstracts on Human Factors in Computing Systems, pp. 99–100. ACM, The Hague (2000)

Czerwinski, M., Christman, S., Rudisill, M.: Interruptions in Multitasking Situations: The Effects of Similarity and Warning. Technical Report JSC-24757. NASA Johnson Space Center, Houston (1991a)

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

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

Dey, A., Abowd, D.: 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)

Dey, A., Abowd, D.: A conceptual framework and a toolkit for supporting the rapid prototyping of context-aware applications. Hum. Comput. Interact. Special Issue Context Aware Comput. **16**(2), 97–166 (2001)

Dey, A., Mankoff, J.: Designing mediation for context-aware applications. ACM Trans. Comput. Hum. Interact. **12**(1), 53–80 (2005)

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

Dix, A.: Challenges for cooperative work on the Web: an analytical approach. Comput. Support. Coop. Work J. Collab. Comput. **6**, 135–156 (1997)

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

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

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

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

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

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)

Grimes, A., Brush, A.: Life scheduling to support multiple social roles. In: Proceedings of CHI 2008, pp. 821–824. ACM, Florence (2008)

Gross, T., Prinz, W.: Modelling shared contexts in cooperative environments: concept, implementation, and evaluation. Comput. Support. Coop. Work **13**(3–4), 283–303 (2004)

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

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

Haake, J., Hussein, T., Joop, B., Lukosch, S., Veiel, D., Ziegler, J.: Modeling and exploiting context for adaptive collaboration. Int. J. Coop. Inf. Syst. **19**(1–2), 71–120 (2010)

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

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

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

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

Horvitz, E., Apacible, J., Subramani, M.: 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, pp. 433–437. Springer-Verlag, Edinburgh (2005)

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

Iqbal, S., Bailey, B.: Understanding and developing models for detecting and differentiating breakpoints during interactive tasks. In: Proceedings of SIGCHI Conference on Human Factors in Computing Systems, pp. 697–706. ACM , San Jose (2007)

Iqbal, S., Bailey, B.: Effects of intelligent notification management on users and their tasks. In: Proceedings of 26th Annual SIGCHI Conference on Human Factors in Computing Systems, pp. 93–102. Firenze, Italy (2008)

Iqbal, S., Horvitz, E.: Leveraging characteristics of task structure to predict the cost of interruption. In: Proceedings of Conference on Human Factors in Computing Systems, pp. 741–750. ACM, Montreal (2006)

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

Iqbal, S., Horvitz, E.: 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), pp. 27–30. ACM, Savannah (2010)

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

Kantor, M., Redmiles, D.: 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)

Kobsa, A.: Privacy-enhanced Web personalization. In: Brusilovsky, P., Kobsa, A., Nejdl, W. (eds.) The Adaptive Web: Methods and Strategies of Web Personalization, Lecture Notes in Computer Science, Vol. 4321, pp. 628–670. Springer-Verlag, Berlin (2007)

Latorella, K.: Investigating Interruptions: Implications for Flightdeck Performance. Technical Report TM-1999-209707. NASA, Washington (1999)

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

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

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

McFarlane, D., Latorella, K.: The scope and importance of human interruption in human-computer interaction design. Hum. Comput. Interact. **17**(1), 1–61 (2002)

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

Pendyala, V.S., Shim, S.S.Y.: The Web as the ubiquitous computer. Comput. Now **42**, 90–92 (2009)

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

Prinz, W., Löh, H., Pallot, M., Schaffers, H., Skarmeta, A., Decker, S.: ECOSPACE—towards an integrated collaboration space for eProfessionals. In: Proceedings of 2nd International Conference on Collaborative Computing: Networking, Applications and Worksharing, pp. 39–45. IEEE, Atlanta (2006)

Rittenbruch, M.: Atmosphere: a framework for contextual awareness. Int. J. Hum. Comput. Interact. **14**(2), 159–180 (2002)

Roda, C., Thomas, J.: Attention aware systems: Theories, applications, and research agenda. Comput. Hum. Behav. **22**(4), 557–587 (2006)

Scott McCrickard, D., Chewar, C.M.: Attuning notification design to user goals and attention costs. Commun. ACM **46**, 67–72 (2003)

Scott McCrickard, D., Czerwinski, M., Bartram, L.: Introduction: design and evaluation of notification user interfaces. Int. J. Hum. Comput. Stud. **8**, 509–514 (2003)

Scupelli, P., Kiesler, S., Fussell, S., Chen, C.: Project View IM: a tool for juggling multiple projects and teams. In: Proceedings of CHI'2005: Human Factors in Computing Systems, pp. 1773–1776. ACM, Portland (2005)

TeamWox: TeamWox GroupWare. http://www.teamwox.com (2011)

Tran, M., Raikundalia, G., Yang, Y.: Using an experimental study to develop group awareness support for real-time distributed collaborative writing. Inf. Softw. Technol. **48**(11), 1006–1024 (2006)

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

van der Aalst, W.: The application of Petri Nets to workflow management. J. Circuits Syst. Comput. **8**(1), 21–66 (1998)

Veiel, D., Haake, J., Lukosch, S.: Facilitating team-based adaptation of shared workspaces. In: Proceedings of 2010 International Symposium on Collaborative Technologies and Systems (CTS), pp. 275–284. IEEE, Chicago (2010)

Wang, Y., Gräther, W., Prinz, W.: Suitable notification intensity: the dynamic awareness system. In: GROUP'07: Proceedings of the 2007 International ACM Conference on Supporting Group Work, pp. 99–106. ACM, New York (2007)

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

Zimmerman, A., Lorenz, A., Oppermann, R.: An operational definition of context. In: CONTEXT 2007, Lecture Notes in Artificial Intelligence n. 4635, pp. 558–6571. Springer-Verlag, Berlin (2007)

Zimmermann, A., Lorenz, A.: LISTEN: a user-adaptive audio-augmented museum guide. User Model. User-Adapt. Interact. **18**(5), 389–416 (2008)

Author Biographies

Liliana Ardissono is an Associate Professor at the Dipartimento di Informatica of the University of Torino, where she obtained her university degree and her Ph.D. in Computer Science. Her research interests include user modeling, adaptive hypermedia, service-oriented computing and cloud computing. She is the Secretary of the Board of Directors of User Modeling Inc. and she is a member of the Editorial Board of User Modeling and User-Adapted Interaction: The Journal of Personalization Research.

Gianni Bosio is a Ph.D. student at the Dipartimento di Informatica of the University of Torino. He obtained his master degree in communication sciences at the Universití di Torino. His research interests include human-computer interaction, cloud computing, computer supported collaboration technologies and user modeling.