

Context-Aware Notification Management in an Integrated Collaborative Environment

Liliana Ardissono, Gianni Bosio, Anna Goy, and Giovanna Petrone

Dipartimento di Informatica, Università di Torino, Italy
{liliana,bosio,goy,giovanna}@di.unito.it,
WWW home page: <http://www.di.unito.it/Anagrafica>

Abstract. The e-collaboration tools available in open environments offer services supporting the interaction and the synchronization between users, who are typically involved in parallel activity spheres; e.g., different projects and home schedules. However, such tools provide the user's with separate views on her activities and collaborations; moreover, they support workspace awareness by delivering unstructured notification streams, which challenge the user's attention and cannot be filtered or grouped on a relevance basis.

As an answer to this issue, we present an Integrated Collaborative Environment offering a unified view of the user's parallel collaboration spheres. This environment includes a notification management model supporting the selective deferral of notifications on the basis of the user's focus of attention.

1 Introduction

With the large availability of wireless connectivity, broad band internet connections and mobile devices, people result being on-line most of the time. This Web presence offers opportunities to manage different spheres of activity, concerning both work collaborations, as done in project management and cooperative work (e.g., see [1]), and personal commitments related to the users' personal life. For instance, as described in [2], working parents increasingly tend to handle life scheduling needs as a "holistic management of personal, family and professional schedules across settings and calendaring tools". Moreover, people exploit various Web applications to share photos, and other kinds of documents, with groups of friends, as well as with larger communities.

Currently, many applications support the synchronous and asynchronous interaction between users; e.g., Instant Messaging, Shared Workspaces, Forums, e-mail, audio and video Web conferences, and similar. However, with the exception of some Project Management vertical tools [3, 4] and domain-specific awareness support tools [5], these applications are mostly available as distinct services which can hardly be integrated in a unified environment supporting the user's activities. Thus, the user is provided with separate views on the state of the collaborations she is involved in. Specifically:

- Each tool separately handles a local definition of the collaboration groups and most tools are unable to import the group definitions from other tools.¹ Thus, the user is forced to manage multiple instance of her collaboration groups.

¹ Unless they are strictly related, such as some Google Apps.

- Each tool presents the state of the user's collaborations concerning the kind of activities it supports; e.g., calendar information, versus document sharing information. Thus, the user is provided with a partial view on the overall set of events concerning her collaboration spheres (e.g., activities, commitments, and tasks to be performed).
- As each tool separately handles its own awareness information, the user is overloaded with a flow of unrelated notifications, which cannot be filtered on a relevance basis, or managed by following a specific notification policy.

In order to get an overview of the events occurred within a specific activity sphere (e.g., those concerning the presence of collaborators, and the scheduling of meetings), the user has therefore to extract the relevant awareness information from the parallel streams of notifications generated by the collaboration tools she exploits. This issue becomes even more relevant if the user uses multiple tools supporting the same collaboration functions, e.g., one at work and a different one at home, as all such information streams have to be fused as well. For instance, [2] reports the difficulties in importing feeds from external calendars (e.g., those used by the children's school to schedule meetings) in the user's working one.

The above discussion highlights the need for collaboration environments that enable users to manage all their spheres of activity, by supporting the integration of external applications, and offering a unified awareness support. The last aspect is particularly important to make awareness information easily accessible and to prevent the user from being overloaded by flows of unrelated notifications.

As an answer to this issue, we developed an open Integrated Collaboration Environment (ICE) supporting e-collaboration in multiple spheres of activity. The ICE is based on the integration of a set of collaboration tools and can be extended with additional applications, in order to provide new collaboration features, or to comply with specific user requirements. Given the set of integrated applications, the ICE manages a unified view of the state of the collaborations the user is involved in and it provides a context-aware delivery of the awareness information to the user. This is achieved by:

- Replacing the subjective view on collaboration groups, which most collaboration tools offer, with a centralized management of the activity spheres and of the associated user groups.
- Enabling the fusion of the awareness information generated by each of the integrated applications. This fusion is based on the introduction of an agent, the *Notification Manager*, that acts as an intermediary between the user and the collaboration environment and generates personalized notifications for the user.

This paper focuses on the provision of awareness information and presents the notification management model developed in our ICE to adapt the awareness support to the user's notification preferences. This model is based on alternative mediation policies [6], which can be selected by the users. In particular, we introduce a selective deferral of notifications based on their relevance to the sphere of activity representing the user's focus of attention.

In the following, Section 2 deals with awareness in e-collaboration environments. Section 3 describes the ICE and the awareness management. Section 4 provides some technical details. Sections 5 and 6 describe the related research and conclude the paper.

2 Workspace awareness and interruption management

2.1 Background

The effects of interruptions on people's activities have been thoroughly studied in the literature: it has been repeatedly noted that an interruption has a disruptive effect on both a user's task performance and emotional state [7–9].

Interruptions are particularly critical in collaboration environments, which base the awareness support on the delivery of notifications to their users. For instance, in Computer Supported Cooperative Work, the notifications of other people's activities support group awareness during synchronous or asynchronous collaboration [11, 12].

A collaborative workplace poses novel issues concerning interruptions: coworkers may be involved in multiple tasks, belonging to different projects [1], therefore multiplying potential interruptions from colleagues or automatic agents. In addition to commonly used e-mail and instant messaging, other software agents such as shared calendars and shared maps can become a source of notifications and, thus, of interruptions.

2.2 Evaluation of notification management policies

The previous discussion suggests that a correct handling of interruptions is critical to achieve a balanced trade-off between interruptions and awareness. In this perspective, we analyzed the impact on users of a set of notification policies providing different filtering criteria for the organization and presentation of the awareness information. Our hypothesis was that the overhead on users might be reduced by mediating the notification delivery. As a collaboration environment can be used to manage parallel activity spheres, each one generating its own awareness information, we hypothesized to filter and defer notifications on a contextual basis. This led us to hypothesize some context-based notification management policies that could be robust enough to significantly reduce the disruptive impact of interruptions in user's work, but also as flexible as to give the user an acceptable level of awareness of her collaborators' activities.

We performed a test with final users (21 participants, 11 males and 10 females) to evaluate the effects of interruptions by notifications in a collaboration environment. Users were divided into three groups of 7 participants each, and each group experimented a different notification policy:

- In the *no filter* situation, all the notifications, from all the projects the user was involved into, were submitted (7 notifications from 3 different spheres, originated from regular users and administrators).
- In the *context filter* situation, only the notifications from the user's focus of attention (i.e. the project she was actually working at) were submitted (3 notifications from the sphere of activity of the user, originated from regular users and administrators).
- In the *priority filter* situation, the notifications of administrators' activities from all projects, plus those included in the context filter, were submitted (4 notifications from 3 different spheres; notifications from administrators were considered with high priority and submitted even if originated from spheres of activity different from the user's focus).

All the filtered notifications, if any, were displayed to the user when she completed her main task, in the form of a single e-mail message.

Users were asked to perform a simple task (alphabetically sorting a list of names) belonging to a shared project, and they were told that this had to be their main focus of attention. They were also instructed that they were involved in two other projects (planning a conference and planning their participation to an English class) and that notifications of other people's activities concerning the same projects could interrupt them.

Notifications pop-up windows were displayed in the low-right corner of the screen. Users were told to behave in the most normal and spontaneous way when reacting to an interruption. For example, they could choose to access their e-mail application to visualize the full text of the e-mail and eventually reply to it or just ignore the notifications, proceed with the primary job and then process all e-mails once their primary job was fully accomplished, according to their personal attitude and current state.

All the interruptions were simulated by the experimenter in a *Wizard of Oz* modality; to be able to monitor a subject's on-screen activity, a RealVNC server was installed on the subject's computer. The experimenter, through a client application, watched the subject's task execution and simulated interrupting events in real time.

At the end of the test, a NASA-TLX survey was submitted to the users in order to evaluate their total subjective workload [13]. We analyzed the difference in the mean workloads between the three groups. The mean workload expressed by users in the no filter situation was particularly high (mean = 57.55 in a scale from 0 to 100). Mean workload did not significantly decrease in the priority filter situation (mean = 48.68, $T = 0.8371$, $p = 0.430$), while a significant difference was noted between no filter situation and the context filter situation (mean = 36.96, $T = 3.3575$, $p = 0.012$).

The context filter emerged as the best choice for our notification policy. The priority filter (which featured only one more interruption than the context filter) performed particularly bad, especially with users that had no previous experience at working in shared ambients, and was therefore discarded.

3 The Integrated Collaboration Environment

Our prototype ICE supports the coordination of personnel activities (professors, students, etc.) within a University Department. The ICE currently includes a calendar management application, a document sharing tool and a process management component which handles the workflows of two University projects.

3.1 Architecture

The ICE is based on the SynCFr environment for the synchronization of applications [14], which supports the sharing of context information among applications, based on the Publish and Subscribe pattern. The context information includes: (i) business data and (ii) synchronization information (concerning, e.g., the events occurring within the applications). The integration of a software component in the environment is performed by wrapping it with an adapter which addresses interoperability issues and enables the

component to subscribe for the relevant context information, and to publish the one it generates; see [14].

Our ICE extends SynCFr with the integration of a set of components supporting the user's collaboration and the management of workspace awareness. In particular:

- The *User Agent* manages the identities and the notification preferences of the users registered in the ICE. Moreover, the agent tracks the sphere of activity in their focus of attention, while they operate within the environment; see Section 4.2.
- The *Group Manager* supports the users in the definition of the spheres of activity and of their associated collaboration groups. The spheres can be private or shared with other registered users. We assume that, within an organization, a set of public spheres is defined to organize projects and other similar activities; moreover, any registered user can create her own spheres, to integrate the management of personal commitments with the workplace ones.
- The *Notification Manager* mediates the delivery of notifications to the user, according to the notification preferences stored in the User Agent.

3.2 Notification management policies

The policies applied in our ICE are aimed at deciding whether a notification should be immediately delivered or it should be deferred. According to the results of the tests described in Section 2.2, the criterion used to steer the deferral of notifications is their relevance to the sphere of activity the user is focusing on. In particular, the ICE offers the following policies, which the user can explicitly select:

- The default policy is the *context filter*.
- The user can however set as her notification preference the *no filter* policy.
- Furthermore, the user can keep the *context filter* as a default, but she can apply the *no filter* policy to one or more specific spheres of activity.

In the management of deferrals, two main factors should be taken into account: on the one hand, as noticed in [10], the burst of user activity on a task typically lasts a short time, after which she can be interrupted with less disruptive effects. Moreover, users should be enabled to select themselves the latency to be applied in the deferral. On the other hand, the users work in multitasking (see, e.g., [15]); in order to support workspace awareness effectively, at each focus shift they should be informed about the deferred notifications concerning the new focus of attention.

Given such requirements, our ICE enables the user to select the maximum amount of time a notification can be deferred. While the user operates in the ICE, she is notified about all the deferred notifications as soon as one of them reaches the deadline. Moreover, the environment delivers all the deferred notifications at each focus shift.

At the actual stage of development, when a set of deferred notifications has to be delivered, it is reported in a single e-mail message, in a format supporting the user in the inspection of a possibly long list of messages. The message is an interactive web page in which the notifications are grouped by sphere of activity. For this purpose, the page contains a set of clickable tabs, one for each sphere, among which the user can switch by means of a click. In order to highlight the notifications concerning the user's focus

of attention, the corresponding tab of the page is presented as the front one. Each tab includes a list of message headers, available as links. For each message, the following information is displayed:

- *subject*, including the application that generated the notification and the object;
- *sender*, i.e., the user who originated the notification;
- *date*.

4 Technical details

The management of the notification policies is based on three elements:

1. The association of the events generated by the ICE applications to their reference spheres of activity; see Section 4.1.
2. The recognition of the user's *focus of attention*, i.e., of the sphere of activity she is working at, at any given moment; see Section 4.2.
3. The mediated notification management; see Section 4.3.

4.1 Contextualization of events

For each application integrated in the ICE, the adapter wrapping the component is in charge of tagging the events it generates with the sphere of activities they belong to, or with the list of users involved in the event, depending on the kind of component.

Specifically, if the component explicitly manages contexts (e.g., process management components do that), the adapter can tag such events appropriately. However, most collaboration tools only support the sharing of objects with sets of users; e.g., documents in GoogleDocs. In this case, the wrapper tags the event with the list of users sharing the object. If the same users participates to more than one sphere of activity, the event is implicitly associated to all such spheres, and thus ambiguously tagged.

4.2 Analysis of the user's behavior

From the viewpoint of the notification management, the user's behavior in the ICE is summarized by two context variables:

- The user's *activity status* specifies if the user is active, idle, or off line.²
- The *focus of attention* stores the systems's hypotheses on which sphere of activity the user is working at: this is a list of spheres, representing alternative hypotheses, and is empty when the user is off line.

The User Agent associated to a user U receives the context information about U 's activities available in the Cross-Application Context and it analyzes such information in order to update her focus of attention (henceforth, F):

² This is sensed by the Instant Messaging application embedded in the ICE.

- When the User Agent receives the notification that the user is active, it initializes F with all the user's spheres of activity. Moreover, it sets the history (H) of the discarded hypotheses to the empty list. H is a buffer of discarded hypotheses, which might need to be rescued, given the new evidence about the user's behavior.
- Each time the Agent receives a new piece of context information describing a user action (e.g., she has uploaded a document in the shared space), it analyzes the reference groups (henceforth, RG) of the event.³ Then, it updates F and H accordingly. We assume that, if there is no evidence of a focus shift, then the user is continuing to work within the same sphere of activity (continuity assumption). Thus, F and H are updated as follows:
 - If $F \cap RG \neq \emptyset$, we hypothesize that the new evidence contributes to restrict the focus of attention. Let F' and H' denote the updated values of F and H , respectively. Then, $F' = F \cap RG$. Moreover, $H' = F \cup RG - F'$. This means that the history is cleaned; then, it is set to the hypotheses just discarded from the focus of attention plus those provided by the new event which were not included in F' because they were not consistent with the continuity assumption (they introduced new focus hypotheses). For instance, suppose that H is $\{G3\}$, that the focus is $\{G1, G2\}$ and that the new event is tagged as $\{G1, G4\}$. Then $F' = \{G1\}$ and $H' = \{G2, G4\}$.
 - If $F \cap RG = \emptyset$, we assume that the user has shifted to a new sphere of activity. Thus, $F' = RG$.

Notice that the ICE components publish events concerning both the actions performed by the user and those triggered within the applications she uses. For instance, Google-Docs can be polled to retrieve events of type [Document X uploaded by user Y at time T] each time a user saves a new copy of a document X , or the document is automatically saved by the application. Thus, the User Agent receives a regular flow of evidence while the user is active in the ICE. When the flow of activities stops, the Agent sets the focus of attention to the empty list.

4.3 Notification Management

The Notification Manager handles the notifications directed to each user registered in the ICE by filtering them according to her preferences. Each time an event concerning the user is published in the Cross-Application Context, the Notification Manager operates as follows:

- If the user is off line, it stores the event in an internal buffer; when the user is on line again, it discards all the events older than 24 hours; then, it merges the other ones (cleaned from redundancies) into a message, structured as described in Section 3.2, and sends the message to the user by e-mail.
- If the user is on line, the notifications are delivered, or deferred, depending on her policy preferences:
 - If the user has selected the *no filter* policy, the Notification Agent notifies the user by generating an Instant Message via the IM application.

³ This is done by retrieving the spheres of activity to which a list of users belongs.

- Otherwise (*context filter*), it reads the user's focus of attention (F), and calculates the intersection between F and the tagging information of the event (RG). If $F \cap RG \neq \emptyset$, or one of the reference groups in RG belongs to the *non filtered* spheres, then the Notification Manager sends the notification to the user; otherwise, it defers it.

If the user's is online but the focus of attention is empty, this means that the user is working outside the ICE. Thus, the Notification Manager defers all the notifications, except for those concerning the *non filtered* spheres.

5 Related Work

The notification management approach presented in this paper is based on the mediated notification management model, which is largely used and has been identified as one of the best performing methods; see [6]. In particular, our approach extends previous work on priority-based notification with the management of parallel notification contexts, representing different priorities for the user. Different from the work in [10], where the notifications are filtered on the basis of their features (e.g., the sender, priority of a message, etc.), we base the management of notifications on the sphere of activity to which they belong. Specifically:

- We introduce a *context filter* policy, which delays the delivery of the notifications belonging to spheres of activity out of the user's focus of attention. This policy implements a context-dependent notion of priority, suitable for the environments supporting the management of parallel activity spheres and multiple collaborations.
- We introduce a context-dependent model for the presentation of the notification to the user, in order to support the inspection of the awareness information concerning the various spheres of activity she is involved in.

This differs from the awareness support offered by e-collaboration environments such as BSCW [16], or MyWebDesktop [17], which support the management of parallel collaboration groups, but only filter the notifications on a subscription basis.

Our work strictly relates with the ecology of collaborations proposed by Mark and Su in [1]. However, our activity spheres are more general than the *working spheres* introduced in [1] and can be used to represent any kind of collaboration the user is involved in: both work and personal ones. Moreover, while Mark and Su focus on whether the user can be interrupted, depending on her working sphere, our work aims at steering the notification management.

6 Conclusion

We have described an Integrated Collaboration Environment (ICE) supporting e-collaboration within multiple spheres of activity; e.g., different projects users are involved in at work, their social activities, and so on. Specifically, we have focused on the notification management issue and we have defined a novel notification policy supporting the context-dependent delivery of messages to the user. We based the definition of

our notification policies on the results of a user test carried out within our lab; the test evaluated the *context filter*, which defers notifications on the basis of their relevance to the user's focus of attention, as the policy most suited to the user's needs.

The next step of our work is the evaluation of our ICE prototype with users. In particular, we will focus on the proposed notification management policies, in order to evaluate the impact of the ambiguity in the identification of the focus of attention on the selection of the notifications to be deferred.

Currently, we do not analyze the user's activities in detail, e.g., to identify different phases in the execution of a task; e.g., see [15]. This analysis might be part of our future work, in order to investigate the adaptation of notification deferrals to the user's attention level. In our future work, we will also deal with privacy issues; see [18–20].

References

1. Mark, G.: Considering Web 2.0 technologies within an ecology of collaborations. In: Proc. of SociUM: Adaptation and Personalisation in Social Systems: Groups, Teams, Communities, Corfu, Greece (2007) 50–59
2. Grimes, A., Brush, A.: Life scheduling to support multiple social roles. In: Proc. CHI 2008, Florence, Italy (2008) 821–824
3. Collanos: Products overview: Team enabling professionals, <http://www.collanos.com/en/products> (2008)
4. ActiveCollab: Project management & collaboration, <http://www.activecollab.com/> (2008)
5. Scupelli, P., Kiesler, S., Fussell, S., Chen, C.: Project View IM: a tool for juggling multiple projects and teams. In: Proc. CHI'2005, Portland, Oregon (2005) 1773–1776
6. McFarlane, D.: Comparison of four primary methods for coordinating the interruption of people in Human-Computer Interaction. *Human-Computer Interaction* **17**(1) (2002) 63–139
7. Bailey, B., Konstan, J., Carlis, J.: The effects of interruptions on task performance, annoyance, and anxiety in the user interface. In: Proc. INTERACT'01, Tokyo (2001) 593–601
8. McFarlane, D., Latorella, K.: The scope and importance of human interruption in human-computer interaction design. *Human-Computer Interaction* **17**(1) (2002) 1–61
9. Czerwinski, M., Horvitz, E., Wilhite, S.: A diary study of task switching and interruptions. In: Proc. CHI 2004 Conference on Human Factors in Computing Systems, Vienna (2004)
10. Horvitz, E., Apacible, J., Subramani, M.: Balancing awareness and interruption: Investigation of notification deferral policies. In: Proc. 10th Int. Conf. on User Modeling, Edinburgh, UK (2005) 433–437
11. Dourish, P., Bellotti, V.: Awareness and coordination in shared workspaces. In: Proc. CSCW '92, Toronto, Canada (1992) 107–114
12. Brush, A., Barger, D., Grudin, J., Gupta, A.: Notification for shared annotation of digital documents. In: Proc. SIGCHI conference on Human factors in computing systems (CHI '02), Minneapolis, USA (2002) 89–96
13. 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*. North-Holland, Amsterdam (1988) 239–250
14. Ardissono, L., Goy, A., Petrone, G., Segnan, M.: SynCFr: Synchronization collaboration framework. In: Proc. of 5th Conference on Internet and Web Applications and Services (ICIW 2009), Venezia, Italy, IEEE (2009) to appear
15. Iqbal, S., Horvitz, E.: Disruption and recovery of computing tasks: field study, analysis, and directions. In: Proc. of CHI 2007, San Jose, CA (2007) 677–686

16. Horstmann, T., Beltley, R.: Distributed authoring on the Web with the BSCW shared workspace system. *StandardView* **5**(1) (1997) 9–16
17. MyWebDesktop.net: Your personal and collaborative desktop on the internet, address = <https://www.mywebdesktop.net/>, year =
18. W3C: Platform for Privacy Preferences (P3P) Project, <http://www.w3.org/P3P/> (2006)
19. Kobsa, A., Schreck, J.: Privacy through pseudonymity in user-adaptive systems. *ACM Transactions on Internet Technology* **3**(2) (2002) 149–183
20. Brar, A., Kay, J.: Privacy and security in ubiquitous personalized applications. In: Proc. 10th International Conference on User Modeling (UM'05) Workshop 9, Edinburgh (2005) 47–54