Feed-Oriented Awareness Services for eLogbook Mobile Users

Sandy El HELOU, Denis GILLET, Christophe SALZMANN and Yassin REKIK

École Polytechnique Fédérale de Lausanne (EPFL)

Key words: Awareness, Mobile Learning, Collaborative Learning, RSS Feeds

Abstract:

The eLogbook is a collaborative Web-based environment deployed by Swiss Federal Institute of Technology in Lausanne (EPFL). It consists of an activity-oriented space where a community of educators, teaching assistants and students can form groups, conduct activities and perform several actions over stored and shared assets. Moreover, the eLogbook offers awareness services in order to sustain collaboration and coordination among the community members. This paper presents adaptive feed-oriented services designed and implemented to deliver awareness to mobile users, taking into account their device limitations and their activity context.

1 Introduction

Dourish and Belloti [1] define awareness as “an understanding of the activities of others, which provides a context for one’s own activity”. This definition highlights the need for providing awareness services in Computer Supported Collaborative Work (CSCW). However, many studies have shown that excessive unnecessary notifications might lead to adverse effects such as a decrease in productivity [2,3]. In order to avoid similar negative awareness repercussions in CSCW, three fundamental issues are to be addressed when it comes to awareness support [4]: What information should be delivered? How should it be presented? And when is it most useful to deliver it? Moreover, as mobile devices are tremendously spreading, addressing the issues mentioned above must take into consideration the limitations of those devices and the need to adapt to them. In fact, mobile devices are characterized by their paucity of resources; they have limited bandwidth and storage capabilities, short battery life, unreliable network connectivity, and often, small display screens [5]. In an attempt to address the issues raised above taking into account the mobile devices intrinsic constraints, the eLogbook activity-oriented collaboration space proposes feed-oriented personalized awareness services particularly adapted to mobile users, in terms of the delivery and display means, as well as the quantity and quality of the received notifications. This paper first presents the eLogbook main functionalities and the types of awareness services it offers. Then, the two aspects of the personalized feed-oriented awareness approach are discussed, the first aspect being the adoption of RSS as a delivery means and the second aspect being the use of notification filtering techniques in order to personalize alerts and make them context-adaptive.
2 Description of the eLogbook

The eLogbook is a general-purpose activity-oriented collaboration space that can be customized by users to serve as an asset management system, as a task management system and/or as a discussion platform. The eLogbook is a collaborative learning environment developed within the Swiss Federal Institute of Technology in Lausanne (EPFL) in the framework of a European research project to support mediation in communities of practice. The eLogbook general features and underlying principles are described in [6].

2.1 The eLogbook 3A Model

The eLogbook 3A model integrates three main conceptual entities (Fig. 1), the actors, the activities and the assets. An actor is any entity capable of initiating an event within the eLogbook workspace. An asset is any kind of resource (e.g. text documents, images, graphs, simulation snapshots) shared between community actors. Actions initiated by actors can be either organizational or operational. Organizational actions are related to structuring the activities of the community through defining common objectives, managing the associated roles and scheduling the related deliverables. Operational actions englobe all other kinds of non-organizational collaborative actions such the manipulation of shared assets or the submission of activity deliverables. Moreover, the interactions between the three main entities are governed by Protocols. Protocols and actions that can be performed in the eLogbook are briefly examined below.

![Fig 1. The eLogbook model.](image)

Two major rules are to be checked whenever an actor wants to perform a particular action. First, the user must have been granted access over the entity (or the entities) involved. Second, the access right granted must allow him to perform the action in question. If the entity involved is an activity, then it is the actor’s role in this activity, which defines his access rights.

It is useful to mention that granting access over an activity can be done either statically or dynamically. A static assignment is straightforward and unconditional; it is equivalent to saying, Actor X is invited to join Activity X with Role Y. However, a dynamic assignment can be perceived as a statute-dependent access, it is equivalent to saying all members of Activity Y are invited to join Activity X or all members of Activity Z with Role W are invited to join Activity X. The first case means that actors will have access to Activity Z only as long as they...
remain members of Activity Z, and in the second case, the actor’s granted access is also conditioned by one more constraint, which is that he maintains Role W in activity Z. Similarly, asset access rules can either be granted statically (on a personal basis) or dynamically (on an activity or role basis).

The table below lists all possible events that can be performed within the eLogbook along with the entities involved.

<table>
<thead>
<tr>
<th>Type of Event</th>
<th>Entity Involved</th>
<th>Entity Type</th>
<th>Event Descriptive Notes</th>
</tr>
</thead>
</table>
| New or Update Activity Invitation                  | Invited entity                   | Actor, Activity and/or Role                      | - An invitation to join a newly created activity or an already existing one. By default the status of a user for a new invitation is pending.  
- The invitation status can be updated by the invited member to accept or reject  
- The admin(s) of the activity can cancel the invitation by setting the status to forbidden |
|                                                   | Inviting or Initiator entity     | Actor                                            |                                                                                                                                                    |
|                                                   | Destination of the invitation    | Activity                                         |                                                                                                                                                    |
|                                                   | Indirectly Involved Entities     | Other Members of this activity                   |                                                                                                                                                    |
| New Asset Access                                   | Target Entity                    | Actor, Activity and/or Role                      | Access rights granted can be read-only or full rights.                                                                                               |
|                                                   | Inviting or Initiator entity     | Actor                                            |                                                                                                                                                    |
|                                                   | Accessed Entity                  | Asset                                            |                                                                                                                                                    |
|                                                   | Indirectly Involved Entities     | Other Actors who have access rights over this assets |                                                                                                                                                    |
| Create, Read, Update, Delete role/deliverable     | Initiator                        | Actor                                            | - Updating a role can consist of changing the rights of this role.  
- Updating a deliverable can consist for example of changing its submission deadline or its order of submission.                                    |
|                                                   | Created Entity                   | Role/Deliverable                                 |                                                                                                                                                    |
|                                                   | Indirectly Involved Entities     | Activity (to which the role or deliverable belongs)  
Actors (having this role/expecting to submit this deliverable) |                                                                                                                                                    |
| Semantic Links                                     | Initiator                        | Actor                                            | There are predefined semantic links such as subactivity for activities, comment, complements for assets, succeeds for deliverables                      |
|                                                   | Linked Entities                  | Activities or Actors or Deliverables or Roles or Asset |                                                                                                                                                    |
| Rates or Tags                                      | Initiator                        | Actor                                            |                                                                                                                                                    |
|                                                   | Tagged or Rated Entity           | Activity or Actor or Deliverable or Role or Asset |                                                                                                                                                    |
| Read, Update, Delete Asset/Activity                | Initiator                        | Actor                                            |                                                                                                                                                    |
|                                                   | Read, Updated or deleted entities| Asset/Activity                                   |                                                                                                                                                    |
|                                                   | Indirectly Involved Entities     | Activities, actors and or roles (members of the activity/owners of the asset) |                                                                                                                                                    |
| Asset Submission                                   | Initiator                        | Actor                                            |                                                                                                                                                    |
|                                                   | Represented Entities             | Actor(s)                                         |                                                                                                                                                    |
|                                                   | Controlling entities             | Actor(s)                                         |                                                                                                                                                    |
|                                                   | Destinated entity                | Deliverable (for which the asset was submitted)  |                                                                                                                                                    |
|                                                   | Indirectly involved entity       | Activity (to which the deliverable belongs)       |                                                                                                                                                    |

Table 1: Listing of the eLogbook events
2.2 Types of eLogbook Awareness Services

The eLogbook provides each and every user three main interconnected types of awareness:

*Artifact-Based Awareness* includes informing each and every member about all actions that revolve around the artifacts or the assets that are being exchanged within the community and within the groups to which the entity belongs. This includes informing members about the creation of new assets, new links and tags over these assets as well as the evaluation, or submission of assets. It also includes statistics about how many members have access over a specific asset and how many have read it, tagged it, and rated it.

*Actor-Based Awareness* includes informing each and every member of the state of other members that are involved with him in activities or sub-activities (connected, absent, busy). It also includes notifications of all tags, links, rates related to the user himself and to other users with whom he shares assets and tasks.

*Activity or Task-Based Awareness* includes notifying members of all changes occurring within the activities he or she is has been invited to. Creating new deliverables and/or roles, updating or deleting existing ones, inviting new members to join the activity, are examples of such activity-related events. This type of task also includes notifications such as reminding members of their dues when the task deadline is close.

3 The personalized feed-oriented awareness approach

This section examines in details the motivations behind the two complementary aspects of the eLogbook awareness approach that targets mobile users in particular. The use of RSS as the awareness delivery means and the rationale behind this choice is first examined. Then the awareness filtering mechanisms and their contributions in making awareness services more adaptive to mobile users are discussed. Finally, the overall awareness process is summarized.

3.1 Using RSS as the mobile awareness delivery means

RSS (Real Simple Syndication or Rich Site Summary) consists of an XML application based on a combination of push and pull technology. RSS feeds are made of channels, each of which is composed of a title, a link and a description. A channel may also include other items such as an image and/or a rating. In addition, it can have any number of elements, with each containing at least a title or a description. By subscribing to RSS feeds of interest, the user can check, on a single screen, all relevant news.

The eLogbook sends notifications through RSS feeds. This approach presents several advantages, in particular for mobile users. In fact, it allows the delivery of information in an unobtrusive way. This is achieved by transmitting information via a familiar interface where users usually expect and intentionally check for updates. Moreover, the format of RSS feeds is particularly useful for mobile users subjected to device constraints. News is sent in a compact way and optional fields (such as images) are skipped for mobile users. In addition, the information sent via RSS is filtered in terms of relevancy in order to make it personalized and adapted to each receiver’s interest and needs (the filtering techniques used are discussed in the next section). Thus, even though members register for the same eLogbook RSS feed, the notifications sent over RSS channels automatically differ from one user to another.
3.2 eLogbook Awareness Filtering Techniques

The awareness mechanism takes as input all the events that occurred with the eLogbook, filters them based on each user’s own notification preferences and on the system default filtering rules which are user and context dependent. The output produced consists of a set of relevant and personalized notifications for each user. The notification filtering exploits the actions performed in the eLogbook workspace to decide on the usefulness and relevancy of a specific event, for a particular user in a particular context.

Awareness mechanisms are essential in cooperative and collaborative environments where people interact through task sharing and assets exchange [7]. Nevertheless, as it was previously mentioned, in the case where notifications are not context-adaptive, relevant and personalized, the awareness services end up having an adverse effect. For example, in a learning community made of educators, teaching assistants and a considerable number of students, if everytime a student comments on a document or an educator decides to change the rights assigned for a role in a specific activity, everyone is notified, an overflood of information will take place and community members will tend to ignore notifications, this at a time where some of them might be important and might require an immediate action [8]! The situation becomes more critical when students are receiving information on mobile devices that impose a bigger need for information selectiveness, especially because of the previously mentioned constraints of such devices, including unreliable communication and limited screen size. A notifications filtering mechanism that takes into account the target user context and interest, is the best way to avoid the notifications adverse by improving the quality of notifications.

Every action done within the eLogbook is registered not only in a chronological order but also and more importantly in a contextual fashion, every action is traced within the context in which it occurred. This functionality is mainly used by the Awareness services to keep track of all the events that are happening within the community. But it is worth mentioning that this functionality respects the privacy of each member and the privacy of each subgroup. In this sense, members are held “aware” of actions and events that only concern the activities in which they are involved and/or the assets, which they can access. This is what we refer to by policy-based natural filtering mechanism which produces a set of notifications that each user is allowed to receive based on predefined activity and asset access rules. If the user chooses to completely disable the automatic filtering techniques and does not specify any personal notification preferences, he/she will be notified of all the events belonging to this set of allowed notifications.

It is argued that from the user’s interaction with the collaborative environment, important information can be inferred regarding which events could be useful and interesting to him/her. Accordingly, a default system’s filtering algorithm is proposed. It consists of a set of generic rules directly related to each user’s interactions with his/her collaborative environment. The question of notification relevancy can be addressed by considering for every action performed, its type and the entities involved in it. The relative importance of an event for a specific target user varies as a function of his/her own context, his/her relation with the entities involved and the type of action performed. Based on these relations, a decision is taken regarding the usefulness of the event with respect to this user. This decision dictates whether or not to notify the user about the event.

The default filtering rules for all kinds of artifact-related, activity-related or actor-related events mentioned in Table 1, are described thereafter. These rules are valid in mobile context where RSS notification channels are chosen as delivery means. In such a context, RSS feeds
are mainly used for notification of urgent actions, which are expected by the target user or which will probably influence his/her immediate or the future course of actions. For example, inviting a user to join an existing activity is an event that frequently happens within the eLogbook. A student accessing the eLogbook rss feeds from a mobile device, in the intention of checking whether a teaching assistant or an educator has posted a new deliverable or validated a new document, would not be interested in knowing that another new student has joined the virtual community. It is important to mention that this does not mean that the information is hidden from the user, because at any time, the eLogbook can be browsed and all the events.

The default filtering rules governing asset-related events are the following:

**Granted Asset Access to target user:** A user will be notified via RSS that he/she has been granted an access right over an asset has been granted if and only if at least one the following conditions is met:

- The target user shares other assets with at least one author of the asset in question and did show interest in those shared assets (by performing actions over them such as reading, tagging, linking and/or rating them.)
- The average rate of the asset higher or equal to an average rating value.
- At least one of the tags of the asset in question matches the user’s topics of interest set in his/her profile.
- The target user had rated at least one author of the asset high.

**Granted Asset Access to another user:** A user will be notified of such an event if and only if he/she is an author of the asset in question and the right granted to the other user is an authorship right.

**Other Asset-related Actions:** A user will be notified of statistics related to an asset and actions such as linking, tagging, deleting, rating, updating existing assets if and only if at least one of the following conditions holds:

- The average rate of the asset is high
- The target user is an author of the asset
- The target user had shown interest in the asset in question, either directly by giving it a high rate, or indirectly by reading it, annotating it or linking it to other assets he had shown interest in.

The default filtering rules governing activity-based events are the following:

**Management of Roles:** A user is held aware of actions revolving around roles (such as giving more rights to a role, deleting a role) if and only if:

- It is his own role
- It is the role of another actor whom the user collaborates with and updating the role of this other actor will influence the current state-of-art collaboration between those two actors. (e.g., the right to access a resource which was granted on a role-basis, was changed from complete access to read-only)
- The target user has administrative right in the activity to which the changed role belongs.

**Management of Deliverables, Submission and Validations:** A user will only be notified of actions related to deliverable management such as the creation of new deliverables, changes
in submission or validation deadlines, ordering, and deletion if only if the user’s role gives him/her the right to submit or evaluate these deliverables. That is:

- All users expected to submit a deliverable will be reminded of the submission deadline if they didn’t yet submit the expected work for a specific deliverable and the submission deadline is close.
- Evaluators will be held aware of all submissions, which they are expected to evaluate. They will also be provided with the URL of the page where they can evaluate the submissions.
- Evaluators are also reminded of validation deadlines when the latter are close and the expected evaluations are not done yet.

Invitation of the target user to join an activity: target user will be notified of such an activity if and only if at least one of the following conditions holds:

- The initiator of the invitation is an actor with whom the target user usually collaborates.
- The average rate of the activity is higher or equal to an average rating value.

Other Activity-related actions: A user is made aware of actions such as rating, tagging and linking activities only if one of the following conditions holds:

- If the target user gave a high rate for the activity in question.
- If the target user has chosen to join this activity and was active inside it through performing actions over it and/or over its resources.

The default filtering rules governing actor-based events are the following:

A user will only be notified of the status (online, offline, busy) of users whom he/she frequently collaborates with, through active manipulation of shared resources. This feature is useful for students working with each other in subgroups and/or supported by teaching assistants, this especially when the user is waiting for the availability of another actor to trigger a real-time synchronous discussion (to ask the teaching assistant questions or to discuss a specific task with another colleague) or a shared experiment.

The eLogbook gives every user, the possibility to specify personal filtering rules by completely or partially disabling the system filtering mechanism, changing some of its default rules and/or define new notification rules. It is also conceived, to make the user’s defined rules public, and if so, another user can simply choose to import another user’s notification rules. Before discussing how a user can add his/her own rules, it is important to introduce the notion of smart lists. A smart list can be defined as a dynamic list of conditions over attributes corresponding to a specific class or entity. For example, a user can define a smart list of activities by specifying filtering conditions on them such selecting only activities where his status is joined and his role gives him the right to evaluate deliverables. Consequently, a user can express his/her interest in a specific event or a set of events or all events, by defining smart lists for the entities involved and specifying a logic combinations of those smart lists. This can be best understood by examining the three following examples:

- A simple user-defined notification rule could be as follows: Notify me of all events related to Activities in my smart list “my favorite activities“.
- A slightly more complex user-defined notification rule would introduce a logic combination of conditions on two or more separate entities smart lists. For example, the user might wish to be notified for all events related to Activities in a specific smart list “my favorite activities“ and where the initiator actor belongs to the smart list “my favorite actors“. 
• Finally, in some cases, it might be useful to define different smart lists or logic combinations of smart lists for every type of event or for a set of events.
• For example, a user might wish to be notified of only the two actions of updating and/or linking an asset only when the initiator actors belongs to his smart list of “important authors“ AND the asset involved belongs to his smart list of assets “assets important to track“.

### 3.3 Overall Process Summary

Every time an action occurs, the filtering mechanism is initiated for every user. First of all, the entities involved in the event and the type of the event are identified. Second, the notification preferences and the automatic filtering settings are checked for each user. Third, natural filtering is initiated; basically, the awareness mechanism will filter out every event, in the case where the target user is not allowed to access at least one of the entities involved in the event. If the automatic settings are completely disabled and there are no user notification preferences, the only filtering that will be done is the natural filtering mentioned above. Otherwise, the notification rules (default system rules, modified system rules or user defined rules) are used to filter information. Finally, each notification of the resulting filtering set will correspond to a new element within the eLogbook RSS channel accessed by the user through RSS aggregators. Figure 2 shows the emulated user interface of the feed-oriented awareness services on a Blackberry® device.

![Figure 2: Emulated user interface of the feed-oriented awareness services on a Blackberry® device.](image)

### 4 Related Work

A lot of research has been done around the field of awareness and notification mechanism in CSCW. Among them, some were mainly focused on the evaluation of interruption strategies [9], while others have tried to compare the notification benefit to the cost of interruption
Other approaches such as the FeedMe approach filters notifications using Bayesian statistical filtering techniques, which rely on a user’s rating of previous notification to decide whether or not a notification is useful to him/her [12]. Our approach differs in the sense that notification filtering relies on the user’s interaction with his workspace to decide on the relevancy or usefulness of an event with respect to a particular user in a particular context. Moreover, eLogbook relies on information filtering techniques, as a way to improve the quality of alerts, and on the use of RSS channels, as convenient feed-oriented delivery means, in order to better adapt to mobile contexts, which was not the purpose for the FeedMe approach. AwareDaV presented a framework to deliver awareness, through XML-based files for people sharing resources on Web Servers [13]. The possibility to subscribe to different nodes and receive notifications related to those nodes only. The eLogbook awareness mechanism differs in two ways. First, the user can rely on an automatic filtering system which does not require any effort from his side to state what is useful for him or not, because this can be inferred from his/her interaction with the collaborative workspace. Second, the user also has the possibility to define his/her own rules according to his/her own notification preferences. This reinforces the control of which information the user wishes to receive through defining dynamic rules rather than subscribing to every single interesting event.

5 Conclusion

Delivering notifications is a major concern in CSCW. However notifications might have adverse effects on the productivity of people working in shared workspaces due to excessive interruptions and unnecessary alerts, many of which are not particularly useful for a specific user in a specific context. Therefore, dynamically adapting the notification delivery system to the target user taking into account the user context (in particular the target device) becomes an important need. Filtering rules were used in eLogbook to achieve dynamic adaption of both the quantity and quality of the alerts to a particular user in a particular context. Due to its adaptivity and its adequation with the eLogbook needs, RSS feeds were chosen as delivery and display means to mobile user.

References:


**Author(s):**

Sandy El Helou,  
École Polytechnique Fédérale de Lausanne (EPFL)  
CH- 1015 Lausanne, Switzerland  
Sandy.elhelou@epfl.ch

Denis Gillet,  
École Polytechnique Fédérale de Lausanne (EPFL)  
CH- 1015 Lausanne, Switzerland  
Denis.gillet@epfl.ch

Christophe Salzmman,  
École Polytechnique Fédérale de Lausanne (EPFL)  
CH- 1015 Lausanne, Switzerland  
Christophe.salzmman @epfl.ch

Yassin Rekik,  
École Polytechnique Fédérale de Lausanne (EPFL)  
CH- 1015 Lausanne, Switzerland  
Yassin.rekik@epfl.ch