

# Tackling Acceptability Issues in Communities of Practice by Providing a Lightweight Email-based Interface to eLogbook: a Web 2.0 Collaborative Activity and Asset Management System

Denis Gillet<sup>1</sup>, Chiu Man Yu<sup>1</sup>, Sandy El Helou<sup>1</sup>, Amagoia Madina Berastegui<sup>1</sup>,  
Christophe Salzmänn<sup>1</sup>, and Yassin Rekik<sup>1</sup>

<sup>1</sup> École Polytechnique Fédérale de Lausanne (EPFL), CH-1015 Lausanne, Switzerland.  
{denis.gillet, chiuman.yu, sandy.elhelou, amagoia.madina, christophe.salzmänn,  
yassin.rekik}@epfl.ch

**Abstract.** eLogbook is a Web-based collaborative environment designed for communities of practice. It enables users to manage joint activities, share related assets and get contextual awareness. In addition to the original Web-based access, an email-based eLogbook interface is under development. The purpose of this lightweight interface is twofold. First, it eases eLogbook access when using smart phones or PDA. Second, it eases eLogbook acceptance for community members hesitating to learn an additional Web environment. Thanks to the proposed interface, members of a community can benefit from the ease of use of an email client combined with the power of an activity and asset management system without burden. The Web-based eLogbook access can be kept for supporting further community evolutions, when participation becomes more regular and activities become more complex. This paper presents the motivation, the design and the incentives of the email-based eLogbook interface.

**Keywords:** Community of practice, email-based system, activity management.

## 1 Introduction

eLogbook [8] is a Web-Based collaborative environment particularly adapted to the needs of communities of practice (CoPs). It is developed at the Swiss Federal Institute of Technology in Lausanne (EPFL). It relies on three fundamental entities: *Actors*, *Activities* and *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 and images) shared between community actors. An activity is the formalization of a common objective to be achieved by a group of actors. *Events* related to these three entities are governed by *Protocols*. eLogbook supports management of invitations, roles, and deliverables for the activities; and supports access rights management for the assets. It provides personalized and context-sensitive awareness information crucial in collaborative environments [4,5]. The features of eLogbook are useful to any kinds of CoPs.

The original interface of eLogbook environment relies on Web 2.0 technologies for enabling effective Web-based user interaction. It can be considered as a flexible and adaptive Web-based collaborative activity and asset management system or service that could easily be adopted by communities of practices.

As pointed out by Moor in [7] and confirmed in the framework of the Palette project (<http://palette.ercim.org>) through a participatory design approach [2], it is difficult for a community to select and adopt new environments and services for enhancing their practice without inducing disturbances. For the communities which have been using email as the main communication tool, email-based interface environment may be more acceptable than Web-based one. It is desirable for enabling communities to choose their suitable environments. Therefore, there is a need for solutions that facilitate the introduction of advanced collaboration services in CoPs. Our assertion is that, by providing an email-based interface to eLogbook, its further adoption can be strengthened and further evolutions of CoPs can be better supported. To validate this assertion, an email-based interface is designed as described in the next section. Its acceptability will then be investigated in the framework of the Palette project with pilot CoPs.

The email-based eLogbook interface enables users to manage their activities, assets, and awareness by sending emails directly to eLogbook and to receive requested information. Additionally, users can trace occurring events by receiving automatic notification emails from eLogbook. The advantages of providing an email-based interface to eLogbook are the following:

- To use the email-based interface, the users' devices only need to have email client installed. It is very common for computers, pocket PCs and even smart-phones to integrate built-in email clients.
- The communication cost induced by using an email-based interface is cheaper than the Web-based one. This is still an important factor for mobile users.
- The users can store emails on their devices. Afterwards, they can manage joint activities, share related assets and get contextual awareness without connecting to the Internet. The mail-based eLogbook interface provides offline information management and relies on standard email synchronization solutions.
- The users can send and check emails at a time of their choosing.
- As pull scheme, email access is intrinsically context-oriented.

The rest of this paper is structured as follows. Section 2 describes work related to email-based collaborative environments. Section 3 introduces the specifications of the email-based eLogbook interface. Section 4 ends with concluding remarks.

## **2 Related Work**

Email systems are widely deployed in workflow, task and activity management systems [3,6,9,10,12]. Workflow systems specify and monitor evolution of business processes. Email usually does not require real-time interactions. It is a convenience means for offline communication between users and systems. *Lynx* [11] is an email extension for workflow systems based on Web Services. It provides a Web service

through which a workflow application can interact with human partners via an email-based forms interface without requiring a specialized client. Its server side is composed of a BPEL execution engine, an outgoing email Web service and other partner Web services, as well as an incoming email gateway. The client side is composed of a standard email client application and an XForms player component. *Taskmaster* [1] is an email-based task management system. It uses an email system that can embed task management information directly in the email inbox. The information includes warning bars (which show task deadlines), action clusters, and task-specific contact lists. This solution enables management of emails and tasks in single application.

EDI (Electronic Data Interchange) systems are used in commercial organizations for trading partners to exchange information with each other. The systems have been using email as a means of exchanging messages. Each trading partner needs to provide an email address for EDI messages, and an email address for personal communications related to EDI. Typically, the MIME encapsulation specification would be used to enclose the EDI data within the email message. The trading partners would need to agree upon an encryption method for secure email.

### 3 Email-Based Interface to eLogbook

The email-based interface to eLogbook allows invocation and information requests via email. The users are authenticated by their registered email address that is hence required to send requests. The requests must follow a number of predefined rules to be interpreted by eLogbook properly.

First, this section presents the system flow for the email-based interface to eLogbook. Second, it states the syntax of the email requests. Third, it describes how notification mechanisms can be controlled via email and how particular information can be requested. Finally, it presents how two kinds of eLogbook actions can be triggered via email.

#### 3.1 System Flow for the Email-Based Interface to eLogbook

Fig. 1 shows the system flow for the email-based interface to eLogbook. The email address for accessing eLogbook is *action.elogbook@epfl.ch*.

Every time eLogbook receives an email from a user, it handles the request according to the following steps:

- Step 1: *Sender Identification*. In this step, a check on whether the email sender is indeed a registered eLogbook user is performed. If this is the case, step 2 is initiated. Otherwise, the mail is ignored.
- Step 2: *Email Dissemination & Action Identification*. The content of the mail is parsed (subject and body), and the action to be performed is identified. In cases of ambiguous requests, an error message is sent back to the user.
- Step 3: *Protocol Checking*. A check is performed in this step to make sure that the sender is allowed to perform the requested action based on the access

rights s-he has been granted over the entities involved. For example, if the user wishes to create a sub-activity of another already existing activity, s-he must have administrative rights over the latter.

- Step 4: *Confirmation Request*. If the sender is allowed to perform the requested action, then an email is sent back for requesting a confirmation. This step is important for two reasons. First, it is used for security purpose in order to make sure that the corresponding eLogbook user was indeed the one who actually sent the request. Second, it ensures that the user indeed wishes to perform the action based on what s-he had sent and how it was interpreted by eLogbook. Each confirmation request has an expiration time, e.g., 24 hours from the sending of the confirmation request.
- Step 5: *Action Execution*. If the user confirms by replying to the email from eLogbook, then the requested action would be executed.

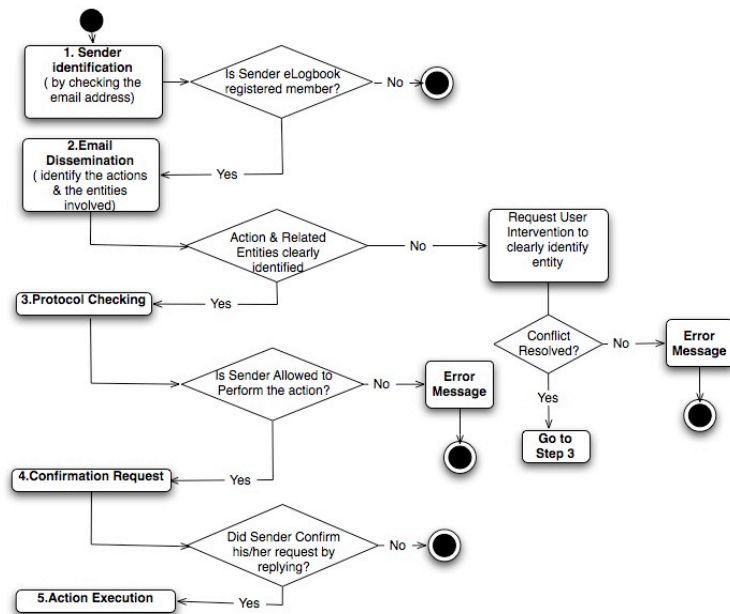


Fig. 1. System flow for the email-based interface to eLogbook.

### 3.2 Syntax of Emails

The lightweight email-based interface to eLogbook allows users to write email in simple syntax to request for actions or information. Figure 2 shows the syntax of the request email from a user, and that of the confirmation email from eLogbook.

A request email contains four parts. The *email recipient* states the email address of the eLogbook server. The *email sender* states the email address of the requester. The

*email subject* states the type of requested action, e.g., “create new activity”. The *email attachment* is used for attaching file for “create new asset” action. The *email body* states the information related to the requested action. It is formatted in plain text such that any actor can compose its request email using simple text editor. In addition, plain text format is compatible with all email clients.

```
Email recipient: <eLogbook email address>
Email sender: <actor email address>
Email subject: <type of action>
Email attachment: <File>
Email body:
'Name: <string>'
'Description: <string>'
'Public tags: <strings>'
'Private tags: <strings>'
'Public role: <string>'
'Relation: <type of relation>'
'Destination: <name of activity or user>'
```

Fig. 2. Syntax of a request email.

The *command tags* used in the email are in the format '*<command string>*'. Currently there are seven command tags for defining names, descriptions, tags, roles and relations of activity/asset/deliverable. Among them, only the *Name* field is compulsory; other fields are optional.

```
Email recipient: <actor email address>
Email sender: <eLogbook email address>
Email subject: Confirmation of the <type of action> action
Email body:
<information of the requesting action>
'Reference: <reference number>'
'Access link: <URL>'
```

Fig. 3. Syntax of a confirmation email.

The confirmation email is sent by the eLogbook server to the sender in order to authenticate her/his identities (Fig. 3). The actor needs to reply to the confirmation email to invoke the action stated in the email subject. The detailed action information is also listed in the email body. The confirmation email contains a unique *reference number* for each particular action request. The eLogbook server also generates and stores a hash value of the email for integrity check. The email contains an access link for the actors to access their new activity/asset/deliverable through the Web interface after confirmation or at a later stage if s-he deems it appropriate.

### 3.3 Requesting Information and Tuning Default Notification Mechanism

By default, an eLogbook user can be notified of several events via email, e.g., the reminder of the submission or validation of deadlines, and the invitation to join a new activity. The user can completely or partially disable the default eLogbook notification mechanism through the Web-based or the email-based eLogbook

interface. In the latter case, the email subject and body are interpreted in order to invoke the user's requested action.

- If the email subject is set to “disable all notifications”, then the user will stop receiving all sorts of eLogbook notifications. An email with the subject “enable all notifications” induces the opposite action.
- If the email subject is set to “disable notifications”, then the body of the mail contains the name of activities, assets and actors, in the format illustrated below. The requested action is to disable all notifications related to one of the activities, assets and actors listed in the mail.
- At any point in time, the user can “ping” eLogbook to get information related to a specific actor, activity or asset by setting the subject of the email in the form: “get info <entity\_type>: <entity\_name>” where the *entity\_type* can be activity, actor or asset, and the *entity\_name* can consist of the name of an activity, an actor or an asset. This feature is mostly suitable for people who prefer “pull” rather than “push” notifications means. eLogbook responds by sending to the user a report of the executed actions related to the stated entity since the last time the user visited the eLogbook site or requested related notifications via email.

### 3.4 Triggering Operational and Administrative Actions

eLogbook users can trigger two kinds of actions. First, *organizational actions* are related to administrating and structuring the activities of the community by defining common objectives, scheduling deliverables and managing the roles assigned to the community members. Second, *operational actions* enclose all other kind of non-organizational collaborative actions such as posting an asset in an activity, linking, tagging and/rating an actor, an activity or an asset.

By simply sending an email to eLogbook ([action.elogbook@epfl.ch](mailto:action.elogbook@epfl.ch)), members can invoke operational as well as organizational actions. Examples on how activities, assets and deliverables are created are given below in order to illustrate the interactions between eLogbook and users via email.

#### 3.4.1 Creating an activity

Setting the subject of the email to “create new activity” creates an activity with its default administrator being the sender of the email. The body of the mail should contain the *name of the activity*, optionally the activity description and the following fields:

- *Public/Private tags*: Public tags will be shown to all the people who can see the activity; private or personal ones are only visible to their creators.
- *A public role*: If the email sender sets this field to “yes”, then the activity is made public with default rights granted to everyone (allowing them to perform all non-organizational actions). The administrator can decide not to rely on default rights but define a new set of rights. This can be done via email as well. If the email sender sets the field to “no” or does not mention it at all, then

the activity is kept secret except for members who will be explicitly invited by the activity initiator.

- A relation to another already existing activity: The sender can define a relation between this activity and an already existing one. The most frequently used unidirectional relation type is “sub-activity of”.

Fig. 4 shows the body of an email example for creating an activity. The actor “amadina” wants to create an activity called “TelCoP07” which is a sub-activity of “Palette”. The email states two public tags, “email” and “usability”, for the activity. After the actor sends the email and before performing the requested action, eLogbook sends a confirmation email to the actor “amadina”, as shown on the Fig. 5. Once confirmed, eLogbook runs the action requested, and the new sub-activity called “TelCoP07” for the existing activity “Palette” will be created. Fig. 6 shows the new activity in the Web-based eLogbook interface.

```
'Name: TelCop07'  
'Description: Workshop TelCoP07'  
'Public tags: email, usability'  
'Public role:yes'  
'Relation:"sub-activity of" "Palette"'
```

Fig. 4. Body of the email for creating a new activity.

```
Confirm the following information:  
  
'Name: TelCop07'  
'Description: Workshop TelCoP07'  
'Public tags: email, usability'  
'Public role:yes'  
'Relation:"sub-activity of" "Palette"'  
  
'Reference: 12'  
Access link: http://elogbook.epfl.ch/context/actor/42  
  
Please, confirm this action by REPLYING to this email. Otherwise, the action will be ignored.
```

Fig. 5. Body of the confirmation email.



Fig. 6. Context view for the new activity “TelCoP07”.

### 3.4.2 Creating an Asset

Setting the subject of the email to “create new asset” creates an asset with its owner being the sender of the email. In the body of the email should contain the *name of the asset*, and optionally the asset description and the following fields:

- *Public/Private tags*: Public tags will be shown to all the people who can see the asset; private or personal ones are only visible to their creator.
- A *public right*: If the email sender defines this field, then the asset is made public. The field can have three possible values: “owner”, “editor” or “reader”. If this field is skipped, the asset is kept secret except for members who will be explicitly granted access rights over the asset.
- A relation to another already existing asset: Any sort of predefined or user-defined semantic link can be used such as “reply to”, “complements”, “in favor”, “against”. For unidirectional links, the “of” preposition may be used.
- A *destination* field: If the user includes this field in the email body then access rights are granted to the stated entities which can be activities or particular actors. If an entity mentioned corresponds to an activity, then the asset is automatically posted in the activity. If it contains an actor’s username or email, then access right over the asset is granted to the actor.

Fig. 7 shows the body of an email for creating a new asset. The actor “amadina” wants to create an asset called “TelCoP07 paper v1” for the activity “TelCoP07”. The email states two private tags, “change chapter1” and “read conclusion”, for the asset. The asset is stored in the email as an attachment. After the actor sends the email and before the requested action is performed, eLogbook sends a confirmation email to the actor “amadina” (similar to the one we have seen in Fig. 5). Once confirmed, eLogbook runs the action requested, and an asset called “TelCoP07 paper v1” is created. Fig. 9 shows the created asset for the actor “amadina” in the eLogbook.

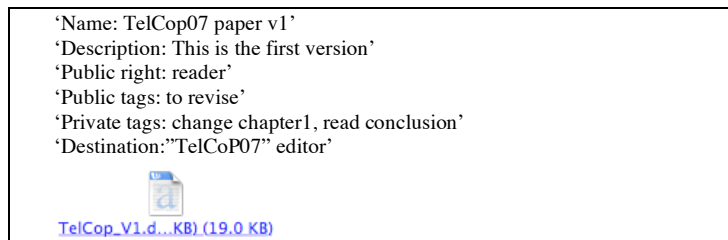


Fig. 7. Body of the email of creating a new asset.

### 3.4.3 Creating a Deliverable

Setting the subject of the email to “create new deliverable” creates a deliverable inside an activity. In the body of the email should contain the *name of the deliverable* as well as the *name of the activity* in which the former should be posted. Optionally the deliverable description and the following fields can be appended:

- *Public/Private tags*: Public tags will be shown to all the people who can see the deliverable; private and personal ones are only visible to their creator.



- The validation and submission deadlines.
- A relation to another already existing deliverable: It is mostly used to define the order of deliverables submission with an activity (using “precedes” or “exceeds”).

Fig. 8 shows the body of an email of creating new deliverable. The actor “amadina” wants to create a deliverable called “del07-07” for the activity “TelCoP07”. Fig. 9 shows the created deliverable in eLogbook after the email request.

'Name: del07-07' 'Activity name: TelCoP07' 'Description: This is the deliverable for the 7 July'
--

Fig. 8. Body of the email for creating a new deliverable.

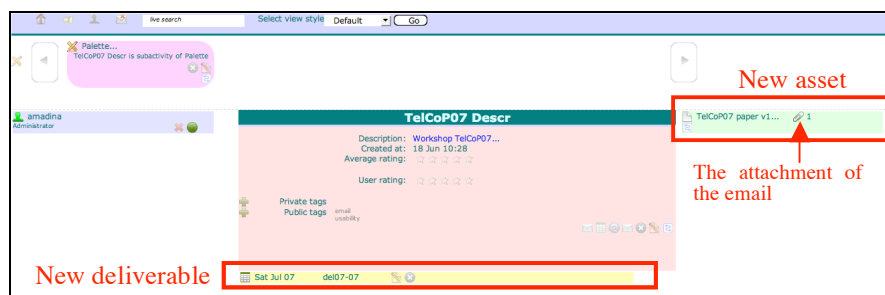


Fig. 9. Context view of the activity “TelCoP07”. A new deliverable is created.

## 4 Conclusions and Future Work

The email-based eLogbook interface enables users to manage their activities, assets and awareness through email. It provides an alternative lightweight non-Web interface to ease eLogbook access when using smart phones or PDA and to facilitate eLogbook acceptance for community members hesitating to learn an additional Web environment. It also has other advantages over Web-based access, such as low communication cost and offline information management.

The email-based eLogbook interface is under development. The user-friendliness, security and efficiency of the system will be further investigated. In addition, the comparative acceptability of the Web-based and email-based systems for communities of practice users will be assessed.

The email-based eLogbook interface not only provides a lightweight solution to the members of communities of practice, it can also be considered as a high-level interoperability mechanism for other services that can benefit from eLogbook features. For example, a semantic analysis service could request an asset stored in eLogbook via email, process it, generate semantic tags associated to the chosen asset, and finally send the semantic tags to the eLogbook via email.

From the view of users, the email-based eLogbook interface provides activity management, asset management, and awareness features. From the view of other

services, the email-based eLogbook interface provides workflow support for collaboration. Therefore, this interface integrates the features of legacy email-based systems to support both CoP users and services.

**Acknowledgments.** This work has been partially funded by the European Union through its Sixth RTD Framework Programme in Information Society Technologies (Palette Integrated Project).

## References

1. Victoria Belotti, Nicolas Ducheneaut, Mark Howard, Ian Smith: Taking Email to Task: the Design and Evaluation of a Task Management Centered Email Tool. Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, pp. 345-352, USA (2003).
2. Amaury Daele, Martin Erpicum, Liliane Esnault, Fabienne Pironet, Hervé Platteaux, Etienne Vandeput, Nathalie Wiele: An Example of Participatory Design Methodology in a Project which Aims at Developing Individual and Organisational Learning in Communities of Practice. Proceedings of the first European Conference on Technology Enhanced Learning (EC-TEL'06), Greece (2006).
3. Mark Dredze, Tessa Lau, Nicholas Kushmerick: Automatically Classifying Emails into Activities. Proceedings of the 11th International Conference on Intelligent User Interfaces, Australia (2006).
4. Denis Gillet, Sandy El Helou, Yassin Rekik, Christophe Salzmman: Context-Sensitive Awareness Services For Communities of Practice. To appear in the 12th International Conference on Human-Computer Interaction (HCI2007), Beijing, 22-27 July (2007).
5. Sandy El Helou, Denis Gillet, Christophe Salzmman, Yassin Rekik: Feed-Oriented Awareness Services for eLogbook Mobile Users. The 2nd International Conference on Interactive Mobile and Computer aided Learning (IMCL 2007), Jordan, April (2007).
6. Nicholas Kushmerick, Tessa Lau: Automated Email Activity Management: An Unsupervised Learning Approach. Proceedings of the 10th International Conference on Intelligent User Interfaces, pp. 67-74, USA (2005).
7. Aldo de Moor, Willem-Jan van den Heuvel: Web Service Selection in Virtual Communities. Proceeding of the 37th Annual Hawaii International Conference on System Sciences (HICSS'04), Big Island, Hawaii, Jan. 5-8 (2004).
8. Yassin Rekik, Denis Gillet, Sandy El Helou, Christophe Salzmman: The eLogBook Framework: Sustaining Interaction, Collaboration, and Learning in Laboratory-Oriented CoPs. The International Journal of Web-Based Learning and Teaching, Vol. 2, Issue 3 (2007).
9. Nelson Siu, Lee Iverson, Anthony Tang: Going with the Flow: Email Awareness and Task Management. Proceedings of the 2006 20th Anniversary Conference on Computer Supported Cooperative Work, pp. 441-450, Canada (2006).
10. Ross Tailby, Richard Dean, Ben Milner, Dan Smith: Email Classification for Automated Service Handling. Proceedings of the 2006 ACM Symposium on Applied Computing, pp. 1073-1077, France (2006).
11. Iván P. Vélez, Bienvenido Vélez: Lynx: An Open Architecture for Catalyzing the Deployment of Interactive Digital Government Workflow-Based Systems. Proceedings of the 2006 International Conference on Digital Government Research, pp. 309-318, California (2006).
12. Steve Whittaker, Victoria Bellotti, Jacek Gwydyka: Email in Personal Information Management. Communications of the ACM, Vol. 49, Issue 1, January 2006, pp. 68-73, ACM (2006).