Project no. FP6-028038

PALETTE

Pedagogically sustained Adaptive LEarning Through the exploitation of Tacit and Explicit knowledge

Instrument: Integrated Project

Thematic Priority: Technology-enhanced learning

D.MED.07 – Second version of e-Logbook (interoperable with CoPe_it! and integrating complete awareness views, contextualized feeds, as well as the e-mail interface)

Due date of deliverable: 31 January 2008
Actual submission date: 20 March 2008

Start date of project: 1 February 2006 Duration: 36 months

Organisation name of lead contractor for this deliverable: EPFL

Project co-funded by the European Commission within the Sixth Framework Programme		
Dissemination Level		
P	Public	**PU**

Keyword List: communities of practice, collaboration, awareness, collaborative learning, activity management, asset sharing, social software.
Responsible Partner: Denis Gillet, EPFL

MODIFICATION CONTROL			
Version	Date	Status	Modifications made by
1	19 February 2008	Draft	EPFL
2	26 February 2008	Draft	EPFL
3	29 February 2008	Draft	EPFL (version sent to evaluators)
4	07 March 2008	Draft	EPFL
5	10 March 2008	Final	EPFL (version sent to SC)

Deliverable manager

- Chiu Man Yu (EPFL)

List of Contributors

- Sandy El Helou (EPFL)
- Evgeny Bogdanov (EPFL)
- Amagoia Madina (EPFL)
- Manolis Tzagarakis (CTI)
- Chiu Man Yu (EPFL)
- Christophe Salzmann (EPFL)
- Denis Gillet (EPFL)

List of Evaluators

- Stéphane Sire (EPFL)
- Stéphane Rieppi (ULg)

Summary

e-Logbook is a collaborative Web-based environment deployed by the Swiss Federal Institute of Technology in Lausanne (EPFL) in the framework of Palette European Project and aimed at sustaining communication for communities of practice. It consists of an activity-oriented workspace where members of a community can form groups, conduct activities and collaborate around shared assets. The purpose of this deliverable is to present the second version of the e-Logbook tool by introducing its features and functions, especially the implemented awareness functions, the email-based interface of e-Logbook, and the interoperability of e-Logbook with Web applications. The e-Logbook tool is accessible online at http://elogbook.epfl.ch.

Table of contents

1 Introduction

The purpose of this deliverable is to present the second version of the e-Logbook tool intended to support collaboration and communication for communities of practice. The e-Logbook is a general-purpose activity-oriented collaboration space that can be customized by users to serve as a task management tool as well as an asset management system allowing the collaboration around shared artifacts. Moreover, awareness services of different types are provided by e-Logbook, because awareness is crucial in collaborative environments.

The e-Logbook is designed and implemented using a participative design approach with the primary objective of proposing adaptive and adequate solutions based on the expressed needs of communities of practice (CoPs). The description of e-Logbook's basic functions (invitation management, role management, deliverable management, asset rights managements, etc.) can be found in D.MED.03 (First version of the e-Logbook tool supporting exchange of artifacts at both a community and a technical level). An usability analysis of e-Logbook can be found in D.PAR.04 (User Centred Description of PALETTE tools and services and first analysis of usability).

In this deliverable, we will examine several additional functions and features supported by e-Logbook. This deliverable is structured as follows. Section 2 briefly describes the e-Logbook 3A model. Section 3 presents the implemented awareness functions of e-Logbook, in addition with a new search function. Section 4 presents e-Logbook email-based interface for users to use e-Logbook through emails. Section 5 discusses how e-Logbook interoperates with other applications through Web services. The section also discusses the interoperability between e-Logbook and CoPe_it! (URL:CPI). Finally, Section 6 outlines future considerations for e-Logbook.

1.1 Acronyms

- API: Application Programming Interface
- CSCW: Computer Supported Cooperative Work
- CoP: Community of Practice
- OpenID: A decentralized single sign-on system
- REST: Representational State Transfer
- RSS: Really Simple Syndication
- XML: Extensible Markup Language

1.2 Reading conventions

References by author, like (Gillet D., et al., 2007), appear in a bibliography at the end of the deliverable. References by code, like (URL: CPI), appear in a "Webography" (list of Web links), also at the end of the deliverable.

2 The e-Logbook 3A Model

From a general point of view, **people** use interaction technology to socialize, work and learn together through informal or formal **activities**. For these purposes, they establish synchronous or asynchronous discussion threads using phones, chat, email, blogs or wikis. They also populate distributed or centralized repositories. In these frameworks, they exchange digital information (**artifacts** or **assets)** at any time or at specified deadlines (**deliverables**). They also keep informed regarding the presence, the emotions and the activities of others trough audio or video conferencing, as well as through embedded dynamical **awareness** cues such as emoticons.

Without trying to provide a thorough classification of all *CSCW* solutions supporting the above situations/scenarios, we propose to map most of the mediated interaction alternatives using three fundamental entities: the actors, the activities and the assets. The definitions for these three entities and their possible relationships constitute what we call the 3A Model (Gillet D., et al., 2007). They are detailed thereafter and illustrated in Figure 1.

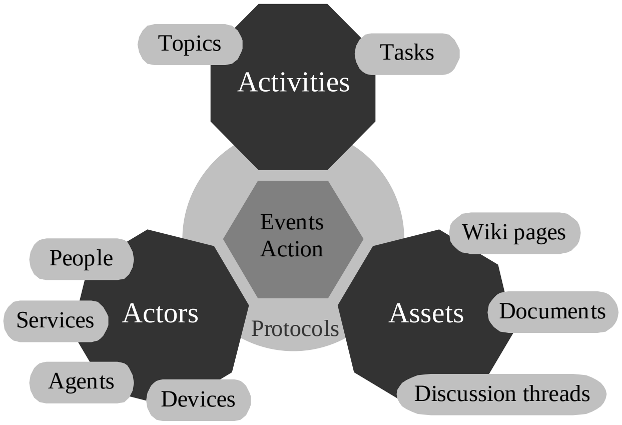


Figure 1: 3A interaction model.

An **actor** is any entity capable of initiating an event in the collaborative environment. It can be a person, a Web service, a software agent or even an online physical device. In that sense, the proposed definition of an actor is broader that the traditional definition of social software users or community members. The next section will show how this extension suits the social software requirements for supporting collaborative laboratory activities. From a constructivist perspective, an actor is an **artifact doing/producing** something, i.e. an agent or an instrument corresponding to “who?” or “which?”. Awareness related to actors can correspond to emotional, social or contextual statuses (i.e. online/offline, available/busy user or machine).

An **asset** is any kind of resource produced by or shared between community actors. The proposed definition goes beyond the typical digital assets like rich-text documents or multimedia resources. It can also include as example discussion threads or wiki pages. From a constructivist perspective, an asset is an **artifact done/produced** somehow, i.e. a product corresponding to “what?”. Awareness related to assets can correspond to preview, completion state (pending, under construction, completed, etc.), revision or moderation state, ownership, accessibility or historical data (trace) (Nutter D., Boldyreff C., 2003).

An **activity** is the formalization of a common objective to be achieved by actors such as discussing topics or completing tasks. From a constructivist perspective, an activity is an **artifact for doing/ for producing** something, i.e a purpose corresponding to “why?”. Awareness related to such artifact could be associated membership, context corresponding to “where?”, deadlines or deliverables corresponding to “when?”.

Events or Actions related to these three main entities are governed by Protocols corresponding to “how?”.

This model is compatible with computer-supported collaborative work solutions or social software dedicated to discussions, activity management or asset management; without putting the emphasis on any of them with respect to the others.

3 Completion of the awareness functions and the related Context-Sensitive View

e-Logbook provides context-sensitive awareness information crucial in collaborative environments (Rekik Y., Gillet D., El Helou S., Salzmann Ch., 2007). Awareness information is sent via email (implemented) and RSS feeds (under construction). The awareness information is also augmented in the context-sensitive Web interface seamlessly with pertinent awareness indicators.
In this section, we briefly outline the completed awareness functions and the related visualization view of e-Logbook. The detailed description of these functions can be found in D.MED.04 (First version of the awareness-support tool). We also describe our newly developed contextual search function.

3.1 Awareness functions of e-Logbook

At February 2008, the awareness functions of e-Logbook provide the following types of awareness information.

Social/Presence Awareness
Indication of presence status of users in e-Logbook can trigger interaction among the users. In e-Logbook, the presence status (online/offline) is perceived through the change of the color of the actor's icon.

Informal/Conversational Awareness
The context-sensitive view (GUI of e-Logbook) embeds several awareness cues of this type. For instance, the average rating of an actor, an activity or an asset, through the use of stars is displayed next to the central element name. The tag frequency is also highlighted through making the tag size proportional to its usage frequency.

Task Awareness
The reminder of the deadlines for the submission and the evaluation of deliverables is a kind of task awareness. Reminders are sent by email or accessible via RSS feeds. Also, in the context-sensitive Web interface, the order in which deliverables of an activity appear and the color in which the corresponding date is displayed varies in terms of the closeness of the deadline and whether or not the related submissions and/or evaluations have been completed.

Historical Awareness
All actions happening within the workspace by actors and related with shared assets, shared activities and other actors, are all logged with a time stamp.

Group-structural Awareness
When an activity is chosen as the focal element in the context-sensitive view, the different roles assigned to its members are provided. Different roles have different operational and administrative rights. In addition, the color of the button displayed next to the role indicates whether or not the user has accepted to join the activity and take the role s-he was assigned. In the same way, awareness cues embedded in the assets area of the context-sensitive view show the rights the central element has over the listed assets (i.e. an "editorship" right is represented with a pen, an "ownership" with a crown and a "view-only" right with an eye).

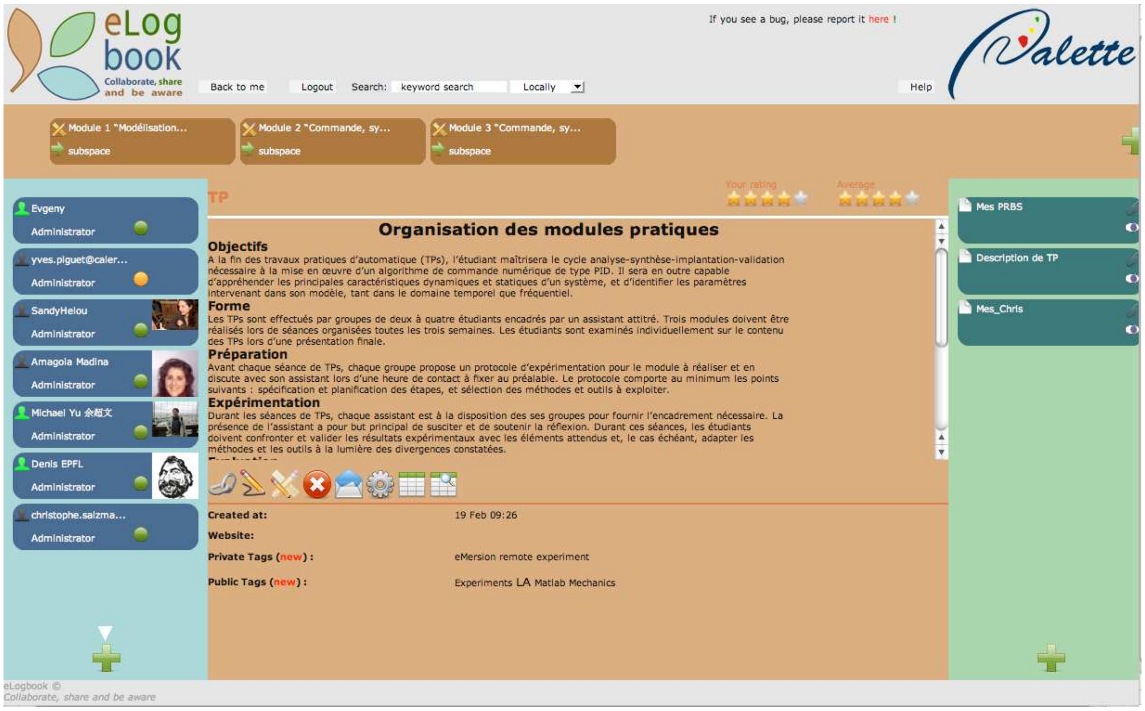
Workspace Awareness
e-Logbook provides awareness of the manipulation of a shared resource. For example, in the context-sensitive view, when a member is editing an asset with e-Logbook, a "Lock" icon replaces the editing and deletion icons for other actors who also have the right to edit this same asset. If users move the mouse over the "lock" icon, they will be made aware of who is editing the asset through a pop up message. In addition, and as it was mentioned earlier, delivering all the previously mentioned

awareness types, such as historical, presence and group-structural awareness are prerequisites for providing workspace awareness.

3.2 Embedded awareness cues in the Context-Sensitive View

The main Web interface of e-Logbook is called Context-Sensitive View. Awareness cues are embedded in the view for providing contextual information to users.

CoPs expressed the need to work in a user-friendly environment that can serve simultaneously as a task and asset management system, a social network, and a discussion platform. The context-sensitive view was designed as a response to this need. In fact, this view consists of a central element, surrounded by three main regions, respectively dedicated to activities, assets and actors (Gillet D., El Helou S., Salzmann Ch., Rekik Y., 2007). Based on the previously mentioned 3A model, the central element can consist of either an asset, an actor or an activity. Selecting an entity to become the context or the central element causes a change in the surrounding areas to display related entities, their relation with it and the eventual related actions that a user can perform on it. Consequently, just by changing the type of the focal point from an activity to an actor or an asset, the interface can serve different purposes, keeping however the same overall skeleton layout and structure. Awareness "cues" of different types are seamlessly incorporated in every area through the use of symbolic icons, colors and the manipulation of the layout order in which information is displayed. Figure 2 illustrates an example where a specific activity is chosen as the focal element. In this case, the assets posted in this activity, the actors participating in it and the other activities related to it, are displayed.



Organisation des modules pratiques

Objectifs

A la fin des travaux pratiques d'automatique (TPs), l'étudiant maîtrisera le cycle analyse-synthèse-implantation-validation nécessaire à la mise en œuvre d'un algorithme de commande numérique de type PID. Il sera en outre capable d'appréhender les principales caractéristiques dynamiques et statiques d'un système, et d'identifier les paramètres intervenant dans son modèle, tant dans le domaine temporel que fréquentiel.

Forme

Les TPs sont effectués par groupes de deux à quatre étudiants encadrés par un assistant attitré. Trois modules doivent être réalisés lors de séances organisées toutes les trois semaines. Les étudiants sont examinés individuellement sur le contenu des TPs lors d'une présentation finale.

Préparation

Avant chaque séance de TPs, chaque groupe propose un protocole d'expérimentation pour le module à réaliser et en discute avec son assistant lors d'une heure de contact à fixer au préalable. Le protocole comporte au minimum les points suivants : spécification et planification des étapes, et sélection des méthodes et outils à exploiter.

Expérimentation

Durant les séances de TPs, chaque assistant est à la disposition des ses groupes pour fournir l'encadrement nécessaire. La présence de l'assistant a pour but principal de susciter et de soutenir la réflexion. Durant ces séances, les étudiants doivent confronter et valider les résultats expérimentaux avec les éléments attendus et, le cas échéant, adapter les méthodes et les outils à la lumière des divergences constatées.

Created at:	19 Feb 09:26
Website:	
Private Tags (new) :	eMersion remote experiment
Public Tags (new) :	Experiments LA Matlab Mechanics

Figure 2: Embedded awareness cues in the Context-Sensitive View. Focal Element: Activity

3.3 Contextual search

A new feature of e-Logbook is *contextual search*. A contextual search function and interface has been implemented. This feature helps users to search for information. Users can choose to search within a local context or global context. "Local search" enables each user to search for information within the context regarding to the current focal element. "Global search" enables each user to search for information within all elements (i.e., actors, activities, and assets) related to the user. Therefore the contextual search enables users to search for information based on different scopes of context.

Currently the search function supports keyword search only. More search options may be provided according to user feedbacks.

4 Non-intrusive email and RSS interfaces

In addition to the Web-based access, e-Logbook also supports information delivery through a non-intrusive email-based interface. Thanks to this interface, the users can manage their activities, assets and awareness in a ubiquitous way. This alternative lightweight interface first facilitates the appropriation of e-Logbook by CoPs members. As a matter of fact, novice users can share knowledge artifacts and be aware of ongoing activities just by using their familiar email client software. Second, it eases e-Logbook access when connecting with smart phones or PDAs. In the coming future, e-Logbook will support RSS feed as another information delivery means. The compact format of RSS feeds is particularly useful for mobile users subjected to device constraints.

4.1 Email-based interface of e-Logbook

The email-based e-Logbook interface enables users to manage their activities, assets, and awareness by sending emails directly to e-Logbook and to receive requested information. Additionally, users can trace occurring events by receiving automatic notification emails from e-Logbook. The advantages of providing an email-based interface to e-Logbook 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-phone 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 e-Logbook interface provides offline information management and relies on PC-mobile 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 email-based interface to e-Logbook 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 e-Logbook properly.

First, this section presents the system flow for the email-based interface to e-Logbook. 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 e-Logbook actions can be triggered via email.

4.1.1 System flow for the email-based interface of e-Logbook

Figure 3 shows the system flow for the email-based interface to e-Logbook. The email address for accessing e-Logbook is *action.e-Logbook@epfl.ch*.

Every time e-Logbook 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 e-Logbook 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 e-Logbook 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 e-Logbook. 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 e-Logbook, then the requested action would be executed.

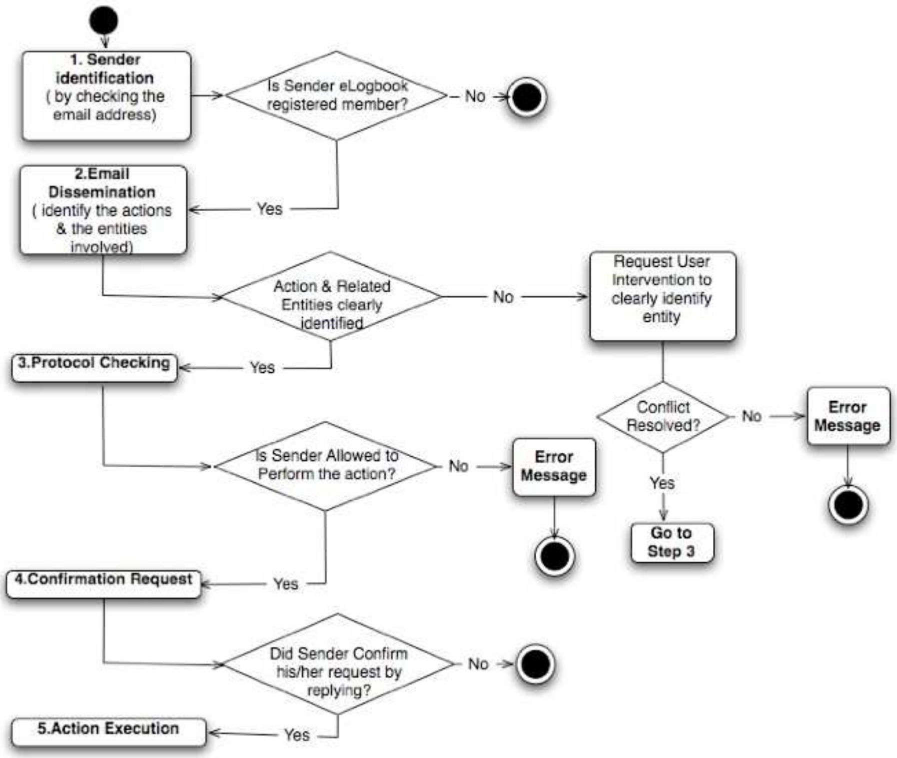

1. Sender identification (by checking the email address)
Is Sender eLogbook registered member?
No
Yes
2.Email Dissemination (identify the actions & the entities involved)
Action & Related Entities clearly identified
No
Request User Intervention to clearly identify entity
Yes
3.Protocol Checking
Conflict Resolved?
No
Error Message
Yes
Go to Step 3
Is Sender Allowed to Perform the action?
No
Error Message
Yes
4.Confirmation Request
Did Sender Confirm his/her request by replying?
No
Yes
5.Action Execution

Figure 3: System flow for the email-based interface to e-Logbook.

4.1.2 Syntax of emails

The email-based interface to e-Logbook allows users to write email in simple syntax to request for actions or information. Figure 4 shows the syntax of the request email from a user, and that of the confirmation email from e-Logbook.

A request email contains four parts. The *email recipient* states the email address of the e-Logbook 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 a 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>**’**

Figure 4: 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.

The confirmation email is sent by the e-Logbook server to the sender in order to authenticate her/his identities (Figure 5). The actor needs to reply to the confirmation email to invocate 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 e-Logbook 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.

Email recipient: <actor email address>
Email sender: <e-Logbook email address>
Email subject: Confirmation of the <type of action> action
Email body:
<information of the requesting action>
‘Reference: <reference number>**’**
Access link: <URL>

Figure 5: Syntax of a confirmation email.

4.1.3 Requesting information and tuning default notification mechanism

By default, an e-Logbook 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 e-Logbook notification mechanism through the Web-based or the email-based e-Logbook interface. In the latter case, the email subject and body are interpreted in order to invocate the user’s requested action.

- If the email subject is set to “disable all notifications”, then the user will stop receiving all sorts of e-Logbook 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” e-Logbook 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. e-Logbook 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 e-Logbook site or requested related notifications via email.

This feature is not implemented yet. It will be available at the end of July 2008.

4.1.4 Triggering administrative and organizational actions

e-Logbook 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 sending an email to e-Logbook (*action.e-Logbook@epfl.ch*), members can invocate 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 e-Logbook and users via email.

Creating an activity

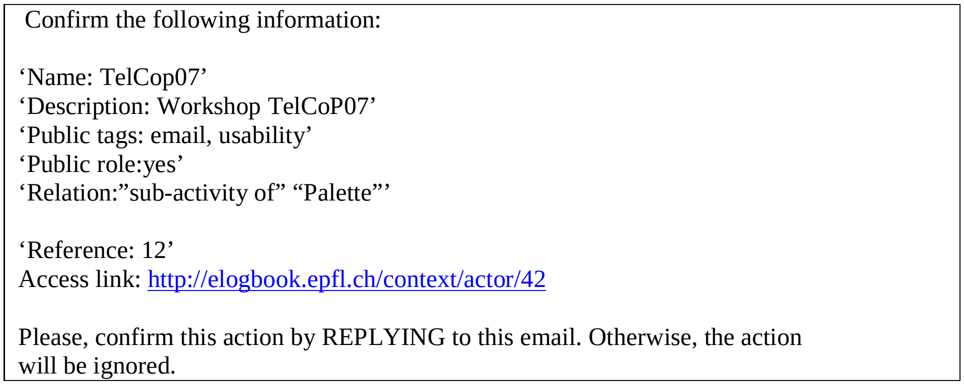
Setting the subject of the email to "create new activity" creates an activity with its default administrator being the sender of the email. The body of the mail should contain the *name of the activity*, optionally the activity description and the following fields:

- *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".

Figure 6 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, e-Logbook sends a confirmation email to the actor "amadina", as shown on the Figure 7. Once confirmed, e-Logbook runs the action requested, and the new sub-activity called "TelCoP07" for the existing activity "Palette" will be created.

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

Figure 6: 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.

Figure 7: Body of the confirmation email.

The examples of “creating an asset” and “creating a deliverable” can be found in Appendix.

4.2 Using RSS as the lightweight awareness delivery means

RSS is a family of XML-based data format that is used to publish frequently updated content such as news headlines. 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 information transfer within e-Logbook must abide by three primordial conditions: unobtrusiveness, relevancy and conciseness (El Helou S., Gillet D., Salzmann Ch., Rekik Y., 2007). The display of the context and content of an activity and/ or an asset in the e-Logbook interface was designed in such a way to take those three important aspects into account. Similarly, the RSS feeds were chosen as adequate notification delivery means especially for mobile users, because it allows meeting the three conditions mentioned above. To start with, unobtrusiveness is achieved because notifications are sent via a familiar user interface where users usually expect and intentionally check for updates. Secondly, the compact format of RSS feeds is particularly useful for mobile users subjected to device constraints. News is sent in a concise way and optional fields (such as images) are skipped for mobile users. Moreover, in order to meet the “relevancy” condition, information sent via RSS feeds to a target users, will depend on his/her expressed interests (i.e., they can for instance choose to receive news about particular actors, activities or assets).

5 Interoperability of e-Logbook with Web applications

e-Logbook provides activity management, asset management, and context awareness functions. To enable interoperability of e-Logbook with other Web applications, we have published several Web Services for e-Logbook. The detailed description of Web Services provided by e-Logbook can be found in D.MED.09 (Technical specifications of collaboration support tools as Web services).

Besides, we especially look into the interoperability with CoPe_it!. The ongoing investigation is a transparent profile synchronization between e-Logbook and CoPe_it!.
In the coming months, we will also look into the deployment of OpenID on e-Logbook as a single sign-on solution.

5.1 Interoperability through Web Services of e-Logbook

We have published a number of functionalities of e-Logbook as REST Web Services. They are available to any tool interested in integrating services provided by e-Logbook in the framework of the Palette project. The services are classified according to the following two divisions: entity management services, and query services. e-Logbook currently provides the following REST Web services:
Entity management services: create activities, create assets, create deliverables, and update status on activity.
Query services: query activity, query actors, query asset, and query deliverable.

A number of Web services are under construction. We plan to provide the following REST Web services for e-Logbook at the end of July 2008:
Entity management services: update {activity/asset/deliverable}, delete {activity/asset/deliverable}, and post asset in deliverable.
Metadata management services: tag {activity/asset/actor}, rate {activity/asset/actor}, retrieve tags of {activity/asset/actor}, and retrieve ratings of {activity/asset/actor}.

Before running the Web services provided by e-Logbook, the service consumers need to authenticate themselves. They can do this by the following HTTP request:
http://e-Logbook.epfl.ch/?email=[login email]&password=[password]
where [login email] and [password] are replaced by the user login email and passwords, respectively.

The detailed description of Web Services provided by e-Logbook can be found in D.MED.09 (Technical specifications of collaboration support tools as Web services).

An example of using the e-Logbook's Web Service for interoperation is the "global cross-tool search" service of the PALETTE project. The "global cross-tool search" is a composite service, which aggregates individual search Web Services (SOAP or REST) provided by some PALETTE services. It acts as a query dispatcher to these services, allowing a single access to data and information handled by different PALETTE services. e-Logbook's query services are called by the "global cross-tool search" service for an integrated search among Palette services.

The detailed description of "global cross-tool search" can be found in D.IMP.05 (First version of PALETTE Integration: Conceptual and Technical Integration).

5.2 Interoperability with CoPe_it!

We now look into achieving interoperability between e-Logbook and CoPe_it! through **transparent profile synchronization**.

Knowing that CoPe_it! and e-Logbook are stand-alone Web applications independent from each other, it would not be an optimal solution to centralize profile information in one place. Nevertheless, users should still be allowed to choose to automatically or occasionally synchronize common profile data stored in the different Palette tools, as this would save them from the burden of repeating the same actions in two different places.

It is also worth noting that there are some application-dependent profile data, needless to be propagated from one tool to another. As an example, the option «public profile» in e-Logbook could be set to true or false, depending on whether or not the user wishes to make their account visible to all other e-Logbook registered users. This option does not exist in CoPe_it!, and so, there is no point in propagating any corresponding change. As for common profile data, such as home address, phone number, picture, and Website, an agreement between tools to follow the same naming conventions makes it easy for users to identify these common fields as he/she visits the different Web applications, and makes it easy for services to communicate the corresponding modifications (the presentation or visual integration level is concerned with this issue).

The implementation of this transparent profile synchronization feature can be described as follows. In the e-Logbook page where users can fill in their profile information, the options to always or occasionally propagate changes related to common profile data, to CoPe_it! are made available. Then, every time the user profile information is updated in e-Logbook, the latter checks whether any of the two options mentioned previously are set to true. If this is the case, then e-Logbook invokes an external REST Web service responsible of updating the user profile data in CoPe_it!.

Using an OpenID solution with API authentication will help make this process transparent to the user, as he/she will not be requested to enter login information for CoPe_it!. As a matter of fact, he/she will only need to authorize e-Logbook to invoke the external service on his/her behalf. After having the user's delegation, CoPe_it! will generate an access token for e-Logbook to access the needed service. e-Logbook can then use this token to invoke the function responsible for updating the user profile data in CoPe_it!. The same process is followed to propagate profile information from CoPe_it! to e-Logbook. Finally, the same procedure can also be followed to propagate profile information changes among other Palette tools.

(Note: The description of the transparent profile synchronization between CoPe_it! and e-Logbook is also included in D.IMP.05 (First version of PALETTE Integration: Conceptual and Technical Integration).)

6 Future Considerations

The e-Logbook awareness mechanism is partially implemented in the way that awareness information can be delivered to users through Web interface and email-based interface. The implementation of delivering awareness information through RSS feeds is ongoing. The mechanism of awareness filtering is still in the design phase.
The full implementation of awareness filtering and delivery will be the objective of our future work. We expect that the RSS feed based awareness delivery will be available at the end of July 2008. And this function will possibly be demonstrated as prototype during the review on 8-9 April 2008.

More entity management functions and metadata management functions of e-Logbook will be published as REST Web Services at the end of July 2008.
The approach of interoperability between e-Logbook and CoPe_it! now goes to the direction of transparent profile synchronization. This requires further design. This feature will be implemented before end of July 2008.

7 Conclusion

The e-Logbook can be perceived as a task-oriented workspace, a discussion-platform as well as an asset repository intended to sustain collaboration and communication for communities of practice. Its strength lies in the fact that it was designed to be as general and flexible as possible in order to meet the needs of different types of communities of practice. This is achieved by offering the possibility to tune functions and features (e.g., roles and access rights) based on each community's specific needs.

The email-based e-Logbook interface enables users to manage their activities, assets and awareness through email. It provides an alternative lightweight non-Web interface to ease e-Logbook access when using smart phones or PDA and to facilitate e-Logbook acceptance for community members hesitating to learn an additional Web environment.

e-Logbook now provides a number of REST Web Services. Other Web applications can interact with e-Logbook by calling these services.

8 References

8.1 Webography

All the Web references have been accessed in February 2008.

- CPI : CoPe_it!, at http://copeit.cti.gr/

8.2 Bibliography

S. El Helou; D. Gillet; Ch. Salzmann ; Y. Rekik: Feed-Oriented Awareness Services for e-Logbook Mobile Users. In *3rd International Conference on interactive Mobile and Computer Aided Learning*, Jordan, April 16-18, (2007).

D. Gillet; S. El Helou; Ch. Salzmann; Y. Rekik: Context-Sensitive Awareness Services for Communities of Practice. In *HCI International 2007*, Beijing, China, July 22-27, (2007).

D. Nutter, C. Boldyreff: Historical Awareness Support and Its Evaluation in Collaborative Software Engineering. *Twelfth International Workshop on Enabling Technologies: Infrastructure for Collaborative Enterprises* (2003) 171.

Y. Rekik; D. Gillet; S. El Helou; Ch. Salzmann: The e-Logbook Framework: Sustaining Interaction, Collaboration, and Learning in Laboratory-Oriented CoPs. In *International Journal of Web-Based Learning and Teaching Technologies*, vol. 2, num. 3, (2007), p. 61-76.

Appendix

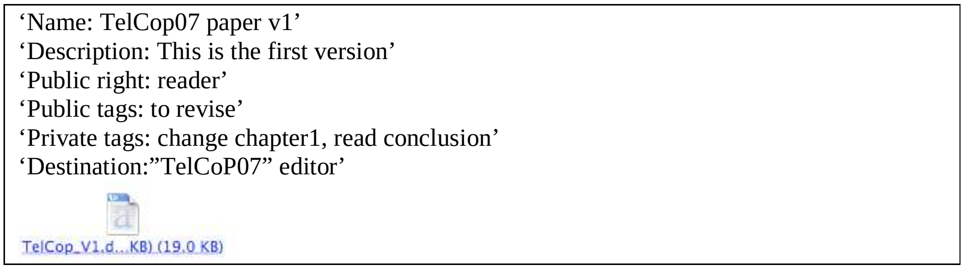
This appendix describes two examples of using email-based interface of e-Logbook.

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. 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.

Figure 8 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, e-Logbook sends a confirmation email to the actor "amadina" (similar to the one we have seen in Fig. 5). Once confirmed, e-Logbook runs the action requested, and an asset called "TelCoP07 paper v1" is created.



'Name: TelCop07 paper v1'
'Description: This is the first version'
'Public right: reader'
'Public tags: to revise'
'Private tags: change chapter1, read conclusion'
'Destination:"TelCoP07" editor'

Figure 8: Body of the email of creating a new asset.

Creating a deliverable

Setting the subject of the email to "create new deliverable" creates a deliverable inside an activity. 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").

Figure 9 shows the body of an email for creating new deliverable. The actor "amadina" wants to create a deliverable called "del07-07" for the activity "TelCoP07".

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

Figure 9. Body of the email for creating a new deliverable.