First version of the eLogbook tool supporting exchange of artifacts at both the community and the technical level

Yassin Rekik, Christophe Salzmann, Chui Man Yu, Sandy El Helou, Amagoia Madina

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.03 – First version of the eLogbook tool supporting exchange of artifacts at both the community and the technical level

Due date of deliverable: 30 April 2007
Actual submission date: June 12, 2007

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

Organization 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: collaboration, assets, artifacts, communication, communities of practice, collaborative learning, information exchange.
Responsible Partner: Denis Gillet, EPFL

MODIFICATION CONTROL			
Version	Date	Status	Modifications made by
1	May 15, 2007	Draft	EPFL
2	May 30, 2007	Draft	EPFL
3	June 5, 2007	Ready for review	EPFL
4	June 8, 2007	Reviewed	EPFL
5	June 11, 2007	Final	EPFL

Deliverable manager

- Denis Gillet (EPFL)

List of Contributors

- Yassin Rekik (EPFL)
- Christophe Salzmann (EPFL)
- Chui Man Yu (EPFL)
- Sandy El Helou (EPFL)
- Amagoia Madina (EPFL)

List of Evaluators

- Manfred Kunzel (UNIFR)
- Manolis Tzagarakis (CTI)

Summary

eLogbook 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 first version of the eLogbook tool by introducing its features and functions, which are intended to support communication and collaboration. The eLogbook tool is accessible online at http://elogbook.epfl.ch.

Table of content

1 – Introduction

The purpose of this deliverable is to present the first version of the eLogbook tool intended to support collaboration and communication for communities of practice. The *eLogbook* 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 eLogbook, because awareness is crucial in collaborative environments.

The eLogbook is designed and implemented using a participative design approach with the primary objective of proposing adaptive and adequate solutions based on the communities of practice expressed needs. The eLogbook features and functions, which address those needs, are examined below. First, the eLogbook 3A model is presented. Second, the eLogbook as an activity-oriented space, ideal for managing and organizing shared tasks, is discussed. Third, an explanation of how eLogbook serves as a repository of assets, where people can collaborate through artifacts sharing and information exchange follows. Then, the underlying awareness mechanism is introduced. Finally, two examples are discussed to illustrate how the eLogbook features and functions indeed satisfy the CoPs requirements.

2 – 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. A typical event example would be posting an asset in the workspace. An *asset* is any kind of resource (e.g. text documents, images, graphs, simulation snapshots) shared between community actors. An *activity* is the formalization of a common objective to be achieved by a group of actors. Actions performed over the entities, and 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 enclose all other kinds of non-organizational collaborative actions such as the manipulation of shared *assets* or the submission of *activity* deliverables. Moreover, the interactions between the three main entities are governed by *Protocols*.

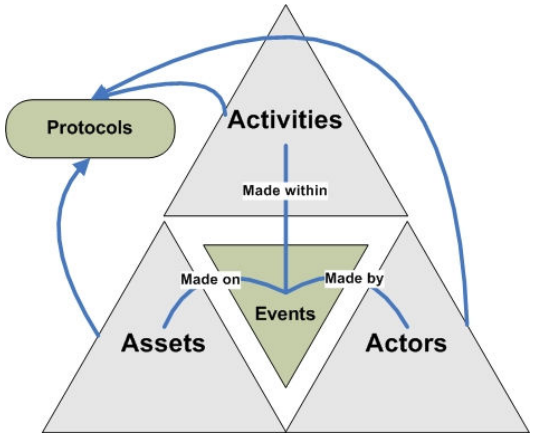

Figure 1: The 3A model.

3 – A Task/Activity-Oriented Workspace

eLogbook consists of a task-oriented workspace where common objectives can be set, tasks can be organized within activities, and roles distributed among different community members. To each activity, one or more role(s), objective(s) and deliverable(s) are associated. Each member taking part of an activity is assigned a role, which not only serves as a « label » for him/her but also defines the set of rights this member has over the activity itself and its resources. A detailed description of the management of activities with eLogbook is given below.

3.1 Types of activities

An **open activity** is an activity to which all the people who have an eLogbook account or enter to eLogbook, as guests know about it. Making an activity open consists of setting one of its roles to be the public one. This being said, if an activity has a defined public role, then everyone will know about it and will be invited to join it and acquire the rights associated with the public role.

A **close activity** is just like an open activity. It is announced to all eLogbook registered members as well as its guests. Nevertheless, unlike the open activity, no one can access the resources of a close activity before he/she decided to join it. Anyone with a pending status only knows about the existence of this activity. As in the case for the open activity, when the user joins it, he/she will be able to acquire all rights associated with its public role. A close activity is done by defining a role as public and defining the public invitation style to be “compact view” rather than “complete view”.

A **private activity** is an activity that is secret to everyone except for the people explicitly invited by the administrator. At any point in time, the administrator of an activity can decide to change the scope of his/her activity by completely or partially opening it to the public, or « privatizing » it. Finally, it is important to note that even though an activity can be made public, direct invitations can be done for specific people and roles different than the public role (usually stronger) assigned to them.

3.2 Invitations Management

To be able to access an activity, a user must receive an invitation from its administrator(s). If the activity is public (close or open), then invitations are automatically sent to all registered users. Still, the administrator can explicitly invite people to join the activity, using person-dependent, role-dependent and/or activity-dependent invitations.

Person-dependent invitations consist of inviting people to join the activity, by specifying their username or email (in case they are not already an eLogbook registered member). In this case, memberships are unconditional. This means that if the invited user decides to join the activity, then he/she will remain a member unless the administrator deliberately decides for some reason or another to « fire » him/her. This is not the case for activity and role-dependent invitations, as the memberships acquired through those kinds of invitations are conditional.

In fact, **Activity-dependent invitations** occur when the administrator of an activity Y invites members of activity X to join his/her activity. This engenders the creation of a dynamic link between activities Y and X. Consequently, if new member joins activity X at time t, then he/she is automatically invited to activity Y without the intervention of the administrator of activity Y. If at time t+1, this member decides to leave activity X, then not only is he deprived from his/her rights over activity X but he/she automatically looses the inherited membership to activity Y.

Role-dependent invitations engender a stronger membership condition or constraint. Eligibility to become a member of an activity Y dynamically linked to activity X, requires maintaining a specific role W within activity X. Role and activity-dependent invitations are commonly used in well-structured CoPs where activities are divided into sub-activities and members of the mother activity are invited to join the sub-activities with the condition that they stay members of the mother activity. The figures below shows role-dependent invitations to Wikiprepas, « Members » of ePrep are invited to take the role « Observer » in Wikiprepas.

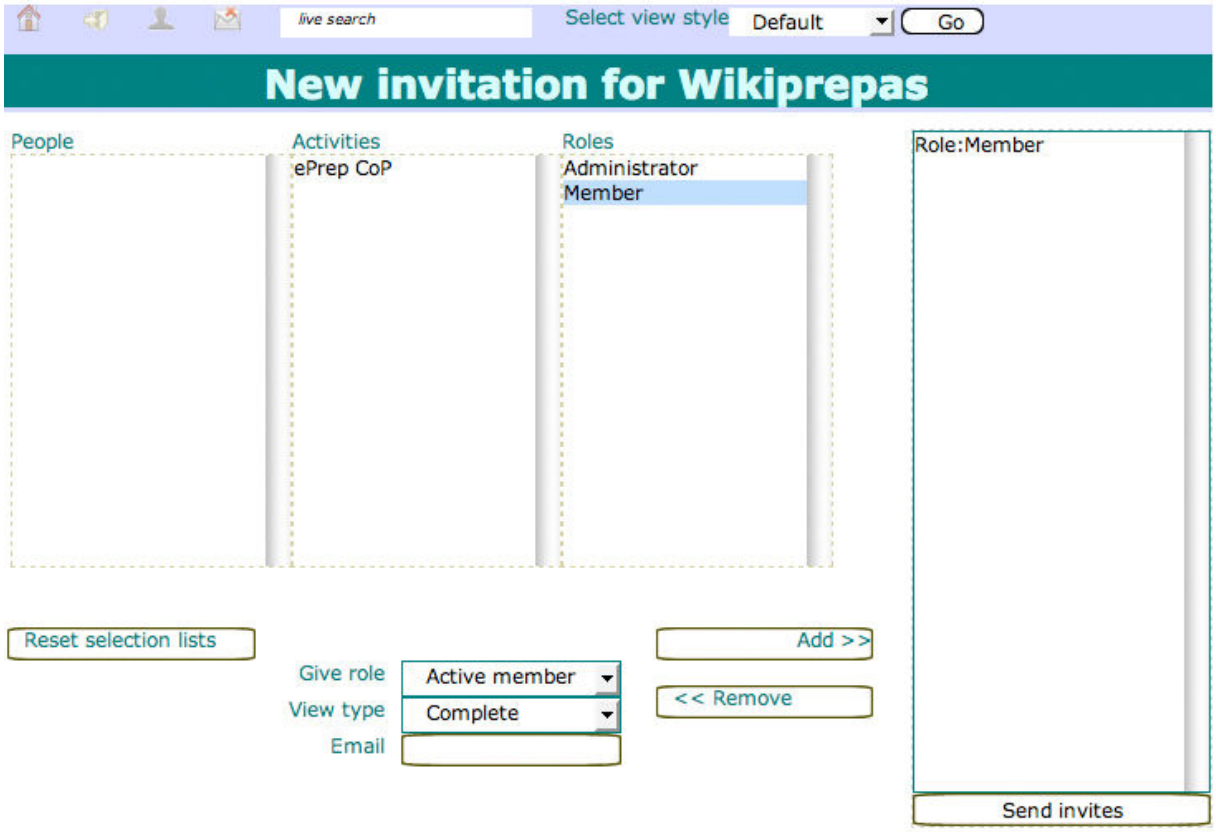

Figure 2: Role-Dependent Invitation to Wikiprepas.

3.3 Roles Management

A user is always invited to join an activity with a particular role. This role is the bridge through which he acquires rights over the activity in question. Rights can be split into two categories: operational and administrative. Operational rights allow users to perform all types of operational actions within an activity such as posting assets, linking, tagging and rating the activity. Acquiring administrative and managerial rights permit the users to perform administrative actions such as managing the roles, deliverables and invitations of the activity.

To facilitate the task of managing roles, the administrator can choose between three different default role types: the "administrator", the "user" and the "limited user".

- To the default type "administrator", are associated all types of administrative and operational rights.
- To the default type "user", are associated all types of operational rights.
- The default right "limited user" allows the administrator to create a role and define a particular combination of rights for it.

The figure below illustrates how the administrator(s) of “Wikiprepas”, a sub-activity of ePrep, can manage roles. Since the public role is set to “None”, then Wikiprepas is a secret or close activity.

Figure 3: Managing Roles in eLogbook.

3.4 Deliverables Management

Managing deliverables include administrative actions such as creating deliverables setting/changing the submissions and validations deadlines, and specifying the maximum/minimum number of submitters. As usual, there are default assumptions, which facilitate the task of managing deliverables. For example, by default, the submitters within a particular activity are the members whom their role allows them to submit deliverables. Nevertheless, in the case where the administrator wishes to define for a particular deliverable, a subset of submitters from among the members originally allowed to submit deliverables, then he/she can easily do it by setting the flag « specify submitters » to true. The same applies for evaluators. It is also possible to define a specific submission order for the deliverables, by linking them using predefined ordering links (“precedes”, “succeeds”).

4 – A repository of Assets

Assets can be defined, as anything tangible or concrete owned by a member, or shared by a group or by the whole community. They can consist of files of any type, such as images, documents, and graphs. eLogbook allows its members to create, share and collaborate over shared assets by editing (by uploading new versions), tagging, linking, rating and submitting them for deliverables.

4.1 Assets Rights Managements

Acquiring rights over an asset follows the same principle and rules of acquiring membership of an activity. As a matter of fact, granted access rights over assets can be done in an actor, activity or role-dependent way. To explain, a member of a community can see an asset if one of the following three cases hold:

a) If he/she has a specific role within an activity. (Role-dependent access grant).

b) If he/she is member of an activity. (Activity-dependent access grant).

c) If he/she was given right over it on a direct personal basis. (Actor-dependent access grant).

The three different rights, which can be granted over an asset, are: “reader”, “editor”, and “owner”. As it is the case for an activity, if the default or public right of an asset is set, then the asset is made public. “Readers” of an asset can only read it and check its related “metadata” such as its associated tags, links, and rates. “Editor” of assets can upload new versions of the assets, tag it, link it, submit it and rate it. “Owners” of the asset not only have editing rights but also distribution rights. This means they can grant access rights over this asset.

The figure below shows how the « Active member» of Wikiprepas subactivity, is granted the « owner » rights over the assets « Choice of Wikiprepas ».

Figure 4: Setting access rights for "Choice of Wikiprepas".

Posting an asset in an activity can be done by dynamically granting this activity a view-only, edition or ownership right over it. Exceptionally, even if the activity was not granted ownership rights over an asset, it can still « distribute » it for its members. This being said, if the activity owns the asset, the administrator of the activity can freely distribute possible rights for the members. Nevertheless, if the activity has view-only or "reader" rights over an asset, the only rights it can assign to its members is "reader" or "none". As a general rule, an activity can only distribute to its members, asset access rights, which are less, or equal to the right it has originally been granted for a particular asset.

4.2 Actions over Assets

Each member can define tags over the assets he can access. He/She can choose to make the tags private or public. If set to private, then the tags can only be seen by their creator, otherwise all the community members who can see the tagged entity can see its tags. The same applies, for directional and bidirectional semantic links serving to relate one asset to another, which can also be either personal or public. There are predefined link types such as « complements » but the user can define his/her own ones. Assets can be rated or evaluated as they can be submitted or delivered as a response to a specific deliverable requirement with an activity.

It is worth mentioning that, just like assets, activities, actors and deliverables can be rated, tagged and linked.

5 – Underlying Awareness Mechanism

Awareness mechanisms are essential in cooperative and collaborative environments where people interact through task sharing and assets exchange.

Providing awareness requires an adequate underlining infrastructure and mechanisms allowing to intelligibly trace all occurring events. Furthermore, many studies have shown that excessive unnecessary notifications might lead to adverse effects such as a decrease in productivity. For this reason, personalizing the awareness information, delivery and display means, and adapting it to each member needs, interests and expectations become an important need. The eLogbook underlying awareness mechanism is briefly examined in this section. It is worth mentioning, the actual implementation of this mechanism is still in its very early phase.

5.1 eLogbook awareness types

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 assets 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 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.

5.2 Embedded Awareness in the Context Aware View

This view consists of a center or focal element chosen by the user, surrounded by four regions, each of which listing related entities of a special kind. Based on the previously mentioned 3A model, the center element can consist of an asset, an actor, an activity or a deliverable. In the surrounding areas, not only are the entities related to the center

displayed, but also their relation with the center entity and the eventual related actions - that the current user is allowed to perform - are appended. Awareness " cues" of different types are seamlessly incorporated in every area through the use of symbolic icons, colors and the manipulation of the order in which information is displayed. More details about the awareness information embedded within this view will be discussed in a future deliverable. Below is an example of use of eLogbook by the ePrep community. It shows the scenario where a member is selected as the centre element. In this case, the user sees all activities and sub-activities to which she was invited in the top area of the view. The figure shows that she has joined ePrep (the status button is green) and her status in Wikeprepas for the different roles she was suggested. To the left of the center element, the actors who share those activities with her are listed. Below it, the deliverable "Conclusion", she is expected to deliver, appears. The right side of the view is reserved for the display of all the assets the user can access. In this particular example, the document "Choice of Wikiprepas" appears. The "crown" icon displayed next to the document's name, indicates that the user has ownership rights over the document. Linking, editing, distributing access rights, submitting and deleting the assets can be done by clicking on the corresponding icons displayed below the document's name. If a role is still pending, the status button is orange.

Figure 5: Snapshot from eLogbook Contextual View.

5.3 Delivering Personalized Awareness Services

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 and the user's profile to elicit the user interest in a specific event.

5.3.1 Contextual Events Logging

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.

5.3.2 Default Systems Rules

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. It is important to mention that this does not mean that the information is hidden from the user, because at any time, the eLogbook, which keeps track of all events, can be visited to get an overall view of what has been happening in the community's collaborative workspace.

5.3.3 Management of User Profile & Notification Preferences

Every member has a space where he/she can edit his personal profile and his notification preferences. The member's personal profile includes his contact information, as well as his level of expertise and domain of interest. The system relies in this information to elicit the user's interest in a specific event. Moreover, explicit user-defined notification preferences or rules can be defined. These rules have a

priority over default rules and allow the specification of what, how and when a user would like to be notified. It is also planned to make the user's defined rules public, and if so, another user can simply choose to import another user's notification rules.

6 – eLogbook as a solution for CoPs Needs

After having described in details in the eLogbook features and functions, it is useful to discuss how they meet the needs of communities of practice. For that reason, we examine the stated needs of Lancaster and Learn-Nett (https://bscw.ercim.org/bscw /bscw.cgi/ 172617) and the corresponding solutions offered by eLogbook.

6.1 eLogbook & Lancaster

6.1.1 Lancaster expressed needs

- A formalized & systemized archived system.
- An easier access and more intuitive format for discussions.
- An integrated system for copyright clearance.
- An easy access to the documents in the virtual learning environment with possibility of versioning documents such as thesis presentation.
- To have a tool that tells students submit their work and that lets tutors give feedback on-line.

6.1.2 eLogbook proposed solution

To start with, eLogbook offers a flexible and formalized **assets repository** where exchanged artifacts can be easily distributed and accessed. Members will be able to search the assets stored by specifying different selection criteria such keywords, date of creation, owners. They can also easily track all comments, tags, links and/or rates related to an asset. It is also possible to define the scope of visibility of each asset (see section 4.1).

In order for tutors to **track the progress of students**, the latter can upload their prelim, their reflections on their thesis and give access to their tutors, so that they can comment on them, evaluate them, and/or link them to other documents deemed interesting for the students to check.

Last but not least, eLogbook also offers an easy way to **manage assignments** (post them, define deadlines, track the submissions, and evaluate them) by relying on the eLogbook deliverable management system previously described. Awareness information, embedded within the views (such as highlighting deliverables or assignments with close deadlines for students who haven't yet submitted them) or sent

via email or RSS notifications, is also crucial for the Lancaster community to track the student's progress.

6.2 eLogbook & Learn-Nett

6.2.1 Learn-Nett expressed needs

- Need to find traces of student's learning & evaluate Learn-Nett training accordingly
- An easier access and more intuitive format for discussions.
- Tutors need a private space.
- Need to collect exchanged practice, store & share them.
- Need to sustain "orphan activities" such as task sharing.

6.2.2 eLogbook proposed solution

In order to **sustain task sharing**, activities and sub-activities, in which objectives or headlines are specified, can be created. Concrete deliverables with due dates can also be added. Task or Activity based awareness (e.g.: reminder of task final evaluation dates and requirements) and group structural awareness (e.g.: showing the role and rights assigned to each activity member) are provided by eLogbook to promote task sharing.

Concerning the need for a **private space**, tutors can indeed create private activities (as described in sections 3.1 and 3.2) in which they can put assets outside the reach of the students and possibly schedule the task they share.

Finally, by logging all events and providing the tutors (and other community members) relevant and meaningful awareness information and related to the activities in which they are involved and the assets they can access statistics (e.g.: how many students have read the FAQ questions and how many have rated or tagged an important document posted by their tutor), the eLogbook answers to the tutor's need to **trace the students learning process**.

7 – Future Considerations

The eLogbook awareness mechanism is still in the design phase and its full implementation will be the object of our future work. After that, a testing phase of its usability shall be tackled in order to get a feedback from the communities of practice and improve the awareness mechanism accordingly.

A primitive scenario of interoperability with Cope-it! has already been demonstrated and will be developed further. The idea consists of calling Cope-it! from eLogbook in order to sustain argumentation for a community of practice using eLogbook, at a time where the latter does not offer this feature. Alternatively, Cope-it! users could benefit from the eLogbook Context-Aware View, a feature or service not supported by Cope-it!. Importing this eLogbook feature is very useful to get seamlessly embedded informal, conversational, task-based, presence and group structural awareness, because this can highly influence the position of a person in collaborative environments supporting mediation and argumentation.

The general framework for interoperability with other Palette tools will also be design in the next months.

8 – Conclusions

The eLogbook 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 taking general default assumptions and rules with however offering the possibility to tune functions and features based on each community's specific needs.

Advanced features such as deliverable management are optional. Basic usage of the eLogbook by CoPs with no formal operational requirements (see the Palette CoP model) is possible without defining deliverables. As such, the eLogbook can follow the evolution of the CoPs by offering over the time the new features required by their new interaction modes and requirements.

9 – References

P. Dourish; V. Belloti: Awareness and coordination in shared workspaces. In *Poc. Computer supported cooperative Work (CSCW'92*), 1992.

C. Speier et al.: The effects of task interruption and information presentation on individual decision making, In *Proc ICIS'97.*

J.B Spira; J.B. Feintuch: The Cost of Not Paying Attention: How Interruptions Impact Knowlege Worker Productivity. In *Basex*, 2005.

C. Gutwin; S. Greenberg: A descriptive framework of workspace awareness for real-time groupware. In *Journal of Computer-Supported Cooperative Work*, pp.411-446, 2002.

L. Capra et al.: Middleware for Mobile Computing: Awareness vs. Transparency. In *Proc. of the 8th Workshop on Hot Topics in Operating Systems (HotOS-VIII), (Schloss Elmau, Germany)*, pp. 164, May 2001.

Y.Rekik; D.Gillet; S.El Helou; Chr.Salzmann: The eLogBook Framework: Sustaining Interaction, Collaboration, and Learning in Laboratory-Oriented CoPs. In *International Journal of Web-Based Learning and Teaching Technologies* (in press).

D. Gillet; Ch. Salzmann; Y. Rekik: Awareness: An Enabling Feature for mediated Interaction in Communities of Practise. In *1st European Conference on Technology Enhanced Learning*, Crete, Greece, October 2, 2006. (online version: http://cnm.open.ac.uk/projects/ectel06/pdfs/ECTEL06WS9d.pdf)

D. Gillet; A. V. Nguyen; Y. Rekik: Collaborative Web-based Experimentation in Flexible Engineering Education. In *IEEE Transactions on Education, Special Issue on Web-based Instruction*, Vol. 48, No. 4, pp. 696-704, 2005.

D. C. McFarlane: Comparison of four primary methods for coordinating the interruption of people in human-computer interaction. In *Human-Computer Interaction*, 17(1), pp 63-139.

S. El Helou; D. Gillet; Ch. Salzmann ; Y.Rekik: Feed-Oriented Awareness Services for eLogbook 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.

10 – Annex

This section explains the underlying system's architecture for eLogbook and provides some technical specifications. The eLogbook was implemented using the Ruby on Rails Web development framework. Figure 6 illustrates The Model-View-Controller (MVC) architecture used in Ruby on rails and Figure 7 shows a summarized example of the system architecture we used to deploy the application. On the front end, Apache 2 is used with mod_proxy_balancer to balance the Rails application servers running in Mongrel. MySQL 5.0.27 is used to store the application data.

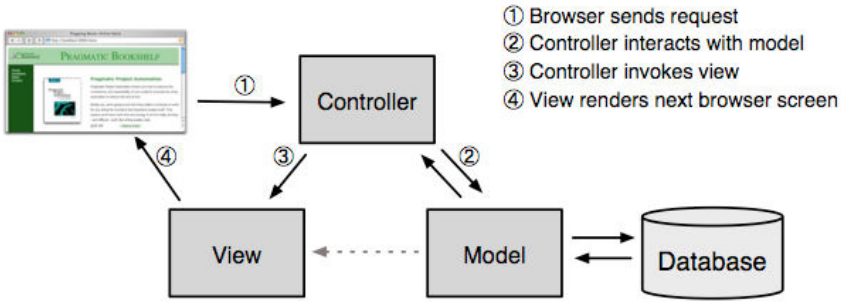

Figure 6: System architecture.

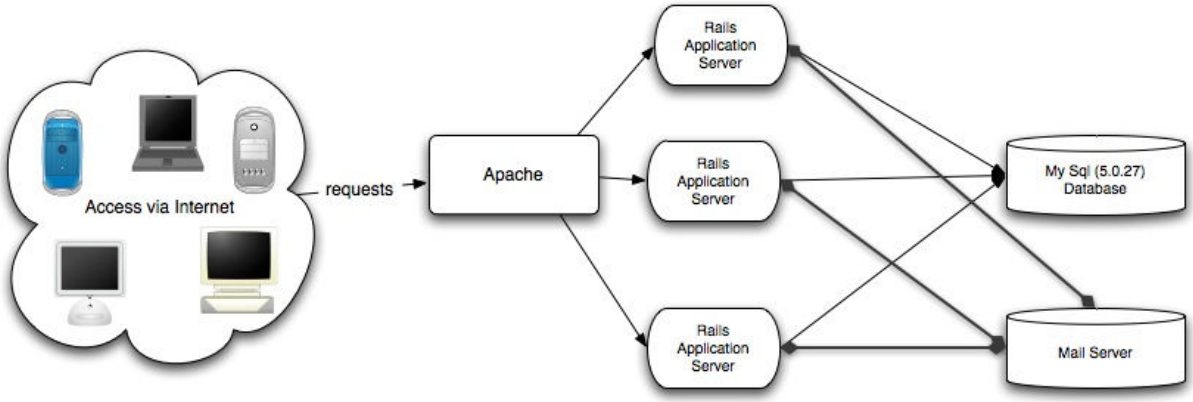

Figure 7: System architecture.