

Association for Information Systems AIS Electronic Library (AISeL)

CONF-IRM 2010 Proceedings

International Conference on Information Resources
Management (CONF-IRM)

5-2010

41P. Practical Lessons Learned while Developing Web 2.0 Collaboration Services for Communities of Practice

Manolis Tzagarakis

University of Patras, Greece, tzagara@upatras.gr

Nikos Karacapilidis

University of Patras, Greece, nikos@mech.upatras.gr

Nikos Karousos

Research Academic Computer Technology Institute, karousos@cti.gr

Sandy el Helou

Swiss Federal Institute of Technology In Lausanne, dy.elhelou@epfl.ch

Denis Gillet

Swiss Federal Institute of Technology In Lausanne, denis.gillet@epfl.ch

See next page for additional authors

Follow this and additional works at: <http://aisel.aisnet.org/confirm2010>

Recommended Citation

Tzagarakis, Manolis; Karacapilidis, Nikos; Karousos, Nikos; el Helou, Sandy; Gillet, Denis; and Esnault, Liliane, "41P. Practical Lessons Learned while Developing Web 2.0 Collaboration Services for Communities of Practice" (2010). *CONF-IRM 2010 Proceedings*. 24.

<http://aisel.aisnet.org/confirm2010/24>

This material is brought to you by the International Conference on Information Resources Management (CONF-IRM) at AIS Electronic Library (AISeL). It has been accepted for inclusion in CONF-IRM 2010 Proceedings by an authorized administrator of AIS Electronic Library (AISeL). For more information, please contact elibrary@aisnet.org.

Authors

Manolis Tzarakis, Nikos Karacapilidis, Nikos Karousos, Sandy el Helou, Denis Gillet, and Liliane Esnault

41P. Practical Lessons Learned while Developing Web 2.0 Collaboration Services for Communities of Practice

Manolis Tzagarakis, Nikos Karacapilidis
University of Patras, Greece
tzagara@upatras.gr, nikos@mech.upatras.gr

Nikos Karousos
Research Academic Computer Technology Institute, Greece
karousos@cti.gr

Sandy el Helou, Denis Gillet
Swiss Federal Institute of Technology In Lausanne (EPFL), Switzerland
{sandy.elhelou, denis.gillet}@epfl.ch

Liliane Esnault
EM LYON, France
esnault@em-lyon.com

Abstract

Although a plethora of Web 2.0 applications exist today, there is little literature reporting on experiences, concrete recommendations or best practices when developing such applications. The scarcity of such records makes it difficult for developers to determine how best to support the practices of communities with the use of Web 2.0 technologies. In this paper, we report on eight practical lessons learned while developing Web 2.0 collaboration services for Communities of Practice in the framework of a three years long European research project on Technology Enhanced Learning. The main objective of this project was to investigate how Web 2.0 technologies could impact the communication and collaboration needs of Communities of Practice interacting online and, conversely, how new interaction needs could impact Web 2.0 technology. The above lessons are presented in a way that could aid people engaged in various phases of the development of Web-based collaboration support services.

Keywords

Collaborative Work, Collaborative Learning, Knowledge Management, Communities of Practices, Web 2.0 Social Software.

1. Introduction

Designing software systems that can adequately address users' needs during diverse collaboration settings has been a major research and development activity for more than twenty years. Technologies and software tools supporting collaboration usually provide the means for discourse structuring and visualization, knowledge management, and user administration. Two such tools, namely CoPe_it! and eLogbook, were recently developed in the context of a large-scale, three years long, EU funded research project, which aimed at facilitating and augmenting individual and organizational learning in Communities of Practice (CoPs). These services have thoroughly exploited Web 2.0 technologies, were designed and developed from scratch, are interoperable and extensible, and have been validated in CoPs of diverse contexts.

Specifically, CoPe_it! is a tool of the Web 2.0 era, which aims at assisting and enhancing collaboration activities held among members of diverse types of communities (Karacapilidis & Tzagarakis, 2007). It provides a cognitive argumentation environment that stimulates reflection and discussion among participants, giving emphasis on the provision of various visualizations (views and projections) of the argumentation discourses, and on the use of reasoning mechanisms in more formal representations of the collaborative workspace. Users may join or create new communities, and collaborate in different (public or private) workspaces. The tool enables CoPs' members to better organize and depict their points of view in an ongoing knowledge exchange and knowledge creation process. Emphasis is put on allowing members of communities to assert and elaborate their arguments in varying levels of formality: users may start with an informal (human understandable) assertion of their positions and arguments, while they are then able to refine, structure and link them with those of their peers, using appropriate semantics (which are also interpretable by the system). This last feature enables the exploitation of advanced decision making support mechanisms.

eLogbook is a Web 2.0 social software application that aims at sustaining collaboration and learning in CoPs (El Helou et al., 2009). It offers community members a networking and communication platform, a repository for sharing and managing resources, a task and activity management system, as well as a community structuring tool allowing defining roles and distributing tasks. It also provides different types of notifications (via email or RSS feeds) in order to motivate contribution and sustain collaboration.

Both of the above tools were developed through a participatory design (PD) approach (Zeiliger et al., 2008), which may be considered as a process of negotiation of usefulness to be achieved through reconciling the contrasting perspectives of various stakeholders, including users, designers and others (Abreu de Paula, 2004). This alignment of actors' interests (Latour, 1999; Law, 1992) differs from the traditional system development along crucial dimensions: there can be no strict top-down control over such a collection of things (Monteiro, 2000); the notion of participation has to be extended to take into account the influence of non-human actors, such as artifacts and organizations; the role of human actors as mediators of other actors (technical or organizational) is also considered. Enrolling an actor in Participatory Design requires going through participative activities where actors can discover and share their common interests (Ehn, 2003). Participatory design was carried out in teams consisting of CoPs members, CoPs mediators (representing a CoP in the development process), service mediators (representing the developer team in the development process), software developers, and scientists from the fields of education and pedagogy. The authors of this paper were involved in the above teams with various roles: some as software developers, others as CoP and/or service mediators. As one of the main project's objectives was to introduce Web 2.0 concepts into the realm of CoPs, both tools made use of the relevant technologies and paradigms to deliver collaboration services to CoP members.

While - at a first look - it is straightforward to see the importance of the Web 2.0 principles for CoPs, practical matters are not without impediments and difficulties. In particular, engineering questions such as how to design, implement and instantiate the Web 2.0 principles, which existing technologies or architectures to use and how to integrate them into the CoP's workflow are not easy to address. The lack of methodologies, evaluation frameworks and the absence of detailed recommendations, instructions and "best practices" on how to use this new technology to address collaboration concerns in specific contexts complicate the tasks of developers (Annett & Stroulia, 2008).

2. Anatomy of CoPs and their needs

In this paper, we report on the experiences gained while developing CoPe_it! and eLogbook in the context of the Palette project (<http://palette.ercim.org>). These experiences are presented in the form of lessons that resulted out of the continuous live interaction in the participatory design teams and documented using evaluation studies based on questionnaires, interviews and discussions in teams, as well as indirect observation of the interaction in CoPs, thanks to the analysis of log files. The overall aim is to contribute to the efforts of establishing an agenda of best practices for developing Web 2.0 applications for CoPs or other types of online communities.

It is noted here that the Palette project aimed at facilitating and augmenting individual and organizational learning in CoPs. Towards this aim, an interoperable and extensible set of innovative services - as well as a set of specific scenarios of use - were designed, implemented and thoroughly validated in CoPs of diverse contexts. These services and scenarios support: (i) incremental convergence towards a comprehensive representation of practices; (ii) argumentative debates about practices; (iii) enhancement of practices through knowledge exploration, inside and outside CoPs; (iv) provision of procedures for the reification and creation of new practices.

CoPe_it! and eLogbook were intended to be used by thirteen CoPs associated with the project. These communities existed before the beginning of the project, while some of them were successfully operating for over ten years. These CoPs were active in various domains including education, engineering, information technology and entrepreneurship, and were distributed across Europe. CoPs varied greatly in their number of members: small CoPs had around twenty, while large ones featured a few hundreds members. All these CoPs were already using a number of Web-based and desktop tools to augment their activities, including online discussion forums, Wikis, office automation tools and e-mail. Generally speaking, these tools fell short to respond to a number of collaboration needs of these communities. More clearly, the development of CoPe_it! was motivated by the limited (or absence of) decision making and knowledge management functionalities of the traditionally used applications. Similarly, the development of eLogbook was motivated by the fact that the tools already used by these CoPs had limited or no support for task management and poor awareness and notification features.

3. Lessons learned

The experiences reported in this paper are related to two main areas: (i) the methodology and process of developing collaboration services (lessons 1-5) and (ii) the collaboration concept *per se* (lessons 6-8). In this section, we report on the actual lessons learned during the development of the abovementioned two collaboration support tools. For each lesson, we describe its context and discuss characteristic instances from various stages of the project's development.

Lesson 1: To make PD effective, the development team should be considered as an online community with its own mediator, namely the service mediator, able to negotiate with the CoP mediator. For Distributed Participatory Design (DPD) to be effective, collaboration tools such as wikis and synchronous communication tools are essential as they help in harvesting and explicating design- related considerations.

From the beginning of the project, CoP representatives worked closely with service mediators. Teams of developers and CoP representatives were formed in order to work closely and negotiate the different features that would be offered or not (end-users were continuously and actively involved into the design process, being considered as an integral part of the development team). Once the teams were set up, the question of how to coordinate the collaboration among all participants came up. To address this issue, two main ideas were found useful: one was to use the notion of scenario as a cornerstone of the participation process; the second one was to conceive the development team as a multidisciplinary online community. The design of scenarios went through several steps of de-construction and re-construction (Esnault et al., 2009), which supported the synchronous participation of all actors (users and developers) during the whole process.

Lesson 2: A scenario-based PD approach enables an efficient participation of actors and mediators in collaborative activities that enable and sustain the design and development of useful and usable tools supporting the development of the evolving activities of users.

Participatory design in this project led to a boundary process of scenarios building. A scenario is the description of a set of activities and actions, supported by specific services and common support services in order to achieve an intention. The intentions taken into account are those that concern mainly a CoP's life: knowledge reification and document management, debate and decision making, facilitation and animation of the CoP's life. Scenarios are tools for envisioning the future. They convey stories that happen in the real world, as well as stories we imagine happening in possible worlds. According to (Carroll, 1995), scenarios describe key situations of use, in terms of actors, goals, context, tools, actions and events. Valuable aspects of scenarios include that they do not come with a strong semantic; they require that their semantic be constructed. This scenario-based evolution enabled most stakeholders to participate and contribute.

Another important aspect of scenario descriptions is that - in a PD process - most stakeholders would understand them, even though they shed different perspectives on them. Scenarios are thinking tools; they are not requirements – they are deliberately incomplete and easily revised; they facilitate the innovative exploration of design possibilities; they are more than technological capabilities; they help developers coordinate design action and reflection; they help designers manage tradeoffs; they are both concrete and flexible; they help focus interaction among stakeholders in a PD process, by enabling multiple levels of details and multiple perspectives; finally, they afford multiple views of an interaction, thus helping developers to manage the consequences of design change (Carroll, 1995).

In the final stages of the project, the scenarios appeared as real "cornerstones": they were referred to equally in the different communities of the project; they were used as reference to classify other elements of productions; they were used in CoPs to organize their activities. This approach greatly improved the collaboration and coordination of the work within teams. It permitted the gathering of the necessary requirements, as well as the discussion and review of design decisions. In addition, the resources created in the course of the collaboration act as a project memory capturing the design rationale of features. In the same way, as soon as a first version of eLogbook was put online, the idea of forming a community around the tool for its design, evaluation and continuous evolution was pushed further. CoP mediators and members as well as tools' evaluators were encouraged to use the tool itself to express their satisfaction or dissatisfaction with respect to the usefulness and usability of its features. This is how one project member, whose task was to study the eLogbook usability, created a list of

reported bugs and questions within a dedicated space and used the eLogbook chat feature to discuss the different issues with eLogbook developers. This helped her to get acquainted with the tool, while simultaneously discussing its pros and cons.

Lesson 3: Tools should release their features frequently and incrementally. Yet, incremental changes introduce the problem that new functionalities may go unnoticed and efforts should be put in making the use of new functionalities apparent to CoPs.

Throughout the project, CoPe_it! and eLogbook released or improved their functionalities incrementally. CoPe_it! scheduled roughly two releases per year along with introducing or improving functionalities in between. Each release was announced in advance to all CoPs and the new functionalities were documented on the service's support Web site. eLogbook pushed further this release paradigm by continuously updating the public online version every time a new feature was added or an existing one was improved. The frequent releases gave the opportunity to all CoP members to comment on the existing features and to propose new ones, which could augment their collaboration, as the appropriation of a new service in part of the identity building of a CoP. For example, by early releasing new versions of CoPe_it!, feedback of users identified the need to easily reference resources that were brought into the workspace, in order to reuse them across different workspaces. The design team of CoPe_it! did not initially consider this feature, but its usefulness to the CoPs led to the implementation of a REST-based service for identifying any individual resource within CoPe_it!. As such shortcomings were very early discovered in the process, the cost of adding and implementing them was low, without impacting the development plan of CoPe_it!.

The frequent and consistent releases of CoPe_it! and eLogbook had another profound impact on the attitude of users towards the tools: it gave the impression of tools that are “alive”, constantly evolving and improving, which stimulated the user's interest on how to use them and what features the next release would have. This can be considered as an additional awareness feature. This interest was documented by users' messages that asked to get informed whenever a new version or a new feature was made available. CoPs members were establishing a personal relationship with the tools and were interested in their evolution.

While it was rather easy to notify users whenever new features were introduced or improved, it was rather difficult to inform CoP members on their intended use. The interviews and log-file analysis not only showed that features introduced gradually were less frequently used than features that were initially available but also that they were used in ways unanticipated by the developers. Simply announcing new features on the tool's support site or via email, by providing a short textual description of the service, proved to be an ineffective way in communicating the role of each feature. To improve on this situation in CoPe_it!, two changes were made. First, announcements on the support site were accompanied with videos animating the use of the new functionalities and tutorials presenting typical scenarios where these are useful. Second, in order to make users aware of the published material on the support site, upon login, a splash screen provided links to the related material (video, tutorial) whenever new functionality was available. Preliminary analysis of log files confirmed the obvious consequence that access to the support material would increase. Yet, this increase in accesses did not coincide with an increase and steady use of the newly introduced functionalities. Moreover, many users found the splash window rather annoying and some reported to be confused about the window's purpose. For Web 2.0 applications, such findings raise the issue of how to properly inform users whenever new functionality is available,

making them aware on the intended use as approaches adopted by traditional applications fall short in their case.

Lesson 4: Support for tracing users' actions and the system's behavior (i.e. logging) should be a first concern when developing collaboration tools and should be put into use early on.

Although logging the actions of users during their use of a service is in general conceived as an important feature of any tool, it is nevertheless a feature that is typically introduced rather late in the development process. Yet, being able to fully log all user actions even in very early releases of the tools is crucial for the assessment of existing functionality and the design of new ones. This is especially true for collaboration support software, as some collaborative features, such as how people share and exchange, are difficult to be assessed in a live or 'in vivo' setting. The development of CoPe_it! and eLogbook showed that log files provide valuable feedback for the improvement of the collaborative services of the tools.

Initially CoPe_it! made limited use of logging facilities which were mainly used as a mechanism to check the correct execution of the tool i.e. to catch errors and software malfunctions. Such use of logging facilities made it difficult to gain insights on issues such as which features were used most, or how exactly the available features were used. In addition, this setting made it also difficult to spot potential performance problems, which for CoPe_it! that utilizes AJAX technologies was a critical factor. Subsequent changes to the tool enabled the logging of all user actions along with additional information such as their completion time, which gave valuable insights on how to improve its features. In particular, logging all user actions revealed operations that executed slowly, and which proved to hinder the collaboration of users as they were unable to attain smooth and unobtrusive interaction.

Lesson 5: Design of collaboration tools should start from the user interface and visualizations.

Traditional development methodologies require the design of applications to start from the back-end (the so-called "infrastructure" of applications) that include elements such as the database and domain specific services, and proceed then to the design of the user interface. Such way of developing applications usually lead to big and complex infrastructures, where the user interface is tightly bound to the back-end and hence difficult to change. When developing Web 2.0 for CoPs, a more flexible approach is to start the design from the user interface and the corresponding visualizations. Our experience during the development of the two collaboration tools shows that tackling the development in such a way not only addresses better the needs of CoPs, but also leads to a much simpler infrastructure. It seems to be easier to break down the user interface into individual infrastructure services than to assemble and relate existing services to provide a particular user interface. In addition, user interface and visualization services should provide users with engaging environments.

Lesson 6: Innovative metaphors of collaboration, although useful, may confuse users and should be introduced in a way that is close to what users are expecting.

Whenever radical new and innovative metaphors to collaboration are provided to users, these must be carefully introduced, as there lurks the danger of tool rejection due to encountering new and unexplored territory. In general, when users get to use collaborative systems they expect (based on their experience) to fall on traditional categories of wikis, discussion forums

and tagging systems, as these are the prevailing systems nowadays on the Web. Our experience showed that radical new ways to collaboration are initially causing confusion rather than excitement.

In the case of CoPe_it!, for instance, initial evaluation feedback showed that the adoption of a spatial metaphor to facilitate discussions was not immediately grasped by users, and caused great confusion with respect to how to use the tool (Figure 1). To address such concerns, CoPe_it! made improvements on two fronts: first, it revised and enhanced its help and tutorials emphasizing on providing examples in order to convey its use. Second, it extended its functionality and included the ability to render the discussion in a way that is familiar to users. For example, CoPe_it! enabled a forum-like view of the discussion (called time-order-view in CoPe_it!) that displayed the discussion in a way that is found in traditional Web based discussion forums. This functionality proved very helpful as the spatial metaphor of workspaces (constituting CoPe_it! main way of supporting collaboration) was now regarded as simply another way of viewing and conducting the discussion, amongst others, with which the user was already acquainted to. This also enabled users to easily see the importance of conducting discussions spatially by comparing it to traditional ways to supporting discussions.

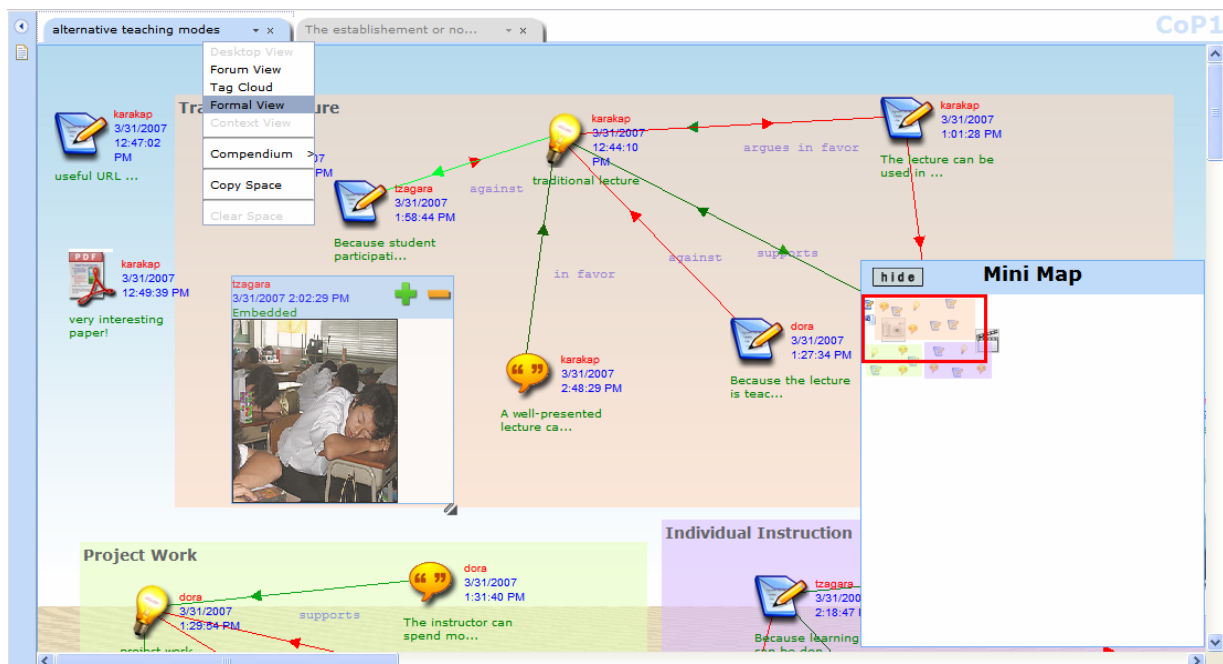


Figure 1: Spatial metaphor of a discussion

With respect to eLogbook, in its first releases, the interface consisted of a mirror of eLogbook 3A model influenced by the actor-network and activity theory. Even though applying a general theory appeared to be nice at the conceptual level, it ended up confusing the first eLogbook users. As a matter of fact, they were expecting familiar labels such as “group”, “community”, and “community space” and found all these constructs or labels fused into one general term “activity”. The initial idea was to focus on the objective of a collaboration space rather than on the space itself. So the community members would be gathered in one mother activity which objective was nothing but the reason behind the creation of the community. In turn, this mother activity can consist of different activities corresponding to different projects within the community. Nevertheless, at the presentation level, this naming wasn’t suitable

and had to be replaced by more familiar and concrete terms such as “collaboration space”. Moreover, the tools acceptability increased when users were allowed to distinguish different collaboration spaces, define new ones and reuse the ones created by others like “communities of practice”, “group”, “group of interest”, “team”, “theatrical club”, “committee” so on and so forth.

Lesson 7: Formality in collaboration management should not be considered as a predefined and rigid property, but rather as an adaptable aspect that can be modified to meet the needs of the tasks at hand.

Generally speaking, when engaged in the use of existing technologies and systems supporting collaboration, users have to follow a specific formalism. More specifically, their interaction is regulated by procedures that prescribe and - at the same time - constrain their work. This may refer to both the system-supported actions a user may perform (e.g. types of discourse or collaboration acts), and the system-supported types of collaboration objects (e.g. one has to strictly characterize a collaboration object as an idea or a position). In many cases, users have also to fine-tune, align, amend or even fully change their usual way of collaborating in order to be able to exploit the system’s features and functionalities. Such formalisms are necessary towards making the system interpret and reason about human actions (and the associated resources), thus offering advanced computational services. However, there is much evidence that sophisticated approaches and techniques often resulted in failures (Shipman & McCall., 1994). This is often due to the extra time and effort that users need to spend in order to get acquainted with the system, the associated disruption of the users’ usual workflow, as well as to the “error prone and difficult to correct when done wrong” character of formal approaches.

Incremental formalization of collaboration, which was adopted in the development of CoPe_it!, proved to be a successful approach to address the above concerns. In this approach, formality and the level of knowledge structuring is not considered as a predefined and rigid property, but rather as an adaptable aspect that can be modified to meet the needs of the tasks at hand. By the term formality, we refer to the rules enforced by the system, with which all user actions must comply. Allowing formality to vary within the collaboration space, incremental formalization, i.e. a stepwise and controlled evolution from a mere collection of individual ideas and resources to the production of highly contextualized and interrelated knowledge artifacts, can be achieved.

Lesson 8: Collaboration services do not operate in isolation and should not be regarded as “application islands”. Data and services accessibility through seamless interoperability with existing tools is a crucial factor for their adoption and success.

From the users’ initial needs and ongoing feedback, openness and seamless interoperability appears to be a primordial need and constitute one of the biggest challenges of today’s social software applications. Users want to gain “real ownership” over the information that they have provided and/or that belongs to them (e.g. their profile information, projects, and friends). They want to be able to easily import/export from one environment to another. They want to be able to synchronize information across different tools and visualize it in different ways via different applications. Moreover, the learning and knowledge management processes in CoPs always exhibit stages that typically require different sets of tools. A seamless integration of distributed tools and services is instrumental for providing of new collaboration solutions. As a plethora of resources are already available on the Web and used by CoP members during their day-to-day tasks, collaboration services must explicitly address

issues regarding the integration of these resources into their environments. Otherwise, the danger of becoming isolated may surface and ultimately lead to their rejection.

To respond to the data openness and portability need, different useful interoperability scenarios were designed and implemented. For instance, the possibility to export or propagate profile updates from eLogbook to CoPe_it! is now possible. In the same way, the CoPe_it! profile can be imported to eLogbook. The implementation of OpenID in eLogbook is also intended to respond to the need of having one identity across different services. Moreover, the ability to visualize Facebook friends within eLogbook was included and to combine a Facebook user and an eLogbook user into one single entity (Figure 2). The final goal was to supply a user with means to manage different social software accounts in one place and have a login-transparent access to different applications.

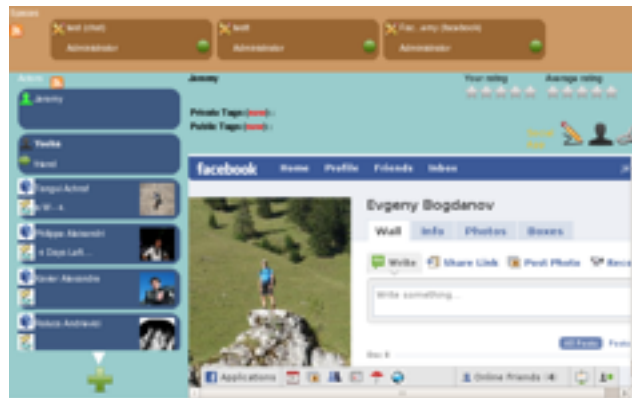


Figure 2: eLogbook with Facebook friends integration

In CoPe_it!, the efforts focused on giving the ability to integrate resources from two popular and relevant systems: Compendium (Shum et al., 1993) a tool to capture design rationale and Web-based discussion forums. In addition, every resource in CoPe_it! or eLogbook was given the ability to be referenced via a unique and simple URI that enables their use in other applications.

4. Conclusions

Although a plethora of Web 2.0 libraries, tools and paradigms are available, there is lack of reports on how these new means can be used to address the needs of communities. Towards filling this gap, we reported a series of practical lessons learned in the framework of a large-scale European research project. These lessons resulted out of the development of two collaboration support tools, namely CoPe_it! and eLogbook. The lessons reported were relevant to two issues: the process of developing the tools and the support of the collaboration. The overall aim of this work is to contribute to the agenda of assembling a body of experiences related to developing Web 2.0 tools, which can lead to the establishment of best practices when building such tools for CoPs or any other type of online communities.

References

Abreu de Paula, R. (2004) *The Construction of Usefulness: how Users and Context Create Meaning with a Social Networking System*. Doctoral Thesis. University of Colorado at Boulder.

- Annett, M. and Stroulia, E. (2008) *Building highly-interactive, data-intensive, REST applications: the Invenio experience*. In Proceedings of the 2008 Conference of the Center For Advanced Studies on Collaborative Research: Meeting of Minds (Ontario, Canada, October 27 - 30, 2008).
- Carroll, J. (1995) *Scenario-based design: envisioning work and technology in system development* John Wiley and son, NY, USA.
- Ehn, P. (2003) *Participation in Interaction Design – actors and artefacts in interaction*. In Invited paper to the International Symposium “Foundations of Interaction Design”. Interaction Design Institute, Ivrea, Italy, November 12-13, 2003.
- El Helou S., Gillet D., Salzmann C., and Rekik. Y. (2009) *Social Software for Sustaining Interaction, Collaboration and Learning in Communities of Practice*. In Solutions and Innovations in Web-Based Technologies for Augmented Learning: Improved Platforms, Tools, and Applications, Advances in Web-based Learning (AWBL) Book Series, pages 300-316. Information Science Reference.
- Esnault, L., Daele, A., Zeiliger, R., Charlier, B. (2009) *Creating an Innovative Palette of Services for Communities of Practice with Participatory Design*, in Proceedings of the EC Tel 09 conference, Nice, 1-2 October 2009, Springer-Verlag
- Karacapilidis, N. and Tzagarakis, M. (2007) *Supporting Incremental Formalization in Collaborative Learning Environments*. In E. Duval, R. Klamma and M. Wolpers (eds.), Proceedings of the 2nd European Conference on Technology Enhanced Learning (ECTEL 2007), Crete, Greece, September 17-20, Springer-Verlag, Berlin, Lecture Notes in CS, Vol. 4753, (2007), 127–142.
- Latour, B., (1999) *On Recalling ANT, in Actor network Theory and After*, John Law and John Hassard editors; Blackwell Publishing, Oxford, UK.
- Law, J., (1992) *Notes on the Theory of the Actor-Network: ordering, strategy, and heterogeneity*, Systems Practise, 5(4), pp. 379-393.
- Monteiro, Eric, (2000) *Actor-network theory*. In: C. Ciborra (ed.), From Control to Drift. The Dynamics of Corporate Information Infrastructure, Oxford Univ. Press, 2000, pp. 71 – 83
- Shipman, F.M. and McCall, R. (1994) “*Supporting knowledge-base evolution with incremental formalization*”, Proc. CHI '94 Conference, 1994, pp. 285-291
- Shum, S., MacLean, A. Forder, J. and Hammond, N. (1993). *Summarising the Evolution of Design Concepts within a Design Rationale Framework*. In Adjunct Proceedings of InterCHI'93: ACM/IFIP Conference on Human Factors in Computing Systems, April 24-29, Amsterdam, pp. 43-44.
- Zeiliger, R., Esnault, L. Vermeulin, F., Cherchem, N., (2008) *Experiencing Pitfalls in the Participatory Design of Social Computing Services*, in Proceedings of the Participatory Design Conference 2008, Bloomington, IN (USA), October 1-4, 2008