Adding Social Awareness to Jazz for Reducing Socio-Cultural Distance between Distributed Development Teams

Fabio Calefato, Domenico Gendarmi, Filippo Lanubile
Dipartimento di Informatica, Università degli Studi di Bari, via E. Orabona 4
70125 Bari, Italy
{calefato, gendarmi, lanubile}@di.uniba.it

Abstract. A Collaborative Development Environment (CDE) provides a shared workspace with a standardized toolset that helps distributed development teams cope with geographical distance. However, CDEs lack any support to reduce socio-cultural distance, which poses practical barriers to the development of connections and shared culture in distributed settings. The recent rise of the Social Web created several opportunities to publish personal information and develop connections from a distance. We argue that disseminating additional social awareness information to developers, who have little or no chances to meet, can help to speed up the establishment of organizational values, attitudes, and trust-based inter-personal connections. In this paper, building on existing literature, we first propose our definitions of three distinct types of awareness. Then, by means of scenarios, we show how our extension of the Jazz, a CDE that already provides presence and workspace awareness, adds social awareness information about coworkers, i.e., interests, emotional state, in order to reduce socio-cultural distance, and improve team openness and well being in distributed settings.

Keywords: Group Awareness, Social Awareness, Social Web, Web 2.0, Mashup, Jazz, Eclipse, Collaborative Development Environment, CDE.

1 Introduction

In distributed settings, due to distance, software teams often rely on Collaborative Development Environment (CDEs), such as SourceForge[1], GoogleCode[2], Github[3], and Assembla[4], which are an integrated and flexible set of tools (e.g., code compiler and debugger, version control, issue tracking) that help distributed teams control their software development process. Yet, despite their ability to cope with geographical distance, CDEs provide little support to reduce socio-cultural distance. In fact, differences in culture, which can be intuitively epitomized as a fuzzy set of attitudes,

[1] http://sourceforge.net/
[2] http://code.google.com
[3] http://github.com/
[4] http://www.assembla.com/

beliefs, behavioral norms, basic assumptions and values that are shared by a group of people [21], pose practical barriers to the development of relationships and connections (i.e., common ground, mutual confidence, trust) within distributed teams, with a potential severe impact on project management effectiveness.

The idea of applying social software to help distributed teams deal with socio-cultural distance is rather recent. The rise of the Social Web, also known as Web 2.0 [5], created lots of sources for personal, user-generated content, and opportunities to develop connections from a distance. In [1] we presented our initial prototype, which embeds information collected from social websites (e.g., Twitter[5], Facebook[6], Last.fm[7]) into the IBM Jazz CDE. We used the Jazz CDE, because it already provides both presence and workspace awareness, and leveraged the FriendFeed aggregator service to embed personal information about distributed co-workers, collected from social networks. Here, building on relevant literature, we first propose our own defintions of the existing types of awareness. Then, after showing the features or our Jazz extension, we present some key usage scenarios to illustrate how providing distributed software teams with overall group awareness (i.e., presence, workspace, and social awareness) aggregated in one place can help to speed up the establishment of organizational values, attitudes, and trust-based personal connections between distant team members, with little or no chances to meet.

The remainder of this paper is organized as follows. In Section 2, we present an overview on CDEs and Jazz, in particular. In Section 3, building on relevant previous works, we propose our definitions of different awareness types. In Section 4 we first discuss the rise of Social Web applications in general, and then we present FriendFeed as a particular example of Web Mashup. Our Jazz extension is discussed in detail in Section 5, whereas in Section 6 we illustrate the usage scenarios. Finally, we conclude in Section 7.

2 Collaborative Development Environments

In software development, control is the process of adhering to goals, policies, standards, and quality levels, set either formally (e.g., formal meetings, plans, explicit guidelines) or informally (e.g., team culture, peer pressure). Because in distributed settings it is not possible to control units by walking, organizations had to fall back to using collaborative tools to control the software development process from a distance. Collaborative Development Environments (CDEs), such as SourceForge, Gforge, Google Code, are the most used and full-featured process-aware tools to support distributed teams.

CDEs were envisioned by Booch & Brown, who first acknowledged the need for 'frictionless surface' in development environments [4], motivated by the observation that much of the developers' effort is wasted in switching back and forth between different applications to communicate and work together. According to this vision, collaborative features should be available as components that extend core applications

[5] http://twitter.com/
[6] http://www.facebook.com/
[7] http://www.lastfm.it/

(e.g., the IDE), thus increasing the users' comfort and productivity. Therefore, CDEs support developers by incorporating the standard toolset needed (e.g., compiler, debugger, version control system, bug tracker) within a single project workspace, reducing the effort of running multiple different applications to communicate and work together.

Earliest CDE were developed within open source software (OSS) projects because OSS projects, from the beginning, have been composed of dispersed individuals. Today a number of CDEs are also available as commercial products.

2.1 IBM Jazz CDE

Jazz [8] is one of the most noticeable commercial CDE because it can be customized to support any process. Besides, Jazz is an extensible platform, which leverages the Eclipse's notion of plugins to build CDE products. The present version has a wide-ranging scope, but in the former version of Jazz the goal was to integrate synchronous communication and reciprocal awareness of coding tasks into the Eclipse IDE, following Booch & Brown's vision.

Jazz is an extensible team collaboration platform based on a client-server architecture, which integrates many different technologies in a single environment. The Jazz server hosts a set of services (e.g., generate reports, resolve work items from the Web) and houses data in its repository (e.g., configurations, source code). Remote clients communicate with the Jazz server over the network, using SOAP/XML over HTTP (Fig. 1). The full-featured Jazz client is Rational Team Concert, an extension of the Eclipse IDE, packed with all the plugins necessary to the Jazz development platform. It provides presence awareness, thanks to the integration with Lotus Sametime, and workspace awareness, by generating an RSS feed of all project-related events occurring the workspace.

The essential components of Jazz are the Repository and Team Process, which represent the platform kernel and are developed by the Jazz Project Member (Fig. 2). Other members of the Jazz community develop additional components to add new capabilities to Jazz, such as source control and reporting. While the Team Process component is meant to make Jazz a customizable, process-aware platform, Repository allows to store tool-specific information in a central place where it can be made available to all other components in all client and server configurations. Thus, the Repository plays a key role in Jazz extensibility.

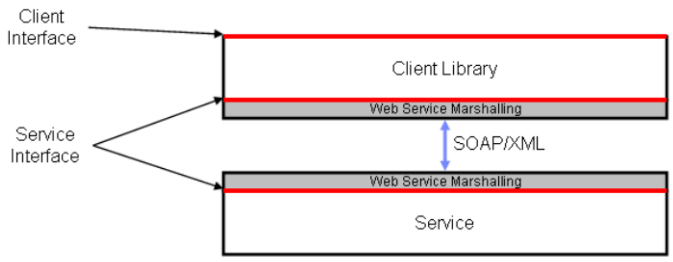


Fig. 1. Jazz Client-Server Communication [11]

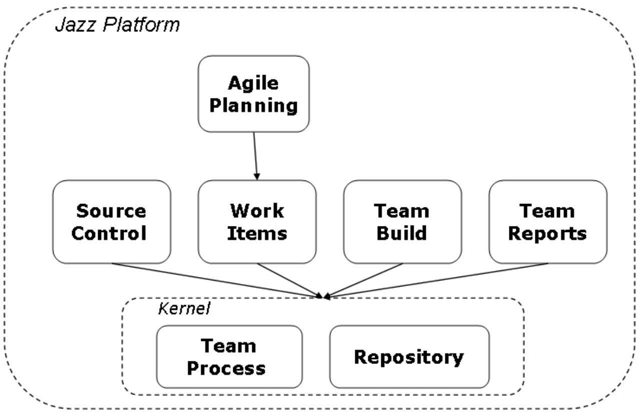


Fig. 2. Components of the Jazz Platform [11]

3 Types of Awareness: Definitions

Dourish and Bellotti were among the first to define the concept of awareness as an understanding of the activities of others, which provides a context for one's activity, so that individual contributions are relevant to the group's activity as a whole [7]. The concept of awareness is strictly connected to the activities of displaying and monitoring of information [16]. When performing a shared, collaborative activity, *displaying* refers to the notification of one's information (e.g., presence and actions, typically) that can be relevant to others involved. *Monitoring* is the complement of information displaying, as it refers to the peripheral observation of others in an unobtrusive way, in order to avoid interruptions.

Being a complex, information-intensive, and highly collaborative activity, software development can greatly benefit from awareness, especially in distributed settings, where teammates have to collaborate from a distance. Ko et al. [13] conducted a study to understand the information needs in software developments teams, showing that the most frequently sought information included awareness about tasks, artifacts, and co-workers. We have identified three major types of awareness: presence, workspace, and social.

Presence awareness is the awareness of what distant colleagues are doing, their availability for interaction, and how they prefer to be reached, helping coworkers to minimize interruptions and disturbances when engaging in collaborative processes [9]. Presence awareness has almost become synonymous with IM and VoIP because such tools represent the preferred, lightweight means to broadcast information or questions, as well as ascertain and negotiate availability to accommodate opportunistic interaction between co-workers.

Workspace awareness means knowing project teams and their internal structure, as well as team members and artifacts. Tools such as Palantìr [18], Hipikat [6], and Mylyn [12] provide developers with workspace awareness information that helps developers to identify other teammates, artifacts, and tasks that are related to the artifact/task at hand. Workspace awareness is particularly relevant to project managers and team leads as it helps to track the state of a project.

Social awareness is the awareness about interests, opinions, and emotional state of members of a group, which can be extremely beneficial to increase the sense of "teamness" in distributed software settings [16]. Social awareness has been acknowledged only recently, and unlike presence and workspace awareness, it cannot be directly considered contextual to a software development project. Nonetheless, since it helps to develop an organizational culture and to consolidate connections and trust-based relationships between distant collaborators, social awareness contributes to project success by improving team's well being and social health [18]. There are very few software development-oriented tools that support social awareness. One of the most noticeable is Github[8], a software repository that combines standard features of social networking sites (e.g., following or messaging developers, watch projects' activity timeline through feeds) with Git, a distributed source-control system. Codebook [3], instead, is a Microsoft prototype that aims at developing a social networking services over code, in which people can also be friends with the artifacts they share.

Because presence, workspace, and social awareness provide an answer to specific requests of information, if aggregated in one place they can help teams maintain an overall *group awareness*. A recent research study by Omoronya has shown that tool support for distributed software development teams are still inadequate in enhancing distributed awareness because most tools are designed to support a specific kind of awareness in isolation [16]. To date no tool has provided distributed development teams with support to group awareness. IBM Jazz, discussed in the next section, is one of the most recent and full-featured Collaborative Development Environments, which provides presence and workspace awareness in one place, but lacks any support to social awareness.

4 Social Web

Recently, the rise of the Social Web [5], created new incentives and motivations for publishing personal information on the Web. Nowadays, plenty of user-generated content is public available. Applications like Wikipedia[9], Flickr[10], Delicious[11], LibraryThing[12], for example, provide each day new wiki articles, photos, bookmarks and book reviews, as well as new metadata, which are directly added by users.

However, these forms of collaborative contributions are restricted to one single application and current Social Web applications are isolated from one another, like 'walled gardens'. The main reason for this lack of interoperation is that for the most

[8] http://github.com/
[9] http://www.wikipedia.org/
[10] http://www.flickr.com/
[11] http://delicious.com/
[12] http://www.librarything.com/

part in the Social Web, applications' owners are quite reluctant to provide programmatic access to user generated content, which is hosted within their web sites.

In such a context, Web mashups have emerged providing a dynamic approach to compose content and functionalities originating from disparate web sources [23]. Among the different classes of mashup available on the Web, we focused on News Mashups [15] also known as Syndication Feed Mashups since they use syndication technologies like RSS and Atom to aggregate news related to various topics and create personalized feed views. In particular, we envisioned the opportunity to aggregate feeds about personal information originating from social networking sites in order to foster group awareness within distributed teams.

4.1 FriendFeed, a Social Information Aggregator

FriendFeed[13] is a real-time feed aggregator that consolidates the updates from a number of social networking websites (e.g., Delicious, YouTube[14], Last.fm, LinkedIn[15], Facebook), as well as any other custom website providing an RSS/Atom feed. Thus, FriendFeed users can use this stream of information to create customized feeds to share (and comment) with friends. The main reason of FriendFeed success is that it provides the facility to track users' activities (such as posting on blogs, Twitter and Flickr, or listening to music on Pandora[16]) across a broad range of different social networks, whereas other services exclusively facilitate tracking of their own members' activities on their particular social service.

What looked attractive to us about exploiting the FriendFeed service into Jazz is that: (1) a free API is available for leveraging the service output into third-party applications; (2) private groups can be defined so that updates from members are bound within group and not visible to users on the outside; (3) users can decide what is relevant or appropriate to stream and share, reducing information privacy and overloading concerns.

Members of the same Jazz project-area can create a FriendFeed private groups where they can choose which feeds they want to share and who can see the shared feeds. They can also start a conversation around shared items, or just show that they like a feed someone else has shared. By aggregating in the same place different feeds from social networking services used on a daily basis by the project members, we can thus foster the discovering and discussion of personal information regarding people within the project-area, speeding up the development of connections and shared context between team members.

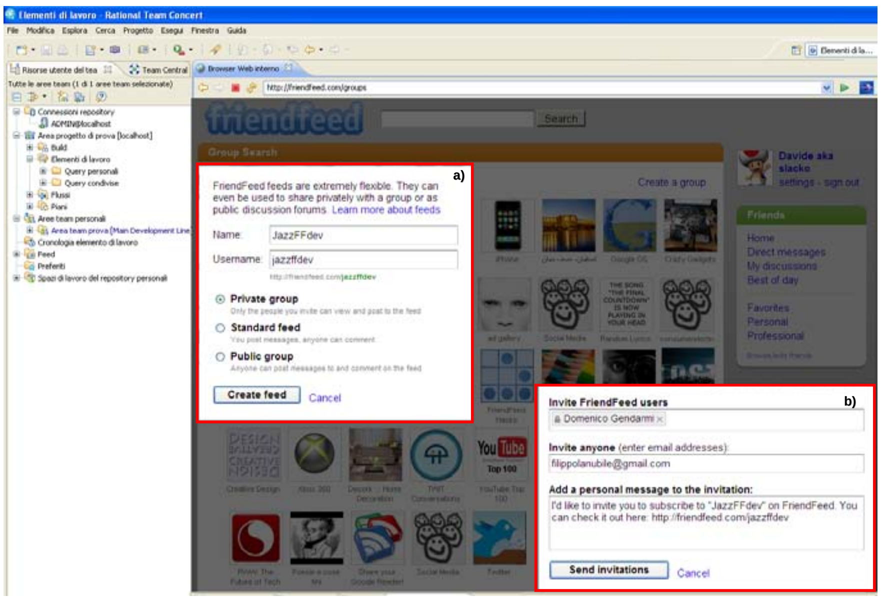
[13] http://friendfeed.com/
[14] http://www.youtube.com/
[15] http://www.linkedin.com/
[16] http://www.pandora.com/

5 The Jazz Client Extension

The Jazz client extension was coded using a Java wrapper of the FriendFeed API. Unfortunately, the API itself does not allow doing as much as desired. For instance, groups cannot be created or managed using the API, which basically allows getting group description and members, reading the feed, and posting messages. In the remainder of this section, we briefly illustrate how the extension works in a simple scenario.

Using the FriendFeed web interface, the project-lead of a Jazz project-area creates a private group for the project. In order to avoid the friction of switching to the web browser and then back to the Jazz client, a browser window can be opened in one of the workspace views to complete group creation (see Fig. 3a).

Other developers needs to be invited to join the group, using either their FriendFeed usernames, in case they are already registered, or just emails (see Fig. 3b). Because the group is private, developers will have to enter their FriendFeed credentials to subscribe the aggregated feed and post contribution. Using the Eclipse Modeling Framework, we extended the project-area model in order to store FriendFeed credentials about both users and groups into the Jazz server repository. This way, even when changing the machine where the Jazz client is run (e.g., in case of switching from the office desktop to the laptop for a business travel), the group feed is available as long as the extension is installed on the client.


Elementi di lavoro - Rational Team Concert
Browser Web interno
http://friendfeed.com/groups
friendfeed
Group Search
a)
b)
Create a group
Friends

FriendFeed feeds are extremely flexible. They can even be used to share privately with a group or as public discussion forums. Learn more about feeds

Name: JazzFFdev

Username: jazzffdev

http://friendfeed.com/jazzffdev

[✓ Private group] Only the people you invite can view and post to the feed

Standard feed You post messages, anyone can comment

Public group Anyone can post messages to and comment on the feed

Create feed Cancel

Invite FriendFeed users: Domenico Gendarmi

Invite anyone (enter email addresses): filippolanubile@gmail.com

Add a personal message to the invitation: I'd like to invite you to subscribe to "JazzFFdev" on FriendFeed. You can check it out here: http://friendfeed.com/jazzffdev

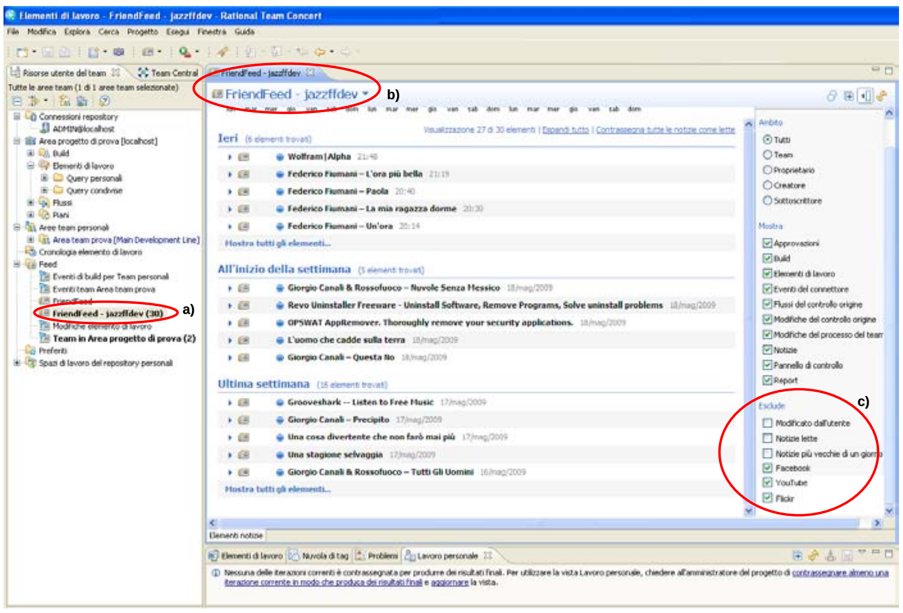
Send invitations Cancel

Fig. 3. FriendFeed group creation *a)* Friend Invitation to the group *b)*

Developers can be invited to the group as regular members or admins: Only in the latter case they are allowed to add a service to the aggregated feed. As a group admin, each developer is allowed to specify what personal information collected from external social web sites can be streamed to the group, without violating his/her privacy. For instance one developer can choose to stream what he likes on Last.fm and Pandora, whereas another one can share her bookmarks saved in Delicious and the book reviews she posted on LibraryThing.

Once the group is created, the project-lead can register it with the project area in Jazz by entering the URL of the aggregated feed. Upon registering the group, the associated feed appears in the Team Artifact view of Jazz, under the Feed folder (see Fig. 4a), along with the default feeds that provide workspace awareness by informing developers about development-related events (e.g., commits, build failures) or any other change occurring in the workspace (e.g., changes to the milestones release date). Hence, the aggregated feed of personal information is visualized using the same internal feed reader provided by Jazz (see Fig. 4b). Because source services can be added to the aggregated stream by each developer who subscribed the FriendFeed group as admin, the larger the development team, the higher the risk of information overload is. Hence, we extended the feed reader to give each developer the opportunity to filter the updates from undesired sources. For instance, one can choose to filter out the updates from Facebook, YouTube and Flickr, while displaying all the others available in the aggregated feed (see Fig. 4c).

Finally, an extra view has been made available in the workspace to let developers post message to the FriendFeed group directly from the Jazz client (see Fig. 5a). All subscribers to the group are then able to view new messages as feeds, through both the FriendFeed web-based interface and the feed reader included within the Jazz client (Fig. 5b).


Elementi di lavoro - FriendFeed - jazzffdev - Rational Team Concert
File Modifica Esplora Cerca Progetto Esegui Finestra Guida
Risorse utente del team
Team Central
FriendFeed - jazzffdev
a)
b)
c)
FriendFeed - jazzffdev (30)
Feed
Elementi notizie
Elementi di lavoro
Nuvola di tag
Problemi
Lavoro personale

FriendFeed - jazzffdev

Visualizzazione 27 di 30 elementi | Espandi tutto | Contrassegna tutte le notizie come lette

Ieri (6 elementi trovati)

- Wolfram|Alpha 21:48
- Federico Fiumani – L'ora più bella 21:19
- Federico Fiumani – Paola 20:40
- Federico Fiumani – La mia ragazza dorme 20:30
- Federico Fiumani – Un'ora 20:14

Mostra tutti gli elementi...

All'inizio della settimana (5 elementi trovati)

- Giorgio Canali & Rossofuoco – Nuvole Senza Messico 18/mag/2009
- Revo Uninstaller Freeware - Uninstall Software, Remove Programs, Solve uninstall problems 18/mag/2009
- OPSWAT AppRemover. Thoroughly remove your security applications. 18/mag/2009
- L'uomo che cadde sulla terra 18/mag/2009
- Giorgio Canali – Questa No 18/mag/2009

Ultima settimana (15 elementi trovati)

- Grooveshark -- Listen to Free Music 17/mag/2009
- Giorgio Canali – Precipito 17/mag/2009
- Una cosa divertente che non farò mai più 17/mag/2009
- Una stagione selvaggia 17/mag/2009
- Giorgio Canali & Rossofuoco – Tutti Gli Uomini 16/mag/2009

Mostra tutti gli elementi...

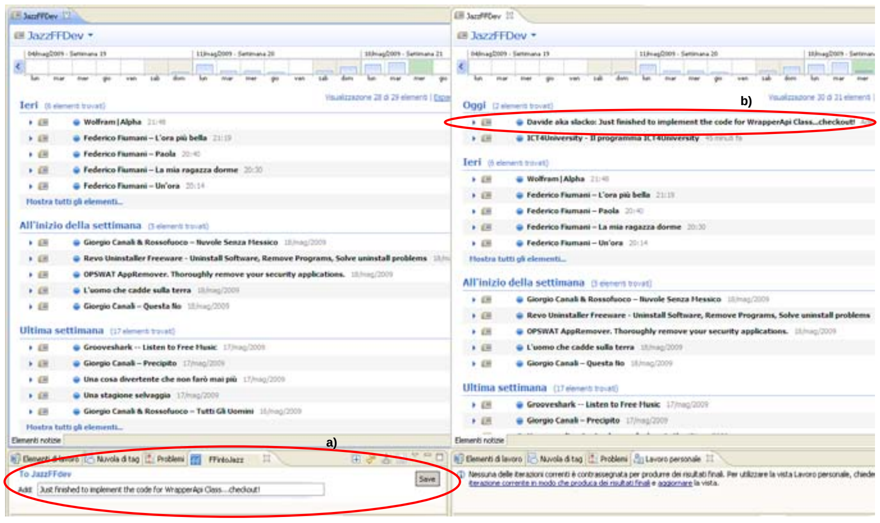
Ambito
- ● Tutti
- ○ Team
- ○ Proprietario
- ○ Creatore
- ○ Sottoscrittore

Mostra
- ☑ Approvazioni
- ☑ Build
- ☑ Elementi di lavoro
- ☑ Eventi del connettore
- ☑ Flussi del controllo origine
- ☑ Modifiche del controllo origine
- ☑ Modifiche del processo del team
- ☑ Notizie
- ☑ Pannello di controllo
- ☑ Report

Esclude
- ☐ Modificato dall'utente
- ☐ Notizie lette
- ☐ Notizie più vecchie di un giorno
- ☑ Facebook
- ☑ YouTube
- ☑ Flickr

Nessuna delle iterazioni correnti è contrassegnata per produrre dei risultati finali. Per utilizzare la vista Lavoro personale, chiedere all'amministratore del progetto di contrassegnare almeno una iterazione corrente in modo che produca dei risultati finali e aggiornare la vista.

Fig. 4. Name of the feed for the created group a) Feed view on Jazz b) Filtering options c)


JazzFFDev
JazzFFDev
04/mag/2009 - Settimana 19
11/mag/2009 - Settimana 20
18/mag/2009 - Settimana 21
lun mar mer gio ven sab dom lun mar mer gio ven sab dom lun mar mer gio
Visualizzazione 28 di 29 elementi | Espa
a)
Elementi di lavoro
Nuvola di tag
Problemi
FFintoJazz
Save
JazzFFDev
b)
Visualizzazione 30 di 31 elementi
Elementi di lavoro
Nuvola di tag
Problemi
Lavoro personale

a)

Ieri (6 elementi trovati)
- Wolfram | Alpha 21:48
- Federico Fiumani – L'ora più bella 21:19
- Federico Fiumani – Paola 20:40
- Federico Fiumani – La mia ragazza dorme 20:30
- Federico Fiumani – Un'ora 20:14

Mostra tutti gli elementi...

All'inizio della settimana (5 elementi trovati)
- Giorgio Canali & Rossofuoco – Nuvole Senza Messico 18/mag/2009
- Revo Uninstaller Freeware - Uninstall Software, Remove Programs, Solve uninstall problems 18/m
- OPSWAT AppRemover. Thoroughly remove your security applications. 18/mag/2009
- L'uomo che cadde sulla terra 18/mag/2009
- Giorgio Canali – Questa No 18/mag/2009

Ultima settimana (17 elementi trovati)
- Grooveshark -- Listen to Free Music 17/mag/2009
- Giorgio Canali – Precipito 17/mag/2009
- Una cosa divertente che non farò mai più 17/mag/2009
- Una stagione selvaggia 17/mag/2009
- Giorgio Canali & Rossofuoco – Tutti Gli Uomini 16/mag/2009

Mostra tutti gli elementi...

Elementi notizie

To JazzFFdev
Add: Just finished to implement the code for WrapperApi Class...checkout!

b)

Oggi (2 elementi trovati)
- Davide aka slacko: Just finished to implement the code for WrapperApi Class...checkout!
- ICT4University - Il programma ICT4University 45 minuti fa

Ieri (6 elementi trovati)
- Wolfram | Alpha 21:48
- Federico Fiumani – L'ora più bella 21:19
- Federico Fiumani – Paola 20:40
- Federico Fiumani – La mia ragazza dorme 20:30
- Federico Fiumani – Un'ora 20:14

Mostra tutti gli elementi...

All'inizio della settimana (5 elementi trovati)
- Giorgio Canali & Rossofuoco – Nuvole Senza Messico 18/mag/2009
- Revo Uninstaller Freeware - Uninstall Software, Remove Programs, Solve uninstall problems
- OPSWAT AppRemover. Thoroughly remove your security applications. 18/mag/2009
- L'uomo che cadde sulla terra 18/mag/2009
- Giorgio Canali – Questa No 18/mag/2009

Ultima settimana (17 elementi trovati)
- Grooveshark -- Listen to Free Music 17/mag/2009
- Giorgio Canali – Precipito 17/mag/2009

Elementi notizie

Nessuna delle iterazioni correnti è contrassegnata per produrre dei risultati finali. Per utilizzare la vista Lavoro personale, chiede iterazione corrente in modo che produca dei risultati finali e aggiornare la vista.

Fig. 5. Posting a message to the group *a)* View of the updated feed *b)*

6 Usage Scenarios

Here we describe some key scenarios to clarify the potential benefits to teams adopting our Jazz client extension.

Facilitate interpersonal connections. Ian has just joined the Irish team of the ALPHA project, which is distributed over multiple sites. Ian's first task is to develop a plugin for Eclipse. Looking at task assignments in the Jazz workspace, he is not able to determine what teammates, if any, have previous experience with the development of plugins.

Adding our FriendFeed extension to Jazz can increase the transparency of group structure and competences, and facilitate the establishment of interpersonal connections. Looking at the project-related FriendFeed stream, Ian decides to filter out any content other than the bookmarks saved in Delicious and the books reviewed on LibraryThing. He finds that Brian has shared several references to articles and has reviewed a couple of books about developing plugins in Eclipse. Thus, Ian, on the one hand, can go through the references he found; on the other hand, he can get in touch with Brian, using one of the communication media available to project members, or he can take advantage of informal, water-cooler conversation, in case they are collocated.

Improve team openness. The BETA project is distributed over two sites, in Ireland and in India. In such settings, one of the solutions that has proved effective to reduce cultural distance and improve team openness is "team buddies", which contemplates that each developer from the Indian site is "buddyed up" with one from the Irish site, providing one-on-one coaching [10]. Thus, Rajiv is buddyed up with Ian. By getting to know each other better, both Rajiv and Ian will broaden their mind, making themselves more open to cultural difference, as well as more indulgent to language issues and prone to think that the others are not going to act, feel, or think the same way.

The use of our FriendFeed extension can help to increase the effectiveness of team buddies. First, our extension gives both Rajiv and Ian insights about interests, opinions, etc. of each other: for instance, books that one has been reading to get started with a new technology they are going to use in the project, or the pictures from a conference that the other has attended. Besides, our extension can be helpful to learn about the daily rhythm and habits of work. For example, Rajiv may learn that when Ian is contributing something to the FriendFeed stream, is a good time to contact him for a help request with lower chances to interrupt him. Finally, our extension can also help to decrease the number of cultural awareness workshops, which are often used as an effective, but expensive, means to reduce cultural distance in distributed projects [2].

Build a socially open workspace. The GAMMA project is a distributed software development project that spans over multiple sites, including one in India and another one in the US. Indian and North American are quite different cultures. Unlike the latter, Indian is a collectivist, strictly hierarchical culture in which for younger developers it is considered unfair to say no to or disagree with senior developers, team leads, and upper management [17,22]. Our Jazz extension can help to lose strict hierarchical relationships and grant a more equal participation to unhindered discussions.

In collocated projects, equality of participation and unhindered communication is encouraged by adopting a physically open environment, with no cubicles or separated offices for managers and team leads [14]. Our Jazz extension can help to break "sir" relationships, by fostering the development of connections established on a more personal basis, and consequently build a socially open workplace where, despite seniority, it is easier for younger developers to deal with senior team leads and participate in discussion with lower peer pressure.

7 Conclusions & Future Work

Jazz is a Collaborative Development Environment that provides developers with both presence and workspace awareness. In this paper we have presented an extension of the Jazz platform, which leverages the feed aggregation service of FriendFeed to embed social information collected from social networks into the Jazz client and provide developers with group awareness as well. We argue that disseminating additional social awareness information to developers working in dispersed teams can help to speed up the establishment of organizational values, attitudes, and trust-based inter-personal connections, thus facilitating the overall distributed software development process.

While current work is focusing on the enrichment of the Jazz client, as future work we plan to extend Jazz also on the server side by augmenting the Jazz user profile with new professional information. Through the mashup of information collected from both business- and development-oriented social websites (e.g., LinkedIn and

Ohloh), we intend to give to Jazz users the opportunity to customize their profile by including information about professional connections and expertise.

Extending Jazz server-side model allows leveraging the information collected in the whole profile rather than within a specific project area, thus enabling information sharing within different teams inside Jazz. Moreover, we plan to develop the server-side mashup as a web service that exposes personal information as RDF Linked Data in order to enable Jazz users exporting professional profiles and reusing them in third-party relevant applications.

Acknowledgments

Our thanks to Davide Fucci for implementing the initial prototype of our Jazz client extension.

References

1. Abbattista, F., Calefato, F., Gendarmi D., and Lanubile, F. Incorporating Social Software into Agile Distributed Development Environments. Proc. 1st ASE Workshop on Social Sofware Engineering and Applications (SOSEA 2008), L'Aquila, Italy, 15 September 2008.
2. Aston, J., Laroche, L., Meszaros, G. Cowboys and Indians: Impacts of Cultural Diversity on Agile Teams. Agile Conference (AGILE '08), Toronto, 4-8 Aug. 2008, pp. 423-428.
3. Begel, A., DeLine, R. Codebook: Social networking over code. Proc. Int'l Conf. Software Engineering (ICSE '09) Vancouver, Canada, 16-24 May 2009, pp. 263-266.
4. Booch, G. and Brown, A.W., Collaborative Development Environments, Advances in Computers 59, 2003.
5. Chi, E.H., The Social Web: Research and Opportunities, IEEE Computer, vol.41, no.9, pp.88-91, 2008.
6. Cubranic, D., Murphy, G.C., Singer, J., and Booth, K.S. Hipikat: a project memory for software development. IEEE Transactions on Software Engineering, 31(6):446-465, 2005.
7. Dourish, P. and Bellotti, V. 1992. Awareness and coordination in shared workspaces. In Proc. ACM Conf. Computer-Supported Cooperative Work (CSCW '92), Toronto, Ontario, Canada, Nov. 1-4, 1992, DOI= http://doi.acm.org/10.1145/143457.143468.
8. Frost, R., (2007). Jazz and the Eclipse Way of Collaboration. IEEE Software, 24(6), pp. 114-117.
9. Herbsleb, J.D., Atkins, D.L., Boyer, D.G., Handel, M., and Finholt, T.A. Introducing Instant Messaging and Chat into the Workplace. Proc. Int'l Conference on Computer-Human Interaction (CHI '02), Minneapolis, MN, USA, 2002.
10. Holmstrom, H., Conchuir, E.O., Agerfalk, P.J, and Fitzgerald, B. Global Software Development Challenges: A Case Study on Temporal, Geographical and Socio-Cultural Distance. Int'l Conf. Global Software Eng. (ICGSE '06), Florianopolis, Brazil, 3-11 Oct. 2006, pp. 3-11.
11. Jazz Platform Technical Overview, https://jazz.net/learn/PrintableLearnItem.jsp?href=content/docs/platform-overview/index.html
12. Kersten, M. Focusing knowledge work with task context. PhD Thesis, University of British Columbia, 2007.

13. Ko, A.J., DeLine, R., and Venolia, G. Information Needs in Collocated Software Development Teams. Proc. 29th international conference on Software Engineering, Minneapolis, 2007.
14. Law, A., Ho, A., A study case: evolution of co-location and planning strategy, Proc. Agile Development Conference '04, Salt Lake City, Ut, USA, 22-26 June 2004, pp. 56- 62.
15. Merrill, D. 2006. Mashups: The new breed of Web app. An introduction to mashups. IBM developerWorks. http://www.ibm.com/developerworks/xml/library/x-mashups.html
16. Omoronyia, I. Sharing awareness during distributed collaborative software development. PhD Thesis, University of Strathclyde, November 2008.
17. Rayhan, S.H., Haque, N. Incremental Adoption of Scrum for Successful Delivery of an IT Project in a Remote Setup. Proc. Agile Conference (AGILE '08), Toronto, 4-8 Aug. 2008, pp. 351-355.
18. Robinson, H., Sharp, H. Organizational culture and XP: three case studies. Proc. Agile Conference (Agile '05), 24-29 July 2005, pp. 49- 58.
19. Sarma, A., Noroozi, Z., and Hoek, A. Palantír: Raising Awareness among Configuration Management Workspaces. Proc. 25th Int'l Conf. on Software Eng. Portland, 2003.
20. Sillito, J., Murphy G.C., and De Volder, K. Asking and Answering Questions during a Programming Change Task. IEEE Trans. on Software Engineering, 34(4):434-451, 2008.
21. Spencer-Oatey, H. Culturally Speaking: Managing Rapport through Talk across Cultures. New York: Cassel, 2000.
22. Summers, M. Insights into an Agile Adventure with Offshore Partners. Proc. Agile Conference (AGILE '08), Toronto, 4-8 Aug. 2008, pp. 333-338.
23. Yu, J., Benatallah, B., Casati, F., and Daniel, F. 2008. Understanding Mashup Development. IEEE Internet Computing 12, 5, 44-52.