SocialCDE: A Social Awareness Tool for Global Software Teams

Fabio Calefato
University of Bari
Dipartimento di Informatica
Bari, Italy
fabio.calefato@uniba.it

Filippo Lanubile
University of Bari
Dipartimento di Informatica
Bari, Italy
filippo.lanubile@uniba.it

ABSTRACT

We present SocialCDE, a tool that aims at augmenting Application Lifecycle Management (ALM) platforms with social awareness to facilitate the establishment of interpersonal connections and increase the likelihood of successful interactions by disclosing developers' personal interests and contextual information.

Categories and Subject Descriptors

D.2.6 [**Software Engineering**]: Programming Environments – *integrated environments.*

General Terms

Human Factors.

Keywords

Social awareness; Trust; Application Lifecycle Management; ALM; Social networks.

1. INTRODUCTION

The concept of awareness comes from the field of CSCW [7] and has had a considerable influence in collaborative software engineering research as well, since it provides mechanisms to coordinate group activities [11]. According to Gutwin *et al.* [8], other than on their coworkers, the members of a group typically also seek information on tasks and artifacts. Following these information needs, four types of group awareness have been acknowledged so far, namely: *informal or presence awareness* (i.e., who is around and their availability), *group-structural awareness* (i.e., members' roles and teams' internal structure), *workspace awareness* (i.e., who changed a shared artifact and when), and *social awareness* (i.e., the information and the understanding that teammates have about their social connections within a group [12]).

Application Lifecycle Management (ALM) is a continuous process of managing the life of an application through platforms that provide software teams with a project workspace and an integrated toolset to support all software development activities, such as requirements management, design, coding, testing, tracking, and release management [3]. The most popular ALM platforms, also known as Collaborative Development Environments (CDEs), unevenly support the four types of group awareness [9]. In general, although to different extents, informal, group-structural, and workspace awareness are well supported in ALM platforms. As for social awareness, instead, its support is either completely lacking or, when available, as in the case of Jazz and GitHub, the level provided is not as adequate as for the other forms of group awareness.

It has been observed that, during the early stage of a project, building personal knowledge about the team is more important than the resolution of other technical issues [13]. More specifically, the establishment of new and stronger social bonds between teammates during kickoff meetings results in an increased level of mutual trust, which later allows teams to efficiently resolve future issues from afar.

Trust is paramount in globally distributed contexts to prevent that physical distance may lead to psychological distance [10], since reduced trust has been reported to (a) aggravate the feeling of being separate teams with conflicting goals, (b) decrease the willingness to share information and cooperate to solve problems, and (c) affect goodwill toward others in case of objections and disagreements [1]. Yet, trust among teams typically grows through close interaction and face-to-face (F2F) communication. Unfortunately, F2F communication is the very activity that global software teams see reduced. Nevertheless, software development organizations have become more and more distributed over the last decade.

To address this challenge, we have developed SocialCDE, a tool that adds social awareness to the most popular ALM platforms. A previous version of SocialCDE, named SocialTFS [4], is included among the Visual Studio Research Tools on the Microsoft Research web site[1].

The goal of the SocialCDE project is to facilitate the establishment of interpersonal connections and increase the likelihood of successful interactions by disclosing developers' personal interests and contextual information – that is, to indirectly foster trust development through increased social awareness. In fact, we argue that conveying such information into the developers' workspace can help to build trust among members of global software teams, thus contributing to create a feeling of

ESEC/FSE'13, August 18–26, 2013, Saint Petersburg, Russia
Copyright 2013 ACM 978-1-4503-2237-9/13/08...$15.00
http://dx.doi.org/10.1145/2491411.2494592

[1] http://research.microsoft.com/en-us/collaboration/focus/cs/tools.aspx

"teamness", a coffee shop-like atmosphere among distant sites. In other words, we expect that having access to information shared on social media as well as the chance to monitor others' behavior on social networks can reduce psychological distance and work as a surrogate of the social interaction occurring in informal F2F meetings, in the same way in which remote conferencing reduces physical distance and surrogates the meeting room for formal F2F meetings.

2. THE SOCIALCDE PROJECT

Usually, it takes several iterations to get a collaborative system right. The first version of our tool was an extension of IBM Jazz, which leveraged the FriendFeed service to aggregate and stream social information within teams [2]. The latest version of SocialCDE, instead, supports Microsoft Team Foundation Server (TFS) and GitHub, as ALM platforms, and most of the largest social networks available today (e.g., Twitter, Facebook, LinkedIn). As per the client side, two are the plugins available, one extends Visual Studio[2] and the other supports any Eclipse-based IDE.[3] Figure 1 shows the main components of SocialCDE. The IDE plugins handle the visualization of all the social content collected from the services enabled by the user (see Figure 2).

The other main component, the Social Proxy Server, is an aggregator that accesses the API of corporate microblogs and social networking websites. Being a proxy, it interacts both with the client plugins and with the ALM platforms through RESTful protocols. Its main duties are retrieving information about both registered users from social network services (SNSs), and software projects from ALM platforms. To make this possible, the Social Proxy Server stores user credentials and caches posts on the behalf of users who give authorization on the first access through OAuth, an authorization protocol used by most social software services. Other than accessing SNSs, the proxy can also handle and store connection data for GitHub, TFS corporate installations, and CodePlex, which is a public TFS installation where Microsoft hosts open source projects. As for the SNSs, SocialCDE allows a user to customize what information it can retrieve from the account (access rights are service-dependent).

As mentioned before, social content is loaded and cached by the Social Proxy Server component. Then, all the information is requested by a SocialCDE client and presented to the end user in the plugin view within the IDE. Such information is shown through three different timelines, namely *home*, *iteration*, and *interactive*. The *home timeline* resembles the same timeline available in microblogging sites such as Twitter or Yammer, as it gets populated by the posts from the current user and his/her followings. We call this type of followings *static*, because to see their posts a user is explicitly required to follow them, that is, add them to the awareness network (i.e., set of colleagues whose actions one monitors and to whom one's actions are displayed) after visiting their profiles. However, awareness networks are *fluid* [6], in the sense that people whose actions one should monitor and to whom one's actions should be displayed change over time, depending on (a) task assignments or (b) the software development phases. Therefore, other than visualizing the stream of static followings in the home timeline, we also designed a *dynamic* type of followings and two other timelines. Unlike static followings, dynamic followings do not require any explicit follow action, as they are automatically added to and removed from a users' awareness network when: i) changes occur to their assignments in the current iteration; ii) an artifact in project repository is opened in the editor view. In the first case, the *iteration timeline* will dynamically add to the current user's awareness network any team member who reported or even commented on a work item assigned to him/her. In the second case, the *interactive timeline* will dynamically displays the posts from anyone who has contributed changes to the artifact opened and currently visualized in the editor view of the IDE.

3. SCENARIO

In order to illustrate how group awareness affects collaboration, we describe a typical GSE scenario in which global software team members face challenges due to the lack of social awareness. Then, we show how disseminating social and contextual information within an augmented ALM platform would help to counteract the same challenge.

3.1 Usage Scenario

Bob is a member of a team that is developing a mobile app for a home automation project. Bob is also taking the responsibility of developing the app component that handles the home-security system settings. The distributed team is using an ALM platform to coordinate the project development. Today, after reviewing the first incremental release of the app, the customer has raised some concerns about the usability of the solution currently adopted, in particular of the user interface of the home security setting. Therefore, Tim, the project manager decides to involve Alice, one of the company's usability experts, and thus, he assigns to her all the work items related to usability issues, including those on which Bob was already working. Tim also requires all the commits with changes applied to user interface to be reviewed by Alice for approval.

3.2 Problems Arising Without Social Awareness Support

Whereas Bob likes to frequently check his code changes and thus ensure growing functionality, Alice tries instead to see the bigger picture of usability and wants to first implement a consistent handling of all functions. Later, as the milestone integration deadline approaches, Bob performs an update of his local repository. He notices that Alice has committed some change sets to resolve the work items previously assigned to him. Looking at the comments in the change sets, he notices that she has frequently pointed out that previous versions made an inconsistent use of sliders widget and dropdown boxes. Therefore, Bob looks at some of the user interface views modified by Alice and thinks that there is no big difference compared to the ones he created in the first place. So he decides to leave the new view he has been coding unchanged and to commit. Hence, in working with Bob, Alice is frustrated that he seemingly doesn't care about the overall performance—Bob doesn't reply to comments she inserts. Bob, on the other hand, sees no progress on her side and perceives the flood of comments without concrete change proposals as slowing down their team's agile style.

So, what happened? Both have a different working style and neither adjusts to the other. They are unaware of each other's strengths, behaviors, and communication needs. Worse, they are unaware of the growing tensions, thus endangering the project, although both are individually trying to make it a success.

[2] http://socialtfs.codeplex.com

[3] https://github.com/collab-uniba/socialcde4eclipse

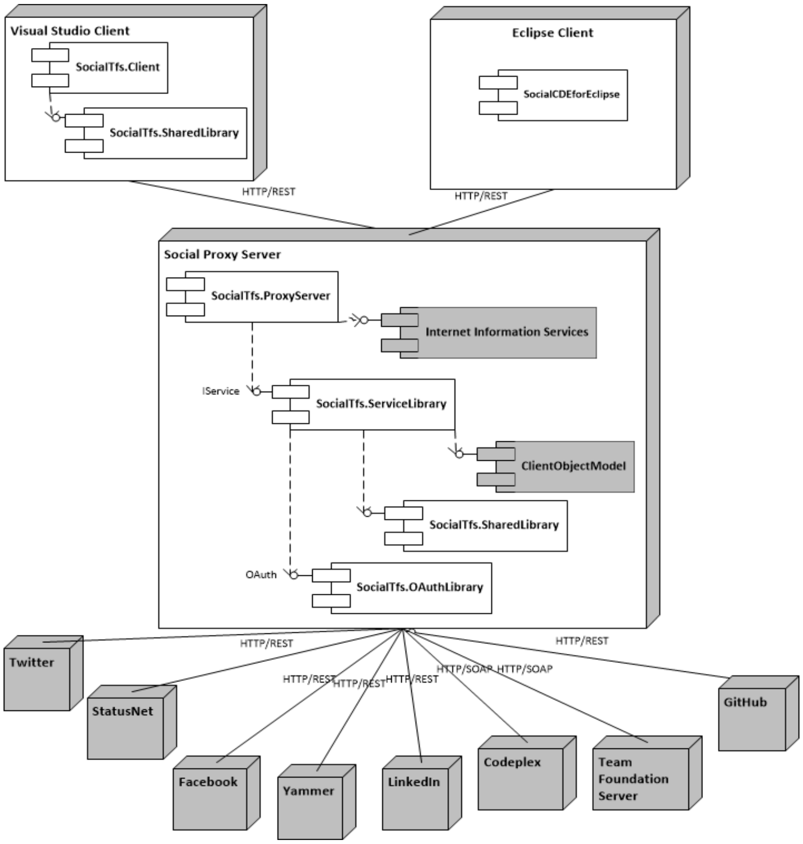
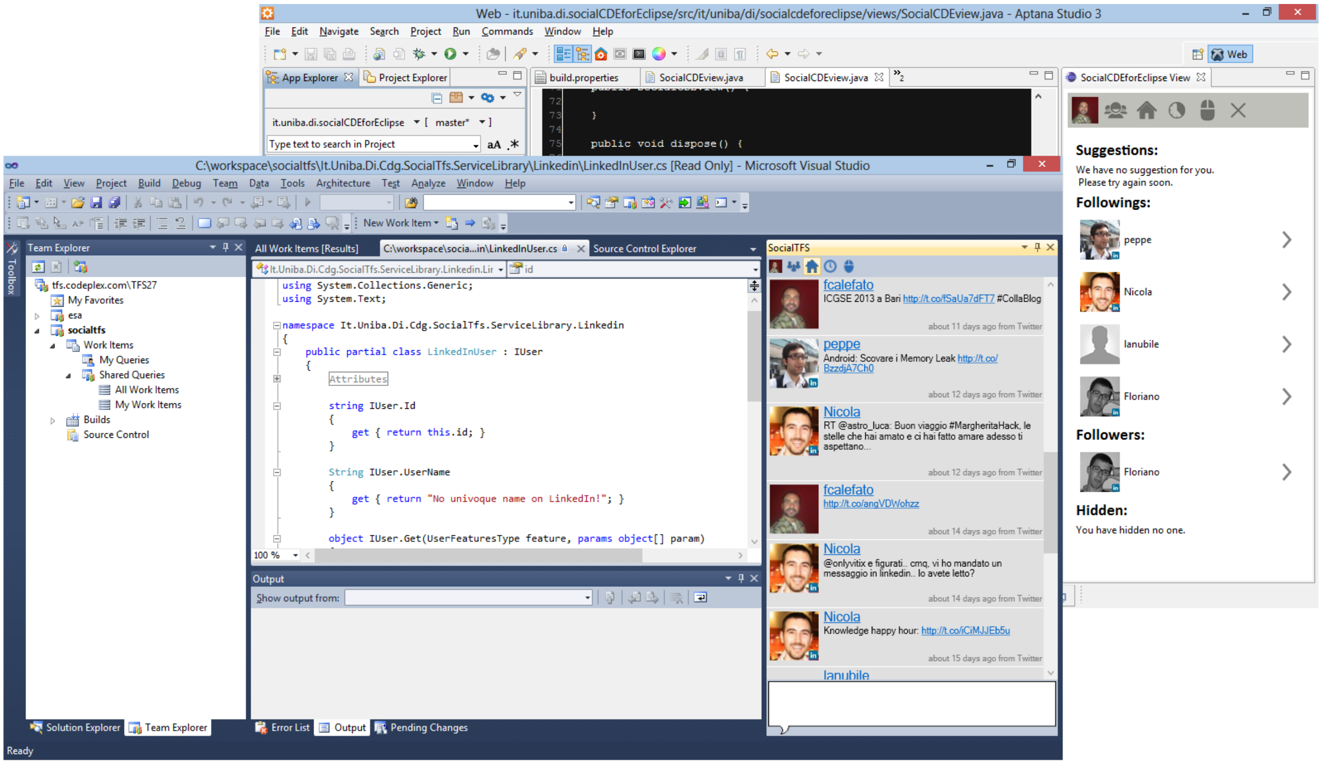


Figure 1. Deployment of SocialCDE components


Web - it.uniba.di.socialCDEforEclipse/src/it/uniba/di/socialcdeforeclipse/views/SocialCDEview.java - Aptana Studio 3
File Edit Navigate Search Project Run Commands Window Help
Web
App Explorer
Project Explorer
it.uniba.di.socialCDEforEclipse
[master*]
Type text to search in Project
build.properties
SocialCDEview.java
SocialCDEview.java
SocialCDEforEclipse View
C:\workspace\socialtfs\It.Uniba.Di.Cdg.SocialTfs.ServiceLibrary\Linkedin\LinkedInUser.cs [Read Only] - Microsoft Visual Studio
File Edit View Project Build Debug Team Data Tools Architecture Test Analyze Window Help
New Work Item
Team Explorer
tfs.codeplex.com\TFS27
My Favorites
esa
socialtfs
Work Items
My Queries
Shared Queries
All Work Items
My Work Items
Builds
Source Control
All Work Items [Results]
C:\workspace\socia...in\LinkedInUser.cs
Source Control Explorer
It.Uniba.Di.Cdg.SocialTfs.ServiceLibrary.Linkedin.Lir
id
100 %
Output
Show output from:
SocialTFS
Solution Explorer
Team Explorer
Error List
Output
Pending Changes
Ready

```
    }

    public void dispose() {
```

```
using System.Collections.Generic;
using System.Text;

namespace It.Uniba.Di.Cdg.SocialTfs.ServiceLibrary.Linkedin
{
    public partial class LinkedInUser : IUser
    {
        Attributes

        string IUser.Id
        {
            get { return this.id; }
        }

        String IUser.UserName
        {
            get { return "No univoque name on LinkedIn!"; }
        }

        object IUser.Get(UserFeaturesType feature, params object[] param)
```

fcalefato
ICGSE 2013 a Bari http://t.co/fSaUa7dFT7 #CollaBlog
about 11 days ago from Twitter

peppe
Android: Scovare i Memory Leak http://t.co/BzzdjA7Ch0
about 12 days ago from Twitter

Nicola
RT @astro_luca: Buon viaggio #MargheritaHack, le stelle che hai amato e ci hai fatto amare adesso ti aspettano...
about 12 days ago from Twitter

fcalefato
http://t.co/angVDWohzz
about 14 days ago from Twitter

Nicola
@onlyvitix e figurati.. cmq, vi ho mandato un messaggio in linkedin.. lo avete letto?
about 14 days ago from Twitter

Nicola
Knowledge happy hour: http://t.co/iCiMJJEb5u
about 15 days ago from Twitter

Ianubile

Suggestions:
We have no suggestion for you.
Please try again soon.

Followings:
peppe
Nicola
Ianubile
Floriano

Followers:
Floriano

Hidden:
You have hidden no one.

Figure 2. SocialCDE client plugins for Visual Studio (left) and Eclipse/Aptana (right)

3.3 Problems Counteracted through Social Awareness Support

The social awareness plugin installed in the ALM platform regularly updates developers' awareness networks to incorporate the posts from the people who are involved in the same work items, e.g., those who have reported or commented on an issue. So, Alice's posts begin to appear in Bob's workspace. Other than Twitter, Alice has also connected her LinkedIn account to the plugin, so Bob can also see from her profile that Alice is very well known in the human-computer interaction community. When Alice tweets that she is happy to have a chance to work with the team on the new mobile platform; Bob decides to reply to her tweet with a "welcome on board" message. As they get more and more in touch, Bob also comes to know from Alice's posts that she has been busy fixing the user interface views that Bob had just coded in the first place. Afterwards, while the deadline of the milestone integration is becoming close, Bobs performs an update of his local repository. He notices that Alice has committed some change sets to resolve the work items previously assigned to him. Looking at the comments in the change sets, he notices that she has frequently pointed out that previous versions made an inconsistent use of sliders widget and dropdown boxes. Therefore, when Bob looks at some of the user interface views modified by Alice, he realizes that she is just right and that he can also apply the same solution to the new view before committing. Afterwards, Alice reviews and approves the change set committed by Bob.

So what happened? Bob and Alice are now more aware of their strengths, behaviors, and communication skills. In other words, they have adjusted their own working styles to a point where their individual efforts have become a shared endeavor to make the project successful.

4. CONCLUSIONS

In this paper we have presented SocialCDE, a tool that augments ALM platforms by aggregating content from multiple social media into the developer's workspace. SocialCDE has been developed to support our hypothesis that information shared on social networks can work as a surrogate of the social awareness gained during informal chats, thus helping to build trust among members of global teams. We are currently running a couple of studies to empirically validate our hypothesis [5].

5. ACKNOWLEDGMENTS

We would like to thank Nicola Sanitate, Giuseppe Santoro, and Floriano Fauzzi for their contribution to the development of the tool. SocialCDE is partially funded by Interreg III 2007-2013 under the project Intersocial and by Microsoft Research through the 2011 Software Engineering Innovation Foundation (SEIF) Award.

6. REFERENCES

[1] Al-Ani B., Wilensky, H., Redmiles, D. and Simmons, E. An Understanding of the Role of Trust in Knowledge Seeking and Acceptance Practices in Distributed Development Teams. *In Proc. 6th Int'l Conf. Global Soft. Eng. (ICGSE '11)*, Helsinki, Finland, Aug. 15-18, 2011, pp. 25-34.

[2] Calefato, F., Gendarmi, D., and Lanubile, F. Embedding Social Networking Information into Jazz to Foster Group Awareness within Distributed Teams. *In Proc. 2nd Int'l Workshop on Social Soft. Eng. and Applications (SoSEA'09)*, 2009, p. 23.

[3] Calefato F. and Lanubile F. Practice: Collaborative Development Environments. *Global Software and IT: A Guide to Distributed Development, Outsourcing, and Supplier Management* (C. Ebert ed.), IEEE, Wiley, 2012.

[4] Calefato F. and Lanubile F. Augmenting Social Awareness in a Collaborative Development Environment. *In Proc. 5th Int'l Workshop on Cooperative and Human Aspects of Soft. Eng. (CHASE'12)*, Zurich, Switzerland, 2 Jun. 2012, pp. 12-14.

[5] Calefato F. and Lanubile F. Can Social Awareness Foster Trust Building in Global Software Teams?" *5th Int'l Workshop on Social Software Engineering (SSE'13)*, Saint Petersburg, Russia, 18 Aug. 2013.

[6] de Souza C.R.B. and Redmiles, D.F. The Awareness Network, To Whom Should I Display My Actions? And, Whose Actions Should I Monitor? *IEEE Trans. on Sw Eng*, 37, 3, pp. 325-340, 2011.

[7] Dourish P. and Bellotti, V. Awareness and Coordination in Shared Workspaces. *In Proc. Int'l Conf. on Computer Supported Cooperative Work (CSCW '92)*, Toronto, ON, Canada, Nov 1-4, 1992, pp. 107-114.

[8] Gutwin, C., Greenberg, S., and Roseman, M. Workspace Awareness in Real-Time Distributed Groupware: Framework, Widgets, and Evaluation. *In Proc. HCI on People and Computers XI (HCI'96)*, London, UK, Aug. 20-23, 1996, pp. 281-298.

[9] Lanubile F., Calefato F., and Ebert, C. Group Awareness in Global Software Engineering. *IEEE Software*, Mar./Apr., pp.18-23, 2013.

[10] Marlow J., and Dabbish, L. Designing interventions to reduce psychological distance in globally distributed teams. *In Proc. Int'l Conf. Computer Supported Coop. Work (CSCW '12)*, Seattle, WA, USA, Feb. 11-15, 2012, pp. 163-166.

[11] Mistrík, I., Grundy, J., Van Der Hoek, A., and Whitehead, J. Collaborative Software Engineering: Challenges and Prospects. *Collaborative Software Engineering*, 1, pp. 389–403, 2010.

[12] Omoronyia, I., Ferguson, J., Roper, M., and Wood M. A review of awareness in distributed collaborative software engineering. *Software: Practice & Experience*, 40, pp. 1107–1133, 2010.

[13] Treinen J.J. and Miller-Frost S.L. Following the sun: case studies in global software development. *IBM System Journal*, 45, 4, Oct. 2006, pp. 773-783.