Augmenting Social Awareness in a Collaborative Development Environment

Fabio Calefato, Filippo Lanubile, Nicola Sanitate, Giuseppe Santoro
University of Bari
Dipartimento di Informatica
via E. Orabona 4, 70125, Bari, Italy
+39 0805443261
{calefato,lanubile}@di.uniba.it, {n.sanitate,g.santoro}@uniba.it

ABSTRACT

Adequate tool support is paramount to enable distributed teamwork, and thus global software teams usually rely on a Collaborative Development Environment (CDE) to cope with geographical distance. The most recent and full-featured CDEs typically provide presence and workspace awareness in one place, but lack any support to social awareness for reducing the sociocultural distance. We argue that disseminating social awareness information within a CDE can both speed up the establishment of a cross-organizational shared context and help developers who have little or no chances to meet and, then, develop trust-based inter-personal connections. For this reason, we propose to extend a commercial CDE in order to provide members of global software teams with information collected from corporate microblogging and professional social networks.

Categories and Subject Descriptors

D.2.6 **[Programming Environments]**: Integrated Environments; K.4.3 **[Organizational Impacts]**: Computer-supported collaborative work;

General Terms

Design, Human Factors.

Keywords

Collaborative Development Environment, CDE, Team Foundation Server, TFS, Visual Studio, Social awareness, SNS.

1. INTRODUCTION

Global software development has emerged as one of the most important phenomenon in today's software and business arena, despite of several issues due to distance. In the case of global software engineering, adequate tool support is paramount to enable distributed teamwork and thus global software teams usually rely on a Collaborative Development Environment (CDE) to cope with geographical distance [15]. To date, awareness support in large-scale development organizations that overcome boundaries is still inadequate [7].

The most recent and full-featured CDEs typically provide presence and workspace awareness in one place, but lack any support to social awareness for reducing the sociocultural distance [5]. Cultural differences pose practical barriers to the development of a shared context as well as relationships and connections (i.e. common ground, mutual confidence, trust) within distributed teams, with a potential severe impact on project management effectiveness.

We argue that disseminating social awareness information within a CDE can both speed up the establishment of a cross-organizational shared context and help developers, with little or no chances of meeting, to develop trust-based inter-personal connections. Professionals, in fact, use content from social networking sites (SNS) to strengthen their weak ties and reach out to employees they do not know [9]. This may be extremely beneficial in global contexts where development units are from different countries and cultures.

We intend to extend the Microsoft CDE, Team Foundation Server (TFS), and its Visual Studio-based client, in order to provide members of large and distributed software teams with information collected from corporate microblogging and professional social networks.

The remainder of this paper is structured as follows. In Section 2 we show the implementation details of our proposed extension of TFS and Visual Studio. Then, in Section 3 we present the potential contribution and benefits of the extension, whereas the related research works are discussed in Section 4. Finally, we conclude in Section 5, presenting our future work.

2. EXTENDING TFS & VISUAL STUDIO

Figure 1 shows a high level view of the three-tier TFS architecture and its building components. The implementation of our extension involves both the application tier (where the Team Foundation Server web application is hosted) and the client tier (where the Visual Studio client runs). More specifically, on the one hand, we build on the *client object model* API and use the same set of key features upon which Microsoft's' own TFS extension for Visual Studio have been implemented. On the other hand, we also have to use the *server object model* API and implement plugins to be deployed on the server. These server-side plugins are necessary to make changes to the project model persistent, so that such changes are always visible to developers, no matter what machine they use to connect to the project area on the TFS server.

SSE'11, Sep.5, 2011, Szeged, Hungary.

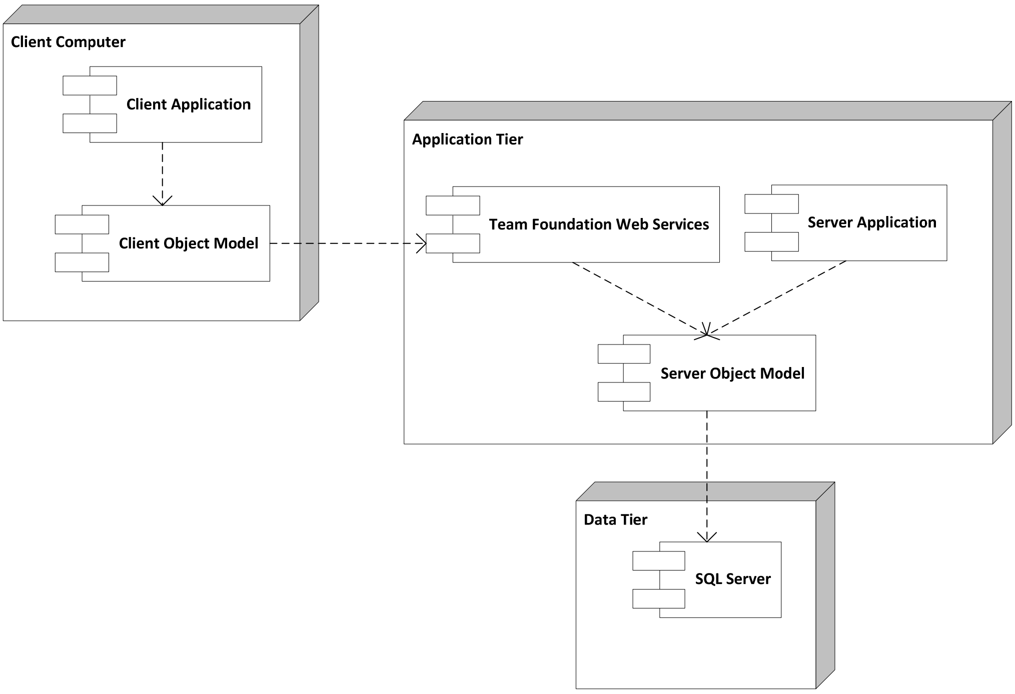

Figure 1. Team Foundation Server extensibility architecture; adapted from [2]

The first step in the design of the proposed extension is the identification of unobtrusive solutions to display and monitor all the information exchanged as microblogging posts within the Visual Studio client.

The microblogging service to use is one that allows corporate installation, such as Yammer[1] and StatusNet[2]. In fact, privacy concerns are reduced as long as individuals know that what they share stays within the company's boundaries (Zhang et al. 2010). One view will thus show the updates from individuals in chronological order.

A timeline will also allow to filter updates occurred during the current day, week, and month. Team members will be able to "follow" (i.e. receive notification of updates) people they know or with similar interests, as typical in every microblogging service. However, we will also enable developers to follow software artifacts that exist in the project workspace (e.g. work items, shared docs, software libraries). By leveraging TFS event service, when an artifact is modified, an update will be propagated to the "followers."

Any member will belong by default to two lists: one including all the people within the company and one including all the developers within the same project team. In addition, however, in order to reduce the volume of update notifications, users will be able to create lists that filter out some notifications and show only the updates from selected individuals or artifacts. For instance, a developer might want to create a list of components in order to filter out some updates and avoid receiving too many notification from the people he/she is following; or, vice versa, the developer might create a list of friends and filter out the updates from the changes occurred to some artifacts.

The "follow" feature will also suggest developers to follow other people (not in any list) because, for example, they are co-followers of the same artifact (e.g. work items).

Finally, another view will be available in the UI in order to allow developers to search for people in the same or other teams, according to their expertise. In fact, expertise profiles of developers will be arranged by aggregating the information retrieved from projects' software repositories, as well as from professional SNSs, such as LinkedIn[3] and Ohloh[4].

3. POTENTIAL CONTRIBUTIONS

There are four main motivations that push professionals to use and share social information within company's boundaries [9],[17]:

1) *caring* – developers are motivated to connect with (i.e. follow) others on a social level because they find easier to work with people they know and like, as well as with those who have similar interests;

2) *browsing* – developers seek help from someone who is knowledgeable on a topic;

[1] www.yammer.com

[2] http://status.net

[3] www.linkedin.com

[4] www.ohloh.net

3) *climbing* – to a certain extent the complement of browsing, as developers are committed to be known as knowledgeable on a topic because they perceive this could help them in advancing their career;

4) *campaigning* – developers are committed to promote their ideas within the team or company.

The development of a social computing extension for a CDE could enhance the support of globally distributed software development projects on all these aspects.

People motivated by caring and campaigning are those individuals more open to contribute content and share social information within the team or company. Giving them the opportunity to post social information (e.g. pictures from a conference they recently attended) on the project list, using a corporate microblogging solution like Yammer, can help to build a socially open workspace and facilitate interconnections across sites. In collocated projects, equality of participation is often encouraged by adopting a physically open environment, with no cubicles or separated offices for managers and team leads. Our extension might help to break "sir" relationships (typical of collectivist, strictly hierarchical cultures) 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.

With respect to browsing, our extension will provide a unified place where to look for specific technical skills and expertise among company's developers, by gathering profile information from professional SNSs such as Ohloh and LinkedIn. Besides, combining the measures of self-described expertise gathered from such socio-professional SNSs with information regarding the geographic dispersion of units and the technical dependencies of the project (e.g. as defined by the project work items assignments available in TFS workspace) our tool could suggest "follow" relationships for particular individuals. For example, the extension could improve team openness by suggesting "team buddies" in order to provide effective one-to-one coaching between a senior and a young developer, as the latter would benefit from building stronger "social" ties with other, more experienced members of the development organization. We expect this feature to provide a complement of the Socio-Technical Congruence (STC) measure, a method proposed by Cataldo *et al.* [6] to calculate the extent to which the social relationships of team members fit their technical, work-related relationships. In fact, to date STC has rather focused on aspects such as communication and team structure. Instead, we aim to complement such aspects with social interaction cues, derived from the network of "follow" relationships existing in the corporate microblog.

Furthermore, the climbing motivation pushes developers to contribute content that indicates them as expert on a topic. Thus, combined with browsing, climbing may provide a measure of members' experience, which can be particularly helpful to managers during team building in order to draw inferences about developers' personal characteristics and expected behavior, and decide what members should be assigned to what teams/units [11] [11].

Finally, given that the next generation of employees uses social software as their dominant communication means, companies should start thinking how to bridge generational gaps and boundaries by supporting this method of communication between employees.

4. RELATED RESEARCH

CDEs were envisioned by Booch & Brown [3], who first acknowledged the need for a "frictionless surface" in development environments. CDEs support developers by incorporating a standard toolset (e.g. compiler, debugger, version control system, bug tracker) within a single project workspace, thus reducing the effort of running multiple different applications to control the software development process and collaborate.

Dourish & Bellotti [10] were instead 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. Software developers most frequently seek information about tasks, artifacts, and coworkers [14]. Three major types of awareness can be identified.

Presence awareness is the awareness of what distant colleagues are doing, their availability for interaction, and how they prefer to be reached, thus helping coworkers to minimize interruptions and disturbances when engaging in collaborative processes [12]. It has almost become synonymous with IM and VoIP because such tools often represent the preferred, lightweight means to broadcast information as well as to ascertain and negotiate availability in order to accommodate opportunistic interaction between teammates.

Workspace awareness means knowing project teams and their internal structure, as well as team members and artifacts. Tools such as Palantìr [18], Hipikat [8], and Mylyn [13] provide developers with workspace awareness information that helps developers to identify other teammates, artifacts, and tasks, which are related to the artifact/task at hand, thus helping 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. 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 and consolidate an organizational culture, connections, and trust-based relationships between distant collaborators, social awareness contributes to project success by improving team's well-being and social health [16].

There are very few software development-oriented tools that support social awareness. One of the most noticeable is Github[5], 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 [1], instead, is a Microsoft prototype that aims at developing a social networking service over code, in which people can also be friends of the artifacts they share. Finally, Calefato et al. [5] developed an extension that made social information available in the Jazz CDE by leveraging the FriendFeed aggregator service.

[5] https://github.com

5. CONCLUSIONS & FUTURE WORK

In this paper we presented our proposal to develop an extension of the Microsoft Team Foundation Server (TFS), and its Visual Studio-based client, in order to provide members of large and distributed software teams with information collected from corporate microblogging and professional social networks.

As of this writing, we have just started to implement our extension. Once the implementation is completed, we plan to run a case study in order to assess the usefulness of disseminating social awareness information within collaborative development environments.

6. ACKNOWLEDGMENTS

This project is funded by Microsoft Research through the Software Engineering Innovation Foundation (SEIF) Award 2011.

7. REFERENCES

[1] 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.

[2] Blankenship, E., Woodward, M., Holliday, G., and Keller, B. *Professional Team Foundation Server 2010*, Wiley, 2011.

[3] Booch, G. and Brown, A.W., Collaborative Development Environments, *Advances in Computers* 59, 2003.

[4] Brocco, M., Groh, G., and Kern, C. On the influence of social factors on team recommendations, 22nd Int'l Conf on Data Engineering Workshops (ICDEW 2010), Long Beach, CA, USA, Mar 1-6, 2010, pp. 270-277.

[5] Calefato, F., Gendarmi, G., and Lanubile F. Embedding Social Networking Information into Jazz to Foster Group Awareness within Distributed Teams, *Proc. 2nd Int'l Workshop on Social Software Engineering and Applications (SoSEA '09)*, pp. 23-28, 2009.

[6] Cataldo, M., Wagstrom, P.A., Herbsleb, J.D., and Carley, K.M. Identification of coordination requirements: implications for the Design of collaboration and awareness tools. *20th ACM Conf on Computer supported cooperative work (CSCW '06)*, Banff, Alberta, Canada, Nov. 4-8, 2006, pp. 353-362.

[7] Cataldo, M. Sources of errors in distributed development projects: implications for collaborative tools. *Proc. Conf. on Computer supported cooperative work (CSCW '10)*, New York, NY, USA, pp. 281-290, 2010.

[8] 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.

[9] DiMicco, J. et al. *Motivations for Social Networking at Work*. Proc. Conf. Computer Supported Cooperative Work (CSCW'08), San Diego, CA, 2008.

[10] Dourish, P. and Bellotti, V. Awareness and coordination in shared workspaces. *Proc. ACM Conf. Computer-Supported Cooperative Work (CSCW '92)*, Toronto, Canada, Nov. 1-4, 1992.

[11] Gorla, N. and Wah Lam, Y. Who should work with whom?: building effective software project teams. Commun. ACM vol. 47, no. 6, June 2004, pp. 79-82.

[12] 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, USA, 2002.

[13] Kersten, M. Focusing knowledge work with task context. PhD Thesis, University of British Columbia, 2007.

[14] Ko, A.J., DeLine, R., and Venolia, G. Information Needs in Collocated Software Development Teams. *Proc. 29th International Conference on Software Engineering (ICSE 2007)*, Minneapolis, USA, 2007.

[15] F. Lanubile, C. Ebert, R. Prikladnicki, A. Vizcaino. Collaboration Tools for Global Software Engineering. *IEEE Software*, vol. 27, 2010, pp.52-55.

[16] Robinson, H., Sharp, H. Organizational culture and XP: three case studies. *Proc. Agile Conference (Agile '05)*, 24-29 July 2005, pp. 49- 58.

[17] Sadat Shami, N. et al. Making Sense of Strangers' Expertise from Signals in Digital Artifacts. *Proc Conf. Human Factors in Computing System (CHI'09)*, Boston, MA, 2009.

[18] Sarma, A., Noroozi, Z., and Hoek, A. Palantír: Raising Awareness among Configuration Management Workspaces. *Proc. 25th Int'l Conf. on Software Eng. (ICSE 2003)*, Portland, USA, 2003.

[19] Zhang, J et al. A case study of micro-blogging in the enterprise: use, value, and related issues. *Proc. 28th Int'l Conf. Human factors in computing systems (CHI '10)*, New York, NY, USA, 123-132, 2010.