INTRODUCING SOCIAL DEVELOPMENT ENVIRONMENTS

Hani Bani-Salameh
Computer Science Department
University of Idaho
Moscow, ID 83843, USA
hani@uidaho.edu

Clinton Jeffery
Computer Science Department
University of Idaho
Moscow, ID 83843, USA
jeffery@uidaho.edu

Iyad Abu Doush
Computer Science Department
Yarmouk University
Irbid, Jordan
iyad.doush@yu.edu.jo

ABSTRACT

A Social Development Environment is a real-time collaborative programming tool with integrated social networking features. This emerging technology is important for distributed software developers, e-learning and technical communities. Social Development Environments advance the state of the art for collaboration, coordination, and project management in software development.

Social Development Environments provide a wide range of facilities for synchronous and asynchronous collaboration and information sharing between team members. This paper provides an overview of SDEs and compares state of the art examples with our novel solution called SCI (Social Collaborative IDE).

Keywords: Collaborative Development Environment, Social Networks, Software Development.

1. INTRODUCTION

Software engineering is a team effort. Software developers spend a majority of their time on programming activities using a development environment coupled with a variety of collaborative tools. Instant messaging systems, bug trackers, and configuration management systems are assembled coherently to compose Collaborative Development Environments (CDEs) for software engineers [1, 2].

Many projects involve a broad community of individuals from different institutions [3]. Software development is a very knowledge intensive and social process where developers exchange information and code fragments, connect with others who share technical interests, post messages in the team or community weblogs, and answer questions and discuss issues in public mailing lists [4]. Software development success depends on the ability to share and integrate information [5].

2. COLLABORATIVE TOOLS GO SOCIAL

Collaborative Development Environments are virtual spaces where developers of a project can meet, brainstorm, discuss, record, and go over ideas to solve design and coding problems, and create software products. During these activities, developers often need to interact with individuals who have expertise in particular technical problem domains. As Booch mentioned [6], "CDEs are essentially team centric, meaning that their primary user experience focuses on the needs of the team (but with points of entry for different individuals)."

Grady Booch [1, 5] stated that the purpose of a CDE is to create a foundation for environments (tools) that reduces the frictions that have an impact on the daily work of the distributed software developers and reduces both individual and group efficiency, and to support the development process during the whole life cycle of the project.

In contrast with the support that CDEs provide for teams working closely together, social networks build online communities of people loosely connected by their common interests or activities. Community is a vital aspect of software development, but software developers tend to focus their attention primarily on their programming environment tools. Software development projects usually include communities of users. Community members play vital roles that reflect the success of such projects, such as reporting bugs, helping other users, and analyzing problems. These observations lead to a new category of tool, the Social Development Environment (SDE).

In this paper the term *social software development* refers to software development in collaborative online communities with social relationships. Software development requires interaction between the people involved in the development process. For this reason, social activities form a big part of their daily work. In software development, developer networks are an instance of object-centered sociality [7], where developers do not usually interact merely to socialize, as in conventional social networks. In contrast, they interact and collaborate primarily through shared project artifacts.

Communication is an important factor for software projects to succeed. With the geographical, cultural, time-zone, or language barriers that face the project members involved in the development, making communication and collaboration more successful is a not an easy task. Communication leads to collaboration between the project's team to accomplish certain tasks. As teams become more and more geographically distributed, collaboration becomes more critical and difficult to achieve. Team members usually use different tools to communicate and that leads to difficulties to collaborate and finish their tasks.

As mentioned earlier, there are hundreds of communication applications that are used by software development communities and these can be combined to support distributed development teams [7, 8]. CDEs integrate collaboration and programming tools in one environment. SDEs integrate the development environment and the social network in a single environment, and support the interaction between the social networking features and the communication and collaboration tools. SDEs add value because they touch on the social and presence elements of software development. An SDE aims to help users maximize their productivity, and to help distributed team members maintain a level of social awareness regarding other team members' roles, activity patterns and contributions to the project, as well as the resources in the community relevant to a given project. In general, the purpose of the SDE is to provide a frictionless environment for software development by eliminating the need to switch between different tools in order to perform their various solitary and collaborative tasks.

To summarize, CDEs and SDEs share substantially overlapping purpose and objectives. SDEs add value by integrating features that help establish and support interaction among distributed developers, strengthen social bonds, communication and interactions, and make it possible for them to work hand in hand, build trust, and have the ability to network. SDEs are a category of CDEs where the system provides a fully featured social network for distributed developer communities. In and SDE, developers can browse profiles, create development projects, and form professional teams. In addition, SDEs provide views for listing people and groups, project data, activities, etc. all from inside the environment. By using these views, users will be able to request help and assist others in their projects, and form strong social bonds among the community members. In general, an SDE is an environment that allows distributed developers to work hand in hand and also have the ability to network. It is a combination of social networking and collaborative software development. The next section describes social teams and their changing nature.

3. THE SOCIAL NATURE OF TEAMS

Teams are central to software development. While some projects involve only immediate team members, many require contributions from a broader community of individuals from different institutions [3]. Many developers contribute to several projects in any given week. Thus, not only must a collaborative tool focus on the activities of the individual project teams, it must also be concerned about the activities and interactions across teams and teams of teams.

In general, teams are not what they used to be. The nature of teams has changed for reasons such as the nature of the work and the increase in geographic distribution. Thus, relationships and communications between distributed team members are becoming more important.

One of the pervasive problems facing any software development team is getting the right level of communication to coordinate their work and perform their tasks effectively, and this problem is more difficult for distributed teams. Factors such as the degree of distribution and the supported technologies affect what types of communication are necessary. Trust between team members is an important factor. It is important to have knowledge of team members' expertise of shared topics and projects, and to have strong informal relationships with team members. Such factors encourage trust between distributed team members [9].

Layzell and others [9], suggest that project members are more enthusiastic when they are personally interested in a project, and when they know and like the other project members. This gives an indication of the social nature of teams and their need to socialize in order to excel. Also, it explains the value of the social interactions between distributed developers, and the role of social interactions in building stronger trust ties among them.

Large scale software development is a social endeavor. For team members to function effectively, they must maintain a certain level of social awareness. Developers must be aware of their other team members' roles and activities, as well as the resources in the community relevant to a given project. Also, in order to catch people when they are "in" and focused on a given task, developers need to be aware of the other team members' primary activity windows and/or activity history [4].

4. TECHNICAL CHALLENGES

Like any other distributed groupware system, an SDE has many technical challenges that need to be addressed. This section touches on a handful of these challenges.

The interactions between distributed team members are disadvantaged due to the lack of the rich social communication and coordination that is only possible when team members are collocated. Being aware of these challenges is important for evaluating the effectiveness of a groupware system. The following are some of the problems that are facing distributed teams [4]:

1. Distributed teams lose awareness of social interactions and other members' activities. They are limited in the number and type of spontaneous interactions that occur between team members.
2. Team members cannot easily observe what each other are working on, the status of their activity, and what tasks are underway.
3. In distributed environments, communication relies on lower-bandwidth tools and applications such as phone and email. Also, they are constrained by transaction delays in comparison with face to face interactions and encounters.

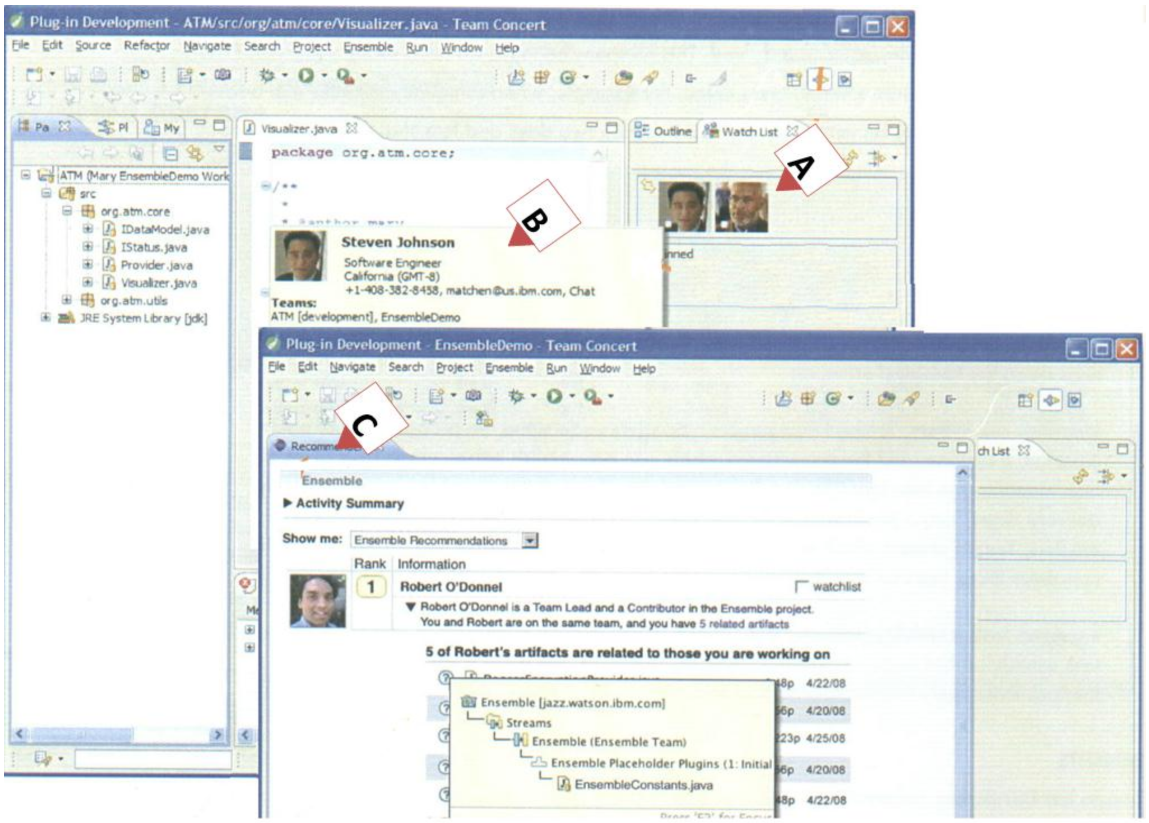

A
B
C
Steven Johnson
Software Engineer
California (GMT-8)
Teams:
ATM [development], EnsembleDemo
Ensemble
Activity Summary
Show me: Ensemble Recommendations
Rank Information
Robert O'Donnel
Robert O'Donnel is a Team Lead and a Contributor in the Ensemble project.
You and Robert are on the same team, and you have 5 related artifacts
5 of Robert's artifacts are related to those you are working on

Figure 1. Jazz's Improvement of Team Collaborating (Rationale Ensemble) *(adapted from [11]).*

4. In order for the distributed teams to manage their project artifacts, they face numerous challenges, including version control and user access management.

5. REAL-WORLD EXAMPLES

An SDE is not a single application, it is a harmony between many different development tools. Several applications support the primary collaborative infrastructures for complete SDEs in a single environment. SDEs include a spectrum of collaborative tools that can be of a big benefit to the development communities, each of these tools adds its own taste and value to the integrated environment, including Web logs (blogs), Mailing lists, Walls, Chat rooms, Whiteboards, and Wikis. This section presents a few related and well-known SDEs.

5.1. Jazz

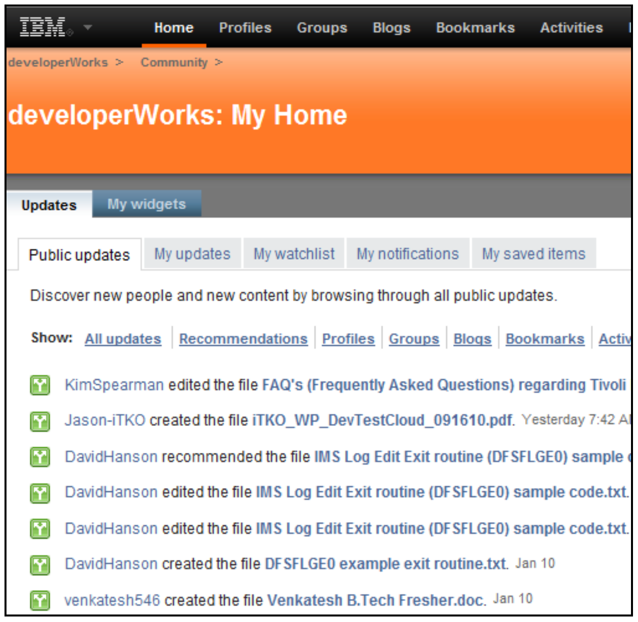
The most widely-publicized example of a social development environment thus far is Jazz, a research project at IBM that adds a set of collaboration tools and features to the Eclipse IDE [2, 10]. The objective is to help foster collaboration within the group. Jazz provides a facility similar to an IM buddy list to monitor who is online and whether they are coding or not. Developers can initiate chats, or use different communication methods such as screen sharing and VoIP telephony. Jazz also provides developers with some awareness of the activities of other team members [10]. Figure 1 shows Jazz's improvement of team collaboration to: 1) support Awareness of similar work; 2) track historic assignments; and 3) link team members [11]. When a user opens a new file, the system shows images of developers who are working in related artifacts (Figure 1(A)), and allows users to hover over developer's images to view their information and what artifacts they are working on (Figure 1(B)), also the system recommends expert developers and allows users to view their artifacts (Figure 1(C)).

IBM added enterprise social networking features to the Rational Team Concert (RTC) development environment in their release of Mainsoft Document Collaboration for Rational Jazz mentioned on SPTechBlog [12]. Developers are able to: (1) Access SharePoint My Sites, with links to blogs and wikis, (2) View SharePoint Personal Profiles, and (3) Use SharePoint People Search. These features are available on top of the existing integration of SharePoint document libraries and workflows with the Jazz development process. Developers can view their other team members' SharePoint My Sites from the RTC's team artifacts view. Blogs and wikis linked from My Sites are also listed in the Team Artifacts view, and they can be opened directly from RTC [13].

5.2. MydeveloperWorks

IBM’s MydeveloperWorks social networking service has the motto: "social networking is the development process". MydeveloperWorks is a new way for distributed developers to connect and interact with their fellow developers. MydeveloperWorks developers can create their own personal profile and customize their home page to get instant access to the people, feeds, tags, bookmarks, blogs, groups, forums, etc. that they care about, and search through user profiles for those with like-minded interests.

IBM's goal with MydeveloperWorks is to connect the global community of software developers and make it easier for them to create new technologies based on open standards such as Java, Linux and XML [14, 15]. Figure 2 [16] shows that users can have access to files uploaded by others.



developerWorks: My Home

Discover new people and new content by browsing through all public updates.

Show: All updates | Recommendations | Profiles | Groups | Blogs | Bookmarks | Activ

KimSpearman edited the file FAQ's (Frequently Asked Questions) regarding Tivoli

Jason-iTKO created the file iTKO_WP_DevTestCloud_091610.pdf. Yesterday 7:42 A

DavidHanson recommended the file IMS Log Edit Exit routine (DFSFLGE0) sample

DavidHanson edited the file IMS Log Edit Exit routine (DFSFLGE0) sample code.txt.

DavidHanson edited the file IMS Log Edit Exit routine (DFSFLGE0) sample code.txt.

DavidHanson created the file DFSFLGE0 example exit routine.txt. Jan 10

venkatesh546 created the file Venkatesh B.Tech Fresher.doc. Jan 10

Figure 2. *My developerWorks Public Files page.*

5.3. CollabVS

CollabVS [17] is an SDE based on Microsoft Visual Studio that allows developers to work together whether planned or ad-hoc. For example, a pair of developers can agree to work together at a scheduled time; in addition, a developer can initiate collaboration with whomever happens to be available online at any given time. CollabVS extends the Visual Studio programming environment by adding collaboration tools, such as text and VoIP chat, watching the unchecked versions of files, and notification of presence in files. Although CollabVS targets collaboration among distributed developers, it still relies on the classical check out/check in model and treats files as the lowest level of granularity [18. 19].

Although it does not provide full social networking tools, CollabVS provides two kinds of presence 1) real-time presence that make the user aware of what other team members are currently doing (it shows what users are online and whether they are editing, debugging, engaged in an instant messaging session, etc); 2) contextual presence facilitates finding relevant information and people quickly [17].

5.4. Zembly

Zembly [20] is a socially-networked development environment from Sun. Zembly supported the development of cloud applications on platforms including Facebook, OpenSocial, iPhone and other platforms. With Zembly users can do social programming, and develop applications with other people using social networking-type features. Not only Developers can reuse pieces and parts of other developers’ projects (a work that they previously implemented to construct new applications), but also inviting friends and colleagues for collaboration. Also, they can see what colleagues are working on via news feeds, and keep up with what others publish and even with what changes they make to their projects’ artifacts. It is a browser-based environment where all activities such as editing, testing, and documenting happen within the browser with the collaboration of other developers.

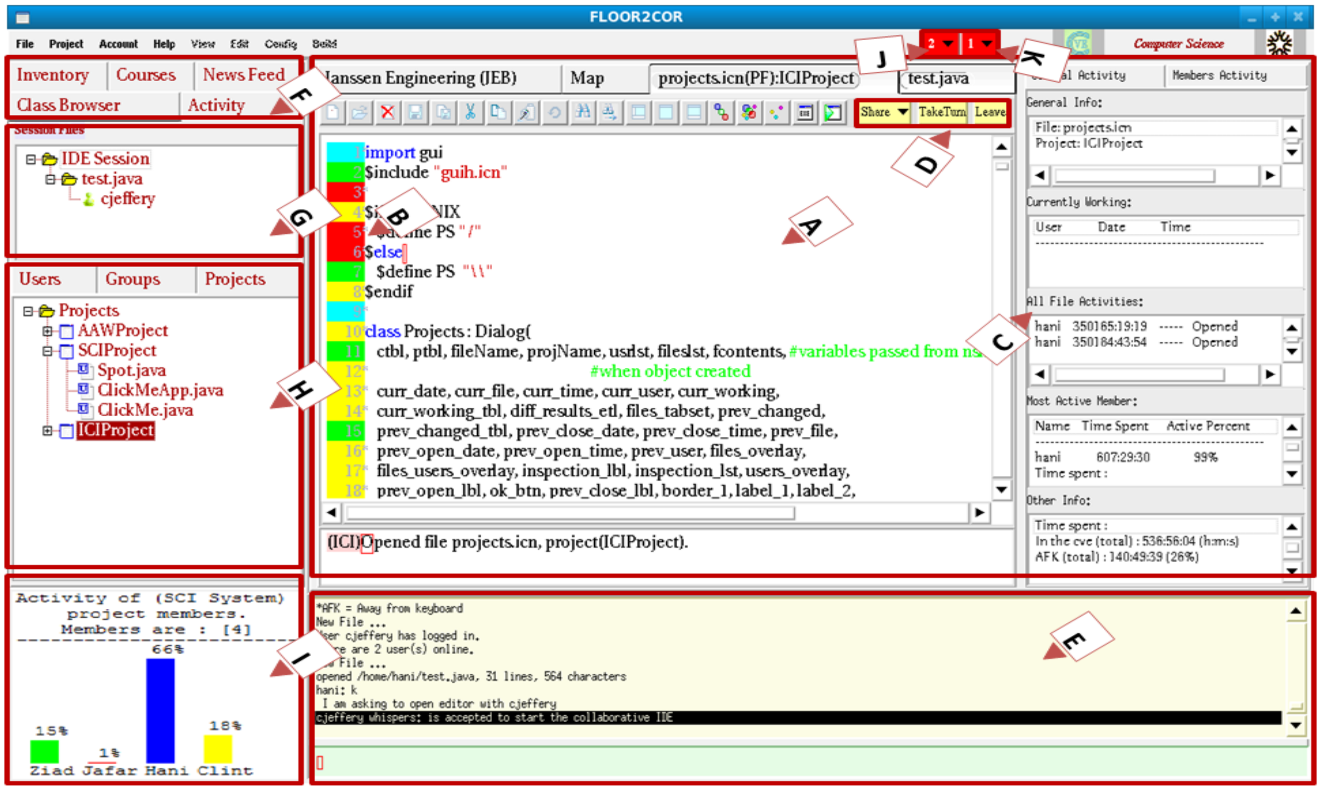
5.5. SCI

SCI [21] is another SDE that allows developers to work together. Developers can use SCI’s collaborative tools to perform standard activities such as: programming, testing, and debugging. SCI supports social presence and messaging within teams and communities, roles and activities, as well as finding relevant information and people quickly.

Figure 3 shows the major SCI components. Tabs (F), show social awareness of the users, their status (online, offline, or idle), active collaborative sessions and members of each session. Information in the tabs allows users to observe the presence of the available teams (groups) and who belongs to each team. Also, they show the presence of each team member, their activity in the project, and their activity history. The social parts (G, H, and I) represent that subset of the awareness information that users get “for free” while concentrating on their project tasks; it includes a sessions tree, users tree, groups tree, and projects tree. Figure 3 (I) is a bar chart that shows additional information on a project from the user projects list. This detail view cycles semi-randomly through the user's projects, allocating more time to projects with high activity. The chart shows the project members' activity and percentage of the time each spent working in the project. Icons (J) and (K) show passive awareness user notifications of pending invitations and requests and emails that they receive from friends and other community members.

5.6. Discussion

The SDE's discussed in this section illustrate just the beginnings of this genre of software tools. To our knowledge, no existing tool takes the full "social networked IDE" concept to its limit. Jazz supports many aspects of social networks. Like SCI, Jazz focuses on increasing the user’s awareness of people, resources, and activities, and on fostering communication among



```
import gui
$include "guih.icn"

$i    NIX
  $define PS "/"
$else
  $define PS "\\"
$endif

class Projects : Dialog(
  ctbl, ptbl, fileName, projName, usrlst, fileslst, fcontents, #variables passed from ns
                                             #when object created
  curr_date, curr_file, curr_time, curr_user, curr_working,
  curr_working_tbl, diff_results_etl, files_tabset, prev_changed,
  prev_changed_tbl, prev_close_date, prev_close_time, prev_file,
  prev_open_date, prev_open_time, prev_user, files_overlay,
  files_users_overlay, inspection_lbl, inspection_lst, users_overlay,
  prev_open_lbl, ok_btn, prev_close_lbl, border_1, label_1, label_2,
```

```
(ICI)Opened file projects.icn, project(ICIProject).
```

```
General Info:
File: projects.icn
Project: ICIProject

Currently Working:
User        Date        Time
-----------------------------------------

All File Activities:
hani   350165:19:19   -----  Opened
hani   350184:43:54   -----  Opened

Most Active Member:
Name   Time Spent    Active Percent
-----------------------------------------
hani       607:29:30          99%
Time spent :

Other Info:
Time spent :
In the cve (total) : 536:56:04 (h:m:s)
AFK (total) : 140:49:39 (26%)
```

```
*AFK = Away from keyboard
New File ...
User cjeffery has logged in.
  re are 2 user(s) online.
  File ...
opened /home/hani/test.java, 31 lines, 564 characters
hani: k
 I am asking to open editor with cjeffery
cjeffery whispers: is accepted to start the collaborative IDE
```

Figure 3. *A View of the SCI Development Environment.*

team members. Both Jazz and SCI support synchronous chat discussions. Also, Jazz provides team-centric discussion boards that compare to the asynchronous news feed supported by SCI. User profiles are not integrated in Jazz, but Jazz users can create their own profile by linking to SharePoint, where in SCI developers can view other developers' profiles, friends, groups, projects, and activities directly from the SCI environment. Jazz supports awareness of the committed code changes with respect to the code repository. In contrast, SCI provides the developer awareness information of the committed and uncommitted code changes, of the currently edited files, and indicates who is responsible for the changes.

Most of the cited and existing projects implement a subset of social networking, but do not fully integrate social networking features within an IDE. They may have some social network-like features, but not others such as user profiles and news feed. The major difference between SCI and almost all the related work cited in this section is the full integration of social network features inside the software development environment, and without linking to third party products.

The SDE tools presented previously in this section are categorized in Table 1. Table 1 shows that SDE tools vary in the number of awareness and social supported features.

6. CONCLUSIONS

Social support for software development is an important emerging field of research. Conventional single-user programming tools and generic communication tools do not provide the needed environment for smooth collaboration between distributed developers due to the size and complexity of today's development projects. SDE tools that support and provide project artifacts' updates in real time have the potential to raise the level of communication, and coordination between distributed developers.

Most current SDEs have limitations, including little support for awareness and online presence, missing social networking features, and weak support for source code repositories' features. The multitude of tools increases the friction that results from switching among different tools.

There is great promise in exploring tool support for the social side of software development. Collaboration plays a crucial role in software development. For this reason, continuing to improve the collaborative tools available inside integrated development environments is of great potential benefit. Collaborative tools can be used alongside a non-collaborative IDE, but integration adds qualitative and quantitative awareness information and reduces the cost of collaboration during the development process, particularly for distributed teams.

While CDEs support distributed communities' work effectively, we believe SDEs will add better benefit development communities and help them increase their productivity and produce better software products.

7. ACKNOWLEDGEMENT

This research was supported in part by a grant from the National Science Foundation under agreement number DUE-0402572. In addition, this work was supported in part by the Specialized Information Services Division of the U.S. National Library of Medicine

8. REFERENCES

[1] Booch, G., and Brown, A. *Collaborative Development Environments*, in Advances in Computers Vol. 59, Academic Press, August 2003.

[2] Cheng, L., Hupfer, S., Ross, S., Patterson, J., Clark, B., and de Souza, C. *Jazz: a Collaborative Application Development Environment*, Demonstration at the 18th annual ACM SIGPLAN Conference on Object Oriented Programming Systems Languages and Applications, Anaheim, CA, USA, pp. 102-103.

[3] Ehrlich, K., Valetto,G., and Helander, M. *Seeing Inside: Using Social Network Analysis to Understand Patterns of Collaboration and Coordination in Global Software Teams*, pp.297-298, International Conference on Global Software Engineering (ICGSE 2007), 2007.

[4] Prasolova-Førland E., and Divitini, M. *Collaborative Virtual Environments for Supporting Learning Communities: an Experience of Use*, Proceedings of the 2003 International ACM SIGGROUP Conference on Supporting Group Work, Sanibel Island, Florida, USA, 2003, pp. 58 – 67.

[5] Lanubile, F. *Collaboration in Distributed Software Development*, in A. De Lucia and F. Ferrucci (Eds.): Software Engineering, LNCS 5413, Springer-Verlag Berlin Heidelberg, pp. 174-193, 2009.

[6] Booch, G. *Introducing Collaborative Development Environments*. Technical report, IBM Rational (2006), Available at ftp://ftp.software.ibm.com/software/rational/web/whitepapers/Grady_Booch_CDE.pdf. Accessed December 20, 2010.

[7] Bouillon, P., Krinke, J., and Lukosch, S. *Software Engineering Projects in Distant Teaching*, 18th Conference on Software Engineering Education & Training (CSEET'05), 2005, pp.147-154.

[8] Bani-Salameh, H., Jeffery, C., Al-Sharif, Z., and Doush, I. *Integrating Collaborative Program Development and Debugging within a Virtual Environment*, Proceedings of the 14th Collaboration Researchers' International Workshop on Groupware (CRIWG 2008), Omaha, Nebraska, USA, 2008, pp. 107-120.

[9] Layzell, P., Brereton, O., and French, A. *Supporting Collaboration in Distributed Software Engineering Teams*, The Asia-Pacific Software Engineering Conference, 2000, pp. 38-45.

[10] Cheng, L., de Souza, C., Hupfer, S., Patterson, J., and Ross, S. *Building Collaboration into IDEs*. ACM Queue 1, 9(2003-2004), pp. 40-50.

[11] *Screenshots of new IBM Rational Jazz products*. http://mikemacd.wordpress.com/2008/06/24/screenshots-of-new-ibm-rational-jazz-products/. Accessed December 29, 2010.

[12] *SPTechBlog. The SharePoint Technology Blog. Bridging the SharePoint-IBM divide.* Available at http://www.sptechblog.com/2009/05/bridging-sharepoint-lotus-divide.html. Accessed January 14, 2011.

[13] *Bringing Social Networking to Jazz.* Available at http://sharepointlotus.net/content/bringing-social-networking-jazz. Accessed January 14, 2011.

[14] *Communities of Practice*. Available at http://www.infed.org/biblio/communities_of_ practice.htm.

[15] *IBM"s Social Network for Software Developers*. Available at http://www.mytechboxonline.com/mtosn/sn-ibmdev-04.html. Accessed December 29, 2010.

[16] *My developerWorks: New ways to build your technical skills and your professional network*. http://www.ibm.com/ developerworks/library/j-mydeveloperworks-intro/index.html ?S_TACT=105AGX01&S_CMP=HP. Accessed January 10, 2011.

[17] *Collaborative Development Environment using Visual Studio*. Available at http://research.microsoft.com/enus /projects/collabvs/default.aspx. Accessed December 29, 2010.

[18] Hattori, L., Lanza, M. *An Environment for Synchronous Software Development*. ICSE Companion 2009: 223-226.

[19] Hegde, R., and Dewan, P. *Connecting Programming Environments to Support Ad-Hoc Collaboration*. ASE 2008: 178-187.

[20] Anderson, G., Anderson, P., Fast, T., and Webster, C. *Assemble the Social Web with zembly*. Prentice Hall PTR (1st Ed.). December 2008. Available at http://www.asgteach.com/books/zemblybook.html. Accessed January 15, 2011.

[21] Bani-Salameh, H., Jeffery, C., Al-Gharaibeh, J. *A Social Collaborative Virtual Environment for Software Development*. Collaborative Technologies and Systems (CTS), 2010 International Conference on DOI 10.1109/CTS.2010.5478525.