
AC 2012-5589: A SYSTEM TO SUPPORT TEACHING GLOBAL SOFTWARE DEVELOPMENT

Robert P. Brazile, University of North Texas
Kathleen Swigger, University of North Texas
Mr. Matt Ray Hoyt, University of North Texas
Mr. Brian Lee, University of North Texas
Brandon Nelson, University of North Texas

A System to Support Teaching Global Software Development

Abstract

Teaching students how to work in globally distributed groups is being done throughout the World. As such, a myriad of different tools have been created to help students work in teams and collaborate. However, there is still the need for effective tools that will enable multi-universities to effectively schedule those distributed group projects. This paper presents a Web-based tool that was developed in the context of a globally distributed software engineering course. The Web-based software enables instructors from multi-universities to view instructors' information and schedule projects throughout the year. The authors of the paper discuss the challenges in creating multi-university projects, as well as present the main features of the software and an example of how it is being used to create globally distributed software projects.

1. Introduction

Teaching students how to work in globally distributed teams is difficult. But figuring out how to schedule the various universities that will participate in the global distributed team projects is even more difficult. Because of ongoing changes in industry, engineering faculty throughout the world are rapidly introducing courses that are designed to teach students how to work in globally distributed teams. The rationale for teaching these courses is that they allow students to learn new types of skills by teaching them how to work in groups and share information. These courses also provide a degree of realism, because they help students become sensitized to issues in concurrent engineering and train them in interpersonal skills, communications, and system integration¹. The importance of these experiences cannot be taken too lightly, especially given the need to prepare students to work in a global economy². Given the advantages of multi-institutional collaboration, however, few universities currently take on the challenges of coordinating projects among several universities. Educators commonly cite logistical and coordination barriers, such as mismatched school calendars and incompatible holiday schedules^{3,4,5,6}. While there are many reports of successful globally distributed engineering projects^{2,7}, questions remain about how to create and sustain meaningful multi-university experiences that can operate within real-world boundary conditions.

To address these particular issues, we developed a special software tool called the Instructional Planning Center (IPC) that allows teachers and educators to schedule globally distributed team development projects. The context for the creation of this tool is a multi-year research project that is aimed at increasing the effectiveness of globally distributed learning teams, particularly student programming teams that are composed of individuals who have different cultures and live in different time zones. The NSF-sponsored project involves researchers from the US, Turkey, Panama, and England and is focused on issues related to teaching computer science students how to work in international teams on large software projects. The universities participating in the research are in the US (2 universities), the UK, Panama, and Turkey (2 universities). Each of the five universities represents a same/different time zone geographical configuration that allows students to experience the full gamut of culture and time zone variations. While the experiences have been productive for both students and educators, the

overhead of administering the schedules for the different projects has been overwhelming. The difficulty in the logistics of organizing multi-university projects stems from the difference in university calendars. In our case, four schools are on a semester system, while one is on a term system. All the schools start at different times (two in August, one in September, and two in late September). One school finishes their spring term in May, while the others finish their terms in late June. Furthermore, one university has a “summer” break from January to March 15, while a second university ends its final term in April. Needless to say, we have had to work hard to find ways to work around the various university-calendar differences and find dates that meet the needs of both the students and instructors. As the number of participating schools has increased (two more universities will join the group this year), finding appropriate project dates that do not conflict with school calendars and holidays is becoming more difficult. To overcome these challenges, we developed a Web-based tool whose sole purpose is to support the creation and management of global software team projects.

The paper is structured as follows. Section 2 describes related work; Section 3 presents the overall design as well as the database for the system. Section 4 describes the interface in detail, and Section 5 provides an example of how we are currently using the software to manage different team projects.

2. Related Research

The idea that engineering students should learn how to collaborate is not new; Engineering educators have been promoting group projects for years⁸. Computer Science faculty have also discovered that distance-learning technology can be effective for teaching certain types of skills^{6,9}. Commercial tools such as Blackboard (www.blackboard.com/) provide general platforms for publishing course materials on the web, while software such as Groove (grv.microsoft.com/) and Microsoft SharePoint (office.microsoft.com/en-us/microsoft-sharepoint-workspace-2010-FX101825648.aspx) can be used for sharing documents and asynchronous messaging. More recently, engineering faculty have experimented with courses that involve multi-institutions and include groups of students from different sites and locations¹⁰. These multi-institutional projects allow students to learn new types of communication skills and, more importantly, software tools that support the sharing of information. Students enrolled in these courses must learn how to use various online tools to build teams, exchange information, and work on projects together¹¹. While there are many examples of online tools and management software to support *student* projects, there are very few tools that have been extended to the teachers who must schedule these multi-institutional events. A variety of studies report that globally distributed learning projects are difficult to manage and require a significant infrastructure to leverage the staff's and instructor's time^{12,13,14,15}. Either institutional regulations or their university calendars discourage most teachers from exploring and maintaining globally distributed learning team projects.

While educators are relatively new to the management of distributed teams, industry personnel have been using a variety of project management tools for years. However, traditional management tools did not always connect processes over different countries and/or time zones. What has eventually emerged is a series of tools to support collaboration among team members, enabling the facilitation, automation, and control of the entire development process. As a result, a number of tools have been developed to help teams, particular global software development

teams, with one or more phases of the software development process. For example version-control systems let team members share software artifacts in a controlled manner. Subversion (SVN; subversion.tigris.org) is a popular open source version-control system that facilitates distributed file sharing. (A complete listing of software for distributed computing can be found in ^{16, 17, 18}). Most of these systems use a centralized architecture, in which a single central server hosts all project data ¹³. Developers normally check out code on their local machines and, when finished, upload the new code to the version control system. Developers can also choose among specific collaboration tools that map to typical life-cycle activities. For example, collaborative project management tools such as ActiveCollab (activecollab.com) and WorldView¹⁹ offer a Web-based interface to manage project information for calendars and milestone tracking. Such tools give managers an overview of project status at different detail levels, such as team member locations and contact information.

There are also document management systems, such as Sharepoint (sharepoint.microsoft.com) and Docushare (docushare.xerox.com) that create databases that allow people to distribute project related documents. Tools that support the testing of software also exist, such as HP Quality Center (hp.com/quality_center) and JUnit (JUnit.sourceforge.net). While such tools can support collaboration for a company with distributed resources, they are generally inappropriate, too expensive or too cumbersome for scheduling team projects in an educational environment.

The latest advances in IT technologies has also enabled and encouraged scientists to form large-scale and multidisciplinary scientific research laboratories to solve complex scientific problems²⁰. Demanding intensive computation and data sharing, these projects are collaborative in nature and usually include multiple domain scientists with domain-specific expertise located at geographically distributed universities²¹. Early laboratories focused mostly on the sharing of large, expensive instruments such as astronomical telescopes, particle accelerators, oceanographic instruments, atmospheric observatories, and space research applications^{22, 23}. More recently, laboratories have focused on communication and collaboration support through the integration of various groupware tools²⁴. For example, the Biological Collaborative Research Environment (BioCore) includes specific tools for biologists (such as visualization programs), asynchronous and real-time communication tools as well as collaborative writing tools²⁵. Laboratories have also emerged in other, more specialized disciplines such as the FBIRN project, which houses information about Schizophrenia²⁶. More recently, researchers have started to build more domain general tools such as Internet-based services that support collaborative scientific workflows²⁷. Although these tools provide mechanisms for automating and accelerating collaborative scientific discovery processes for a wide range of science and engineering disciplines, they are not particularly applicable to the scheduling task that confronts teachers interested in assigning globally distributed team projects.

In spite of the development of a wide range of collaborative tools for students, software developers, and research scientists, there is an absence of software for the scheduling and managing of globally distributed student teams. The problems associated with managing courses and course schedules in different locations is a formidable problem. The software described in this paper attempts to meet this challenge. Although our software is used to support global software development student teams, we believe that it can also be used to support distributed learning teams in general.

3. The IPC Architecture and Database

As previously mentioned, the overall goal of the Instructional Planning Center (IPC) is to provide a repository for information needed to formulate and plan projects for instructors who are trying to teach students how to work in global software development learning teams. It should be noted that the IPC software is not designed to monitor nor record data generated by students who are doing the project. Rather, the software is there to support the instructors who need to coordinate projects among several universities or courses.

The overall architecture of the IPC consists of a database and a Web-based portal, where various components and functions are aggregated together. The IPC database stores information about users, university calendars, courses, and projects. The Web-based portal allows users to input the data into the database. Each Web component displays and accepts relevant information into their own window panels, which is then aggregated by the system and displayed together in a Web page. In order to allow a faster development and easier maintenance of the system, the Web-based application was developed using the Yii framework, a development environment that supports the creation of PHP applications.

3.1 Database

To plan for a successful project, the instructors must be able to view 1) university calendars that have overlapping begin-end dates, 2) courses with similar or complementary topics, 3) courses that have students with similar or complementary skill levels, and 4) courses that have sufficient numbers of students to make up the global learning teams (Figure 1). Data about the instructors, courses, term schedules and proposed projects are stored in a MySQL database. The following is a description of the IPC database.

Users: The Users table records information about instructors who want to be part of a globally distributed project. This table includes the name of the instructors as well as the names of any research assistants who are working on the projects. The table also includes information about the identity of the instructor and their contact information. There is a relationship link to identify the University that an instructor is affiliated with. The User table is linked to the Course table, which can be used to identify the courses that a user teaches (if any).

Course: There is a course entry for each course taught by an instructor who wants to participate in a project. This table includes the course name and a description of the course. The course table also stores the syllabus for a particular course. The course is linked to a Term (e.g. semester), along with a start and end date for the term. Each course has an entry for the number of students enrolled in a course, so that other instructors can determine if appropriately sized teams can be constructed. The level of the course is also included so that instructors can make decisions about whether to include students who have (or do not have) particular skills in one of their projects.

Term: The Term table stores the start and end dates for the terms at each university, as well as the university's holidays. One of the most difficult problems that instructors have each semester is matching a set of consecutive days that students from the participating universities are available to work on a project. The holiday schedule is included in the term table because each

University and each country often celebrates a different set of holidays, and/or has a different length for those holidays.

Project: The Project table contains information about proposed and selected projects. This table stores the name of a project and its unique identifier. The Project table stores a description of the project, the start and end date for the project, the number of groups that will participate in the project, and the number of students in each group. Each project is linked to the courses that are involved in the project. There is a status indicator for each proposed or selected project, which is displayed whenever an instructor agrees to participate in a particular project.

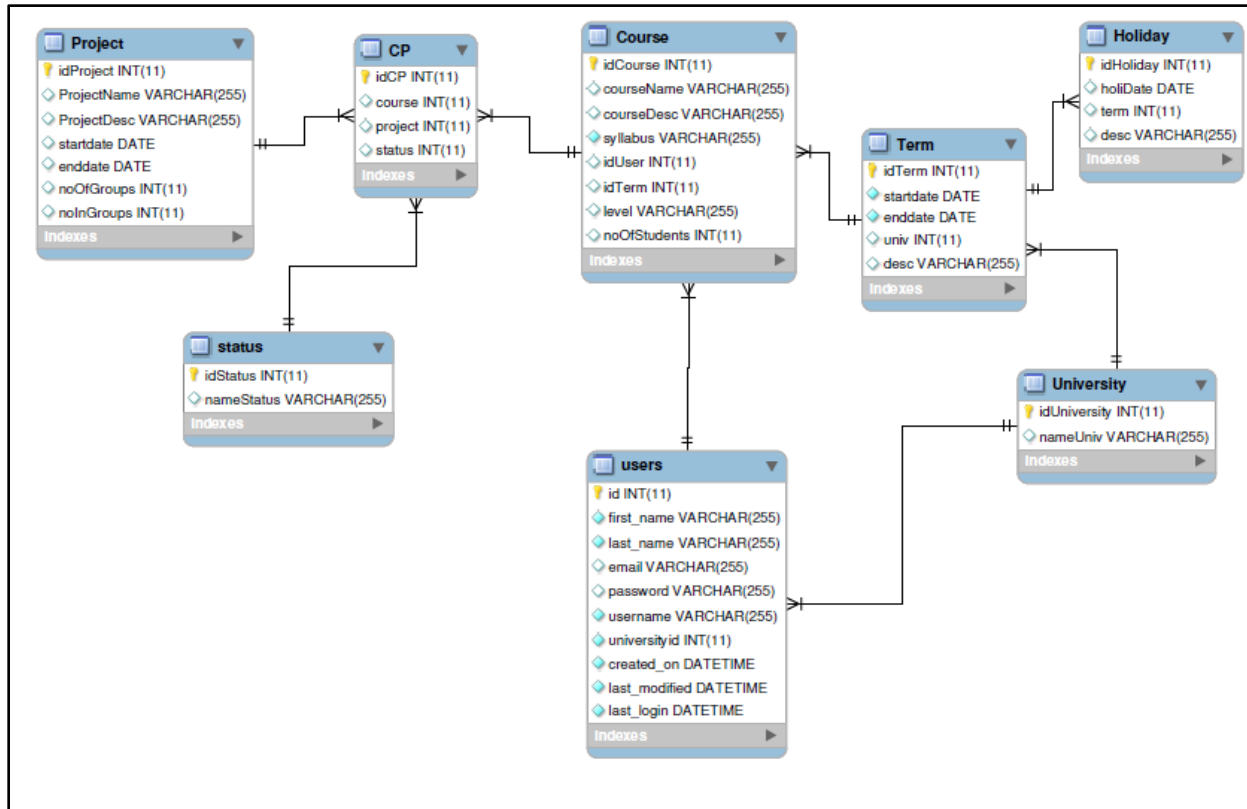


Figure 1. IPC Database

4. IPC Interface

The IPC Web portal provides access to the typical functionalities of a project scheduler through a Web-browser. After the login phase, IPC presents a page with a list of main tabs dedicated to different content: viewing semester/project information; creating and editing semester/project information; and performing administrative functions. By selecting a particular tab, the user is able to access its associated workspace and all its different sub-activities. For example, the *Manage Data* tab provides a set of task specific activities related to entering semester, course, group, and project data. IPC administrators are provided with an additional *Administration* tab, where they can manage users and organize workspaces by adding/removing users and universities. The interface currently supports a number of different functions that help support

the creation of team projects among the different universities. A discussion of each of these functions now follows.

4.1. View Term Data

One of the key issues in multi-university project collaboration is the availability of awareness information about remote collaborators. In particular, realistic global software development projects generally require the use of multiple groups from multiple universities located in different countries and time zones. The IPC *View Term Data* page (Figure 2) provides a way to see all the information about participants and the various times that are associated with that information. For example, instructors can see when universities begin and end their terms (or semesters), when a university has a holiday, which universities are offering courses, which universities are involved in projects, and when those projects are going to take place. These different types of information can be easily distinguished because each data item is highlighted with a different color. Users can also click on an entry (e.g., Monmouth University) and see more detailed information about the semester (see Figure 2).

Instructors can customize the display that appears in the *View Term Data* page by selecting a specific year and time, and then indicating how the data should appear; i.e., annually, monthly, or weekly. The Overview version of the *View Term Data* page is, by default, available to the user when they first enter the system. A compressed image of the Overview data appears at the bottom of this screen (Figure 2a). This display is useful because it provides a quick reference to the semester information, which helps instructors determine the availability of dates among the various universities. Thus, instructors can quickly see which university might be able to participate in a globally distributed project.

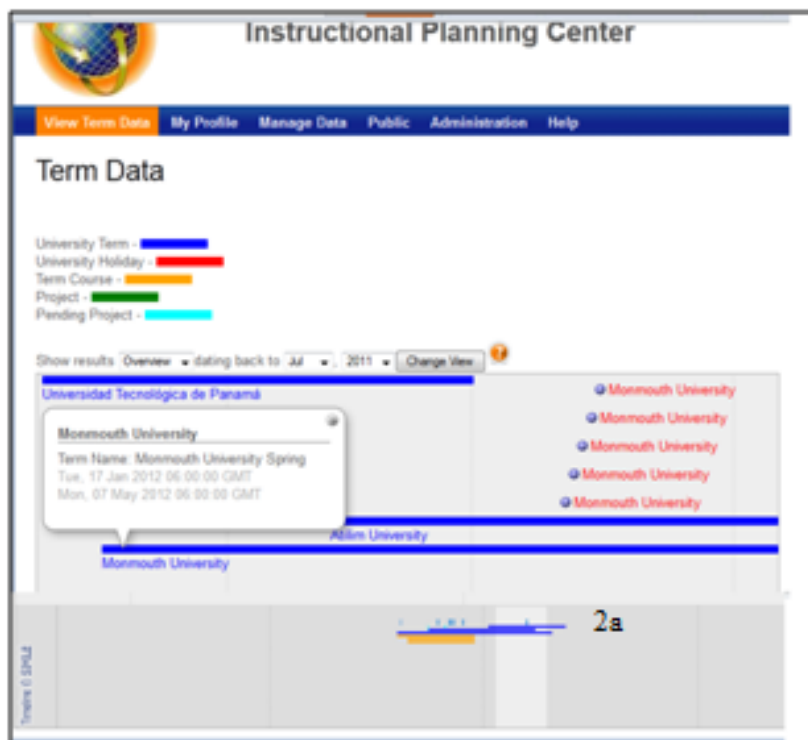


Figure 2. View Term Data Interface (figures 2 & 2a)

4.2. Manage Data

The data creation and management functions are supported through the *Manage Data* page (Figure 3). This page is where instructors enter their semester/term, holiday, course and project information. For example, an instructor enters their semester/term information by selecting the “create term data” selection, at the right-hand side of the page. Once the *Create Term Data* page is displayed, the instructor can enter a data descriptor, start and end date (Figure 3a) for their universities’ semester. Holiday information is entered in a similar manner. Course data is entered by selecting the *Create Course Data* menu item. The *Create Course Data* page allows the instructor to enter information about a specific course such as its name, description, syllabus, level, and number of students (Figure 3b). All of this information is then stored in the IPC database. Finally, an instructor can create a global software development project for a course by selecting the *Create Project* menu item. This selection takes the instructor to a page where they provide a name and description of the project and also specify the number of groups who should participate in the project as well as when the project should begin and end.

Instructors can remove or edit their semester, course, and project information by selecting one of the *Manage* (term, course, project) menu items. For example, if an instructor wants to change their semester data, then they would select the *Manage Term Data* item. The system then displays a page (Figure 3c) that allows the instructor to update any of their entries.



Figure 3. Manage Data Interface (figures 3, 3a, 3b, 3c)

4.3. Public Interface

The *Public* Page (depicted in the bottom-left part of Fig. 4) is the area that facilitates the coordination of global projects among the different universities. It allows the discovery of projects within the system and enables remote instructors to choose the projects they prefer or feel most comfortable with. For example, after an instructor creates a project (Figure 4), then he/she can invite other universities to join the group project by associating a project with a set of courses (Figure 4a). To do this, instructors must select *Create New Relationship* from the menu on the Right. The instructor then selects a project and the various courses that will participate in a particular group assignment (Figure 4b). Instructors in other universities then receive an invitation (via email), notifying them that they have been invited to participate in a specific project. If they wish to accept the invitation, then they click on the url in the email and are immediately sent to the *Accept Pending Operation* page. Here they must indicate that they will participate in the project.

The *Public* interface also allows instructors to modify any project assignment. Any modification that a project participant wishes to make is done through the *Modify Existing Relationship* page (Figure 4b). To update or delete an item, the instructor simply clicks on the appropriate icon that appears with the appropriate project-course entry.

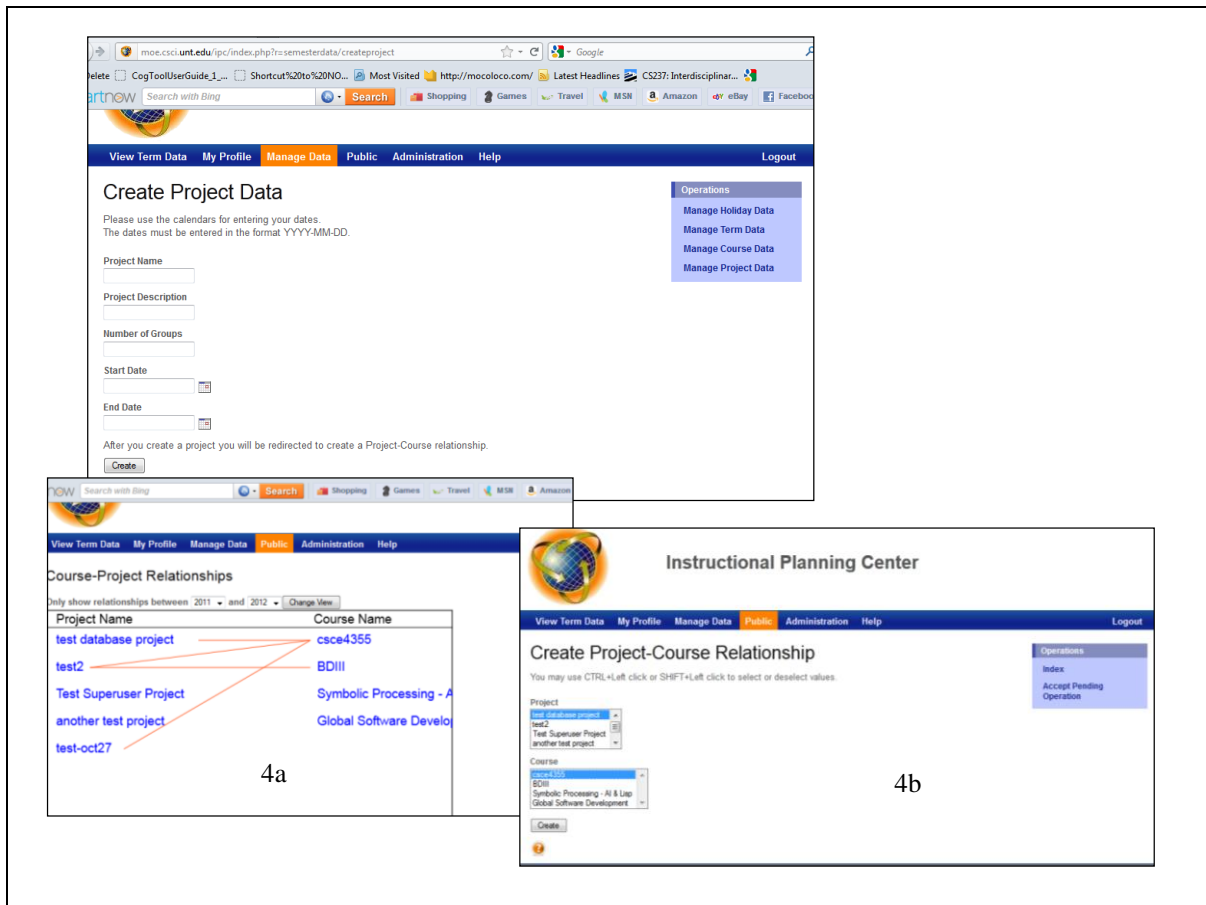


Figure 4. Public Interface (figures 4, 4a, 4b)

4.4. Administrator Functions

The IPC web application also contains special functions for administering user information. For example, an administrator can create new users, add a university, change information about a user or university, view log files, etc. Figure 5 shows the functions that are available to the IPC administrator. Whenever a user is created, the administrator must enter a name, email, username and university. This is how the system knows which universities are available for projects. Figures 5a-5c show what types of information are displayed when administration functions are invoked.

5a: Manage Users

First Name	Last Name	Email	Username	University Name	
Brandon	Nelson	brandonnelson3@my.unt.edu	bgn0015	University of North Texas	[Edit] [Delete]
Brian	Lee	brian@the-lee-home.us	briani	University of North Texas	[Edit] [Delete]
Matt	Hoyt	mthoyt4@yahoo.com	mth0057	University of North Texas	[Edit] [Delete]
Robert	Brazile	Brazile@cse.unt.edu	brazile	University of North Texas	[Edit] [Delete]
teacherfirst	teacherlast	therealcy6@hotmail.com	teacher	University of North Texas	[Edit] [Delete]
Gabriel	Vinado	efg1920@hotmail.com	gabriel	University of North Texas	[Edit] [Delete]

5b: View Logs

Number of results: 10 [Submit]

Username	Ip Address	Action	Action Data	Time
swagger	71.170.14.215	View	users/index	2012-01-03 15:3
swagger	71.170.14.215	View	users/create	2012-01-03 15:3
swagger	71.170.14.215	View	users/index	2012-01-03 15:3
swagger	71.170.14.215	View	viewdata/index	2012-01-03 15:3
swagger	71.170.14.215	View	LOGGED IN	2012-01-03 15:3

5c: Create Users

Empty fields are acceptable, except where denoted.

First Name:

Last Name:

Email *:

Username *:

University Name:

Figure 5 Administrative Interface (figures 5, 5a, 5b, 5c)

5. An Example

The system is currently being used by seven instructors in the US, Turkey, UK, and Panama. In Fall 2011 we used the interface to create and manage two global software development student projects; one between a US university and Turkish university, and one between a Panamanian and a second US university. At the beginning of the fall semester, the seven instructors entered their semester and course information. At a later date, the project information was entered,

followed by the approval of the project by each of the participating instructors. As an illustration of this process, we present the following example.

A typical faculty member begins by bringing up the IPC website and entering his/her user name and password. The system processes this information and verifies that the faculty member's name and password are correct. If this is successful, the system automatically displays the default *View Term Data* page. Once in the IPC system, a first-time instructor would normally select the *Profile* tab, and verify that their name and email address is correct and update any information that might be missing, such as university affiliation. After updating the profile information, the instructor would proceed to the *Manage Data* section of the website. From this page, the faculty member clicks on the *Manage Term Data* and enters all the information about their University's semester or term. Following this, the faculty member would then select the *Manage Holiday Data* and enter the dates for all the holidays that occur within a given semester. Finally, the instructor would select *Manage Course Data* option and enter the name of the course that he/she is teaching that semester, upload its syllabus, input the number of students who are enrolled in the course, and indicate the skill level for the course.

At this point, the instructor would probably return to the *View Term Data* page and scan the data, noting any possible conflicts and/or overlaps among the holiday and semester data. If the instructor finds that another instructor is available at the same time period, and that that same instructor has a course that is compatible with his/her course, then he/she would return to the *Manage Data* page and select the *Manage Project Data* entry. The instructor would now propose a possible project by entering information about the assignment. After completing this form, the instructor then proceeds to the *Public* page and selects courses that might be possible candidates for the proposed global collaboration project. The instructors of the courses that have been proposed for the project receive an email asking them to participate in an upcoming project. If they agree, they click on the url that is embedded in the email. Once forwarded to the appropriate page, the instructor then accepts the invitation to participate in the project. Once everyone agrees, the lead instructor begins the process of setting up the groups for the global software project.

6. Conclusions

This paper describes the Instructional Planning Center (IPC) application, a Web-based system written in PHP and supported by a centralized database. The purpose of the system is to provide an environment where instructors can search, discover, and find partners who will collaborate on global software development student projects. The system is unique because it supports a variety of functions for managing these projects and logs the activities of the participants. Thus, it is possible to collect data on how the software is used by instructors over time.

With respect to future goals of this project, we intend to both extend and refine the current software, as well as to evaluate it in realistic scenarios, such as the ones suggested in this paper. Other collaboration tools under development include an *Event Viewer* for automatic display of instructors' usage information so that it may be possible to suggest real-time meeting times among the participating instructors.

Another feature under development is a notification system that will provide automatic suggestions for course-project match-ups among the various participants. University calendars and instructors' course offerings tend to remain the same from year to year. Thus, it should be possible to study the patterns among the participating instructors and identify the compatible courses. To address this issue, we need to obtain instructor data over several semesters to understand which courses and projects were successfully executed.

6. Acknowledgments

This material is based upon work supported by the National Science Foundation under Grant No. 0705638. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation. We also wish to thank the students who participated in the study, and the many colleagues (in all four countries who helped make this research possible.

7. References

- [1] P. Bouillon and J. Krinke, "A Platform For Teaching Distributed Software Engineering," 2004.
- [2] H. K. Edwards, "Analysis of the Effectiveness of Global Virtual Teams in Software Engineering Projects," in *Proceedings of the 36th Annual Hawaii International Conference on System Sciences (January 06-09, 2003)*. HICSS. IEEE Computer Society, 2003.
- [3] L. J. Burnell, J. W. Priest, and J. B. Durrett, "Teaching distributed multidisciplinary software development," *IEEE Software*, vol. 19, no. 5, pp. 86- 93, Oct. 2002.
- [4] J. Favela and F. Pena-Mora, "An experience in collaborative software engineering education," *IEEE Software*, vol. 18, no. 2, pp. 47-53, Apr. 2001.
- [5] M. Guzdial, P. Ludovice, M. Realf, T. Morley, and K. Carroll, "When collaboration doesn't work," in *Proceedings of the International Conference of the Learning Sciences*, 2002, pp. 125-130.
- [6] C. McNaught and D. R. Vogel, "The converging streams of globalisation and eLearning: Stretching the comfort zone," in *Proceedings of the 21st ASCILITE Conference*, 2004, pp. 637-646.
- [7] I. Cavrak and R. Land, "Taking Global Software Development from Industry to University and Back Again," in *GSD'03 The International Workshop on Global Software Development*, 2003, p. 78.
- [8] J. M. Marquès, L. Navarro, and A. Daradoumis, "WWG: a distributed infrastructure for learning in groups," in *Frontiers in Education Conference, 2001. 31st Annual*, 2001, vol. 2, p. F2C-14.
- [9] L. Layman, L. Williams, D. Damian, and H. Bures, "Essential communication practices for Extreme Programming in a global software development team," *Information and software technology*, vol. 48, no. 9, pp. 781-794, 2006.
- [10] T. Daradoumis, A. Martínez-Monés, and F. Xhafa, "An Integrated Approach for Analysing and Assessing the Performance of Virtual Learning Groups," in *Groupware: Design, Implementation, and Use*, vol. 3198, G.-J. Vreede, L. A. Guerrero, and G. Marín Raventós, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2004, pp. 289-304.
- [11] O. Gotel, V. Kulkarni, D. Phal, M. Say, C. Scharff, and T. Sunetnanta, "Evolving an infrastructure for student global software development projects," in *Proceeding of the 2nd annual conference on India software engineering conference - ISEC '09*, Pune, India, 2009, p. 117.
- [12] P. Layzell, O. P. Brereton, and A. French, "Supporting collaboration in distributed software engineering teams," in *Software Engineering Conference, 2000. APSEC 2000. Proceedings. Seventh Asia-Pacific*, 2000, pp. 38-45.
- [13] C. McNaught, "Supporting the global e-teacher," *International Journal of Training and Development*, vol. 7, no. 4, pp. 287-302, 2003.
- [14] B. E. Munkvold and L. Line, "Training Students in Distributed Collaboration: Experiences from Two Pilot Projects.," 2000.

- [15] D. R. Vogel, R. M. Davison, and R. H. Shroff, "Sociocultural learning: A perspective on GSS-enabled global education," *Communications of the Association for Information Systems*, vol. 7, no. 1, p. 9, 2001.
- [16] J. P. Rodríguez, C. Ebert, and A. Vizcaino, "Technologies and tools for distributed teams," *Software, IEEE*, vol. 27, no. 5, pp. 10–14, 2010.
- [17] F. Lanubile, C. Ebert, R. Prikladnicki, and A. Vizcaíno, "Collaboration tools for global software engineering," *Software, IEEE*, vol. 27, no. 2, pp. 52–55, 2010.
- [18] C. Ebert, *Global Software Engineering: Distributed Development, Outsourcing, and Supplier Management*. Wiley, IEEE CS Press, 2010.
- [19] A. Sarma and A. Van Der Hoek, "Towards awareness in the large," in *Global Software Engineering, 2006. ICGSE'06. International Conference on*, 2006, pp. 127–131.
- [20] G. Almes, J. Cummings, J. P. Birnholtz, I. Foster, T. Hey, and B. Spencer, "CSCW and cyberinfrastructure: opportunities and challenges," in *Proceedings of the 2004 ACM conference on Computer supported cooperative work*, 2004, pp. 270–273.
- [21] W. A. Wulf, "The collaboratory opportunity," *Science*, vol. 261, no. 5123, p. 854, 1993.
- [22] R. T. Kouzes, J. D. Myers, and W. A. Wulf, "Collaboratories: Doing science on the Internet," *Computer*, vol. 29, no. 8, pp. 40–46, 1996.
- [23] G. M. Olson et al., "The upper atmospheric research collaboratory (uarc)," *interactions*, vol. 5, no. 3, pp. 48–55, 1998.
- [24] T. A. Finholt and G. M. Olson, "From laboratories to collaboratories: A new organizational form for scientific collaboration," *Psychological Science*, vol. 8, no. 1, p. 28, 1997.
- [25] M. Bhandarkar et al., "BioCoRE: A collaboratory for structural biology," *Biochemistry*, pp. 242–251, 1999.
- [26] C. P. Lee, P. Dourish, and G. Mark, "The human infrastructure of cyberinfrastructure," in *Proceedings of the 2006 20th anniversary conference on Computer supported cooperative work*, 2006, pp. 483–492.
- [27] S. Lu and J. Zhang, "Collaborative scientific workflows," in *Web Services, 2009. ICWS 2009. IEEE International Conference on*, 2009, pp. 527–534.