Process Support and Knowledge Management for Virtual Teams Doing Agile Software Development

Seth Bowen
Department of Computer Science
University of Calgary
2500 University Drive N.W.
Calgary, Alberta, Canada T2N 1N4
bowen@cpsc.ucalgary.ca

Frank Maurer
Department of Computer Science
University of Calgary
2500 University Drive N.W.
Calgary, Alberta, Canada T2N 1N4
(403) 220-3531
maurer@cpsc.ucalgary.ca

Abstract

Agile practices are arguably improving the productivity of small, co-located software development teams. In this paper, we describe an approach that tries to overcome the constraint of co-location by introducing a process-support environment (called MILOS) that helps software development teams to maintain adaptive practices in a distributed setting. MILOS supports project coordination, information routing, team communication, pair programming and experience management.

1. Introduction

Global computer networks allow developers to collaborate with one another on software projects from remote locations. Virtual teams have shown to be valuable for the development of software that is in common use today, for example the open-source Apache Web Server and Linux operating system. Distributed teams are not constrained by location.

However, virtual teams lose the richness of face-to-face communication, and distribution introduces time delays [2]. Nevertheless, due to market considerations (e.g. lack of locally available skilled resources, cost differences) and the sheer size of projects (e.g. in the telecom industry), distributing software development tasks to several locations is often the only option. Hence, the goal of our research is to provide as much support to the resulting virtual teams as possible.

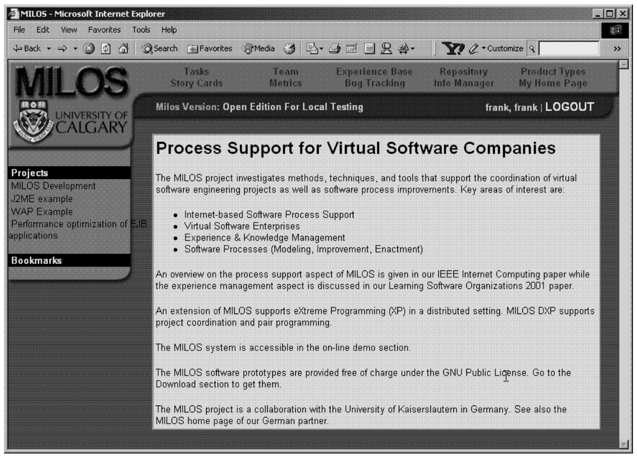
Some applications are available to support individual aspects of distributed software development. Microsoft Project, for example, supports project planning and progress tracking for virtual teams. Version management systems like CVS provide a repository for file sharing. CASE tools like Rational Rose have some capabilities for information sharing. However, these tools are not closely integrated and rather limited in several respects. For example, there is little support for integrating knowledge management into the development process, and they do not support automatic information routing easily. Standard workflow management systems on the other hand are not flexible enough to be used in software development.

The current interest in agile practices, such as Scrum [6] and XP [1], has brought about discussions on the value of using minimally defined, adaptive processes over highly defined processes. These agile practices typically involve using short increments, frequent face-to-face communication, and iterative development to reduce risk and provide increased up-front value to the customer. Development tools are needed that support distributed agile software development processes.

MILOS (Minimally Invasive Long Term Organizational Support) aims at offering support for these agile processes by providing collaboration and coordination technology for distributed software development. It also supports project planning and flexible information routing. The next section outlines the major components of the tool. Section 3 outlines the support for agile practices. Some implementation details are covered in section 4, and then we address conclusions and future work for the system.

2. MILOS Components

MILOS includes integrated components that support the planning and execution of software development, as well as knowledge management. The user interface (see Figure 1) allows access to all MILOS components.



Process Support for Virtual Software Companies

The MILOS project investigates methods, techniques, and tools that support the coordination of virtual software engineering projects as well as software process improvements. Key areas of interest are:

- Internet-based Software Process Support
- Virtual Software Enterprises
- Experience & Knowledge Management
- Software Processes (Modeling, Improvement, Enactment)

An overview on the process support aspect of MILOS is given in our IEEE Internet Computing paper while the experience management aspect is discussed in our Learning Software Organizations 2001 paper.

An extension of MILOS supports eXtreme Programming (XP) in a distributed setting. MILOS DXP supports project coordination and pair programming.

The MILOS system is accessible in the on-line demo section.

The MILOS software prototypes are provided free of charge under the GNU Public License. Go to the Download section to get them.

The MILOS project is a collaboration with the University of Kaiserslautern in Germany. See also the MILOS home page of our German partner.

Figure 1. The MILOS user interface.

2.1. Workflow Engine

The project workflow is modelled using processes and variables, which can be thought of as tasks and work products, respectively. Each task has a required skill set that can be used when developers are assigned to tasks. Tasks and variables can also be associated with information needs objects to provide additional information for doing the task. For example, a task “implement server component using EJBs” could be linked to information needs models that allow the retrieval of the EJB (Enterprise Java Bean) specification or an EJB tutorial. Access to these items will only be provided to users whose skill levels match the information needs profile.

During project execution, each user has a to-do list for managing his or her current tasks. The list is updated when events that affect a user’s tasks are triggered, such as when the input to a user’s task is modified, or the schedule of a task changes.

The workflow structure is highly flexible because processes can be added or altered at any time. Furthermore, process models and associated parameters can be incorporated from the experience base. The workflow engine is tightly integrated with the experience base as a means to reuse process models.

2.2. Experience Base

The experience base is the MILOS knowledge management centre. Processes and process decompositions can be extracted from working projects to create process models, and then associated with tasks in the workflow engine to help in project planning. Process models are represented hierarchically. Each process can have one or more methods of execution, which means if there is more than one method (= process decomposition) for a process than there are alternative ways to carry out the process. For example, the testing process could involve using the following methods: black box, white box, or both. During project execution, the manager will then make a decision how to execute the process by selecting a method or by defining one on the fly.

The information assistant provides context-specific information to users. For example, information (e.g., tutorials) can be retrieved based on a user’s skill set, or the skill set related to a task. Furthermore, information could be retrieved that is required for a task but is not part of the user’s skill set. Little user intervention is required because relevant information can be pushed by the system, although the user can also initiate the retrieval of information (i.e., pulling information).

2.3. Resource Pool

The resource pool contains agent (= team member) and team profiles. An agent profile includes a user’s contact information and skill set. A user can query the skill sets of other users when looking for people with certain skills. As mentioned previously, the information assistant also makes use of user skill sets.

3. Agile Practices

MILOS supports agile software development with the use of some Extreme Programming (XP) practices [4]. During the planning game, user stories for eliciting the requirements can be created and then modified during the development life cycle. A development schedule of releases, iterations, user stories, and tasks is maintained by the system. MILOS also supports the coordination and initiation of pair programming sessions with Microsoft NetMeeting.

MILOS gathers some metrics about the agile project. The velocity is the measure of how much work was accomplished during an increment [5]. The number of lines of source code, and the number of classes and methods are fine-grained measurements to help evaluate the productivity of the team and compare it to the number of tasks or user stories completed. MILOS includes a metrics utility that produces size and complexity metrics for packages, classes, and methods.

4. Implementation Details

MILOS is a collaborative effort between the Software Process Support Group at the University of Calgary (U of

C) and the Artificial Intelligence Group at the University of Kaiserslautern (U of KL). Two versions are being developed separately. Both are written in Java, and so can be deployable on several platforms, including Windows, Macintosh, Solaris, and Red Hat Linux [3]. The U of KL version uses a Java GUI interface and Gemstone as the object-oriented database.

The U of C version's interface uses HTML and so can be accessed using a common Web browser. EJBs are used to map objects to a relational database.

No licensing fees will be required for deploying MILOS because it will be tested to run using the following open-source applications: Apache Tomcat as the Web server, JBoss as the J2EE server, and MySQL as the relational database. Of course, other suitable J2EE application servers can be used in place of the aforementioned products for deployment.

5. Conclusions and Future Work

MILOS is an open-source application that provides process support and experience management for agile software development in a distributed environment. The application is under continual development. Current work is being done on the following items: a skill-based ontology so that searching is more flexible, and changing the client-server architecture to a peer-to-peer architecture to allow separate groups to collaborate on one project without having to sacrifice control of intellectual property.

MILOS is accessible at http://sern.cpsc.ucalgary.ca/~milos

6. References

[1] K. Beck. *Extreme Programming Explained: Embrace Change*. Addison Wesley, New York, N.Y., 2000.

[2] J.D. Herbsleb, T.A. Finholt, and R.E. Grinter, "An Empirical Study of Global Software Development: Distance and Speed", *Proc. 23rd International Conference on Software Engineering*, IEEE Computer Society, 2001.

[3] Java 2 Standard Edition, v 1.4.0 Download, http:// java.sun.com/j2se/1.4/download.html (current May 2002).

[4] F. Maurer and S. Martel, "Process Support for Distributed Extreme Programming Teams", http://sern.ucalgary.ca/ ~milos/papers/2002/MaurerMartel2002.pdf (current May 2002).

[5] Project Velocity, http://www.extremeprogramming.org/ rules/velocity.html (current May 2002).

[6] K. Schwaber and M. Beedle, *Agile Software Development with Scrum*, Prentice-Hall, Upper Saddle River, N.J., 2002.