

IBM Research Report

Jazz: A Collaborative Application Development Environment

Li-Te Cheng, Susanne Hupfer, Steven Ross, John Patterson

IBM Research Division
Collaborative User Experience Group
One Rogers Street
Cambridge, MA 02142

Bryan Clark

Clarkson University
P.O. Box 5815
Potsdam, NY 13699-5815

Cleidson de Souza

Information and Computer Science
University of California, Irvine
444 Computer Science Building
Irvine, CA 92697-3430



Research Division

Almaden - Austin - Beijing - Delhi - Haifa - India - T. J. Watson - Tokyo - Zurich

Jazz: A Collaborative Application Development Environment

Li-Te Cheng, Susanne Hupfer, Steven Ross, John Patterson (Collaborative User Experience Group, IBM Research),
Bryan Clark (Clarkson University), Cleidson de Souza (University of California, Irvine)

Introduction

Over the past several years, software developers have integrated collaborative features into a variety of applications. Despite the fact that software development is itself a collaborative activity, development tools rarely offer direct support for collaboration. Usually collaboration occurs via the source control repository or in applications outside the IDE (e.g. email, instant messaging). As software development teams work under mounting pressure to meet tight deadlines, and as these teams are increasingly geographically distributed, the need for tools to enhance their effectiveness is growing. In this demo, we will show the current working version of Jazz, a project at IBM Research aimed at extending development tools with collaborative capabilities, enabling small teams of software developers to work together more productively.

The Jazz Metaphor

Jazz is based on the metaphor of an “open office” approach to application development. In such a setting, a team of developers works in close proximity at individual workstations, with communal space set aside for team meetings, shared whiteboards, and schedule information. A key aspect of this setting is team awareness. Even while focusing on their personal work, team members have a sense of the work, activities, and conversations occurring elsewhere in the room. Communication among team members is encouraged in this environment, whether shouting out questions to the team or calling over a colleague to consult on an issue at a particular workstation.

The Jazz project extends the Eclipse Java development environment to capture these collaborative features of an open office environment. Our goal is to elevate the core team – a small egalitarian group of developers with common goals and implicit trust in one another – to a first-class object in the development environment, and to facilitate awareness, communication, and coordination among the members of the team.

Elements of Collaboration in the Jazz Environment

The Jazz enhancements to the Eclipse environment focus on increasing the user’s awareness of people, resources, and activities, and on fostering communication among team members.

The most visible aspect of the Jazz enhancements to Eclipse is the *Jazz Band*, a small strip along the bottom of the screen that provides the user with peripheral awareness of the other developers. A small icon in the Jazz Band represents each team member. At a glance, one can tell whether a team member is online, active at his/her workstation, and engaged in his/her own Jazz-enhanced Eclipse environment. Clicking on or hovering

over icons reveals further details about developers' activities and the files they have checked out of source control.

The Jazz Band provides easy access to and awareness of the *Team Jam*, a team-centric discussion board. A user can leave a message or question for the entire team on the Team Jam, or can initiate an instant-messaging style chat with a particular team member through the Jazz Band. At the conclusion of any such chat, the chat contents may optionally be posted to the Team Jam for the benefit of the other team members. Both synchronous chats and asynchronous messages in the Team Jam can be associated with a place in a code file, allowing easy linkage from the code to the discussion or from the discussion to the code. The Team Jam also provides access to a log of team events, such as the check-ins and check-outs of files from source control.

Beyond asynchronous discussions in the Team Jam and synchronous chat discussions, Jazz will also allow a user to share his/her screen with other team members to support activities such as pair programming, joint debugging, code walkthroughs, and consulting.

In addition to people-centered awareness, the Jazz project provides resource-centered awareness through extensions to the Eclipse package explorer. Decorators on the file names in the package explorer indicate which files are checked out, which have been subsequently modified, and which have been checked in, rendering the current files on the workstation out-of-date. An author property added to the explorer shows which individual team members have stakes in particular files. Resource-centered awareness continues downward from the package explorer into the code editor itself. Markers along the left margin of the code indicate where other team members have made modifications. Other markers show the location of code-based annotations left by a team member, and also code-based discussions between developers about a particular region of code. The contents of these annotations and chats are easily accessible through these markers. In this manner, developers maintain awareness of where other team members are working in the codebase and what issues center on particular areas of code. In this manner, they can coordinate their efforts and avoid conflicts.

Future Directions

The goal of the Jazz-enhanced Eclipse environment is to provide easy, in-context access to as much of the team information as possible, without hindering the user's work unnecessarily. Annotations, questions, answers, chats, screen shares, and file reservations are all stored in the Team Jam, and are also accessible from editor markers, personal icons on the Jazz Band, and the Package Explorer. We will be experimenting with cross-linking these artifacts, so that, for example, file reservations will specify an associated task, tasks can refer to locations in the code, code can contain references to annotations, questions, and answers, etc. The goal is to produce a web of related information that knits the members of the team into a well-coordinated, productive working group. We have also conducted a series of interviews with professional developers discussing the management of interdependencies across teams and gathered their reactions to Jazz. We plan to publish the results of this study and use them to inform the design of our next iteration of Jazz.