Argumentation Tools in a Collaborative Development Environment

Furio Belgiorno, Ilaria Manno, Giuseppina Palmieri, and Vittorio Scarano

ISISLab, Dip. di Informatica ed Applicazioni "R.M. Capocelli"
Università di Salerno, Fisciano (SA), 84084, Italy
{furbel,manno,palmieri,vitsca}@dia.unisa.it

Abstract. We describe how we enhanced Rational Team Concert (an IBM CDVE built on Eclipse and Jazz) with structured communication tools by integrating a collaborative platform named CoFFEE, that was developed for structured argumentation and discussion in an educational setting.

1 Introduction

The increased availability and accessibility of network resources and the research of specific competences around the globe are crucial motivations that are pushing the increase of software development teams distributed around the world. In spite of plentiful technological means to communicate, to exchange information/documents and, in general, to share resources, there is still a wide margin of improvement in the collaboration software development, as both the academic and industrial research shows [1–4].

As described in [5], most of the existing Collaborative Development Environments aims to support mainly the collaboration in the *coordination* and *control* activities[1] such as Sourceforge, GForge and TRAC.

Another research direction is exploring the best way to integrate *communication* tools within the development environments [7], and the main examples are CollabVS and Rational Team Concert, development environments which integrate also communication functionalities in the development process.

CollabVS [8] is a Microsoft research project providing a Visual Studio extension that augments the user experience with functionalities aimed at collaborative, distributed development. CollabVS is an effort to introduce collaboration and multiparty, distributed software development enhancements into Visual Studio. Real-time Presence information compensates for the team members not being in the same room, to see what's going on by creating a virtual environment. Communication tools compensate for the team members not being in the same room to talk to each other (an instant messaging session, an audio/video session, or a shared whiteboard). Collaborative development tools change the user experience by augmenting the traditional IDE with support for collaborative activities. These tools aim at providing support for collaborative software construction.

[1] See [6] for the definition of coordination, control and communication.

Rational Team Concert [9] is an IBM product providing a development environment built on Eclipse and Jazz (an extensible platform providing services to support collaboration) integrating advanced control and coordination collaborative functionalities. The choice of leveraging RTC on open source projects (Eclipse and Jazz) allows and stimulates the creation of extensions and research projects based or integrated with it (see, for examples, the list in [10], FriendFeed [11, 12] and Ensemble [13]).

In this paper we present the integration of synchronous communication tools to support structured *argumentation* within Rational Team Concert. The collaborative argumentation tools come from the CoFFEE suite [14, 15], developed for the European-funded project LEAD [16].

The motivation to our work was the consideration of evidence of lack of support for synchronous and structured collaboration in CDVEs. In fact, the synchronous communication available today is often represented either by simple tools (chat) or by shared audio/video/whiteboard communication tools. As an example, RTC is a team collaborative development environment that provides wide support to the teams, by integrating tasks across the software lifecycle. What is missing, currently, is the ability for the team to (first) discuss[2] and debate (with advanced synchronous tools) during the development process by using the platform, and (second) to be able to include the discussions into the project repository in such a way that it builds into the team knowledge. In a way, the development process in RTC is enriched, with our extension, by the interactions among team members, that are, indeed, part of the process, but rarely stored/used. The advantage is that one can re-use previous successful solutions (as design patterns) by accessing also the process (the discussion) that led to design, adopt, and deploy the solution.

2 CoFFEE and Rational Team Concert

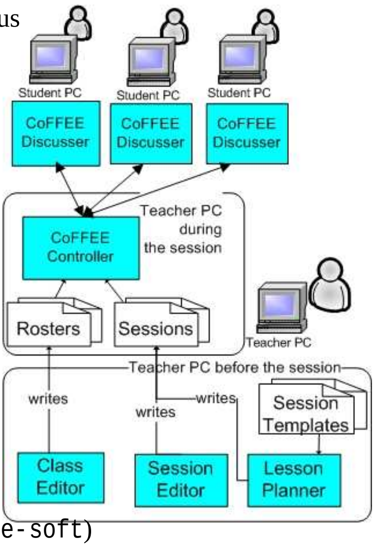
CoFFEE (Collaborative Face-to-Face Educational Environment) is a face-2-face cooperative environment composed of several applications supporting each phase of the collaborative learning process.

CoFFEE is a suite of applications: the Class Editor, the Session Editor, the Lesson Planner, the CoFFEE Controller and the CoFFEE Discusser.

The CoFFEE Controller (launched by the teacher) and the CoFFEE Discussers (launched by learners) are the applications developed to support the face to face collaboration in the classroom, which provides access to a predefined set of collaborative tools (possibly organized in pre-assembled sessions).

Several tools are available and we focus on the Threaded Discussion tool and the Graphical Discussion tool. They may help learners to represent the problem space or to organize their interactions. This kind of support is expected by the possibility to connect related contributions so that the discussion can be organized in a logical order that reflects learners' reasoning. The Threaded Discussion tool allows synchronous messaging between the users, structuring the contributions in threads to improve improving the organization of debates going beyond the temporal sequence and highlighting the relationships among related arguments.

[2] RTC offers limited support for communication, since only simple instant messaging, interfaced with existing Lotus SameTime or Jabber server, is available.



The Graphical Discussion tool allows synchronous messaging among the users, representing the contributions as boxes in a graphical space, eventually linked by several kinds of arrows. This tool is designed to support brainstorming processes and conceptual maps creation, but it is enough generic and malleable to satisfy other usage scenarios.

However, CoFFEE has been designed also to allow the integration of new tools, to evolve following new needs and expectations and currently offers several other collaborative tools providing a variety of collaborative functionalities. Each tool offers a set of configuration options which make possible the tool customization to address several usage scenarios and to pursue several aims. Many more details can be obtained by the project website (`http://sourceforge.net/projects/coffee-soft`) and by the literature available in [14, 15, 17].

Rational Team Concert is a team collaborative development environment that provides wide support to manage software projects by integrating tools to improve the exchange information and tools to support management activities and collaboration (like teams managements, projects areas, work planning, source control, integrated reporting, process support).

RTC provides users with several interfaces: an Eclipse-based client, a Microsoft Visual Studio client and a Web interface. The client interfaces provide developers with a rich, integrated development environment for building and delivering artifacts. The Web interface is well suited for server and project administration and allows users to access project areas, browse repository information, update tasks, or read about recent events.

RTC is the delivered on the Jazz technology platform. Jazz is a scalable, extensible team-collaboration platform that integrates tasks across the software lifecycle. The platform also provides useful building blocks and frameworks that facilitate the development of new products and tools.

Jazz includes an extensible **repository** that provides a central location for tool-specific information. Data is stored in the repository in terms of top-level objects called items.

The repository includes auditable item types, which maintain a history of an item's creation and subsequent modification for audit purposes. The audit trail includes a record of past states of the item, the user who saved the item, and the time of the change. For item types which do not require audit history, the repository retains only the latest state of the item.

The **project area** is the representation of a software project. The project area defines the project deliverables, team structure, process and schedule. A project area is stored as a top-level or root item in the repository. A project area references project artifacts and stores the relationships between these artifacts. Access to a project area and its artifacts is controlled by permissions.

A project can be quite simple or complex in terms of its committed product deliverables, process, and schedules. An established project can have multiple active lines of development, known as timelines. A timeline represents an area of activity within a project that typically has its own schedule, deliverables, teams, and process. For example, a project with both new product release development and current product maintenance might define these two efforts in separate timelines because they have different delivery schedules, teams, and processes.

The structure of the project teams is defined by one or more team areas. Complex projects can have a hierarchy of team areas. Typically, one or more teams are assigned to each line of development. Users might have multiple assignments that require them to work on more than one team. Team areas are optional. A simple project with a small number of users might not need separate team areas.

Project process is the collection of roles, practices, rules, and guidelines used to organize and control the flow of work. The project process is defined in a project area and can be further customized in a team area. In Jazz, you use process to define user roles and their permissions for performing operations within the tool, such as changing the state of a work item.

RTC includes process templates that you can adopt and customize. Process templates provide a starting point and guide for a project area process configuration and iteration structure. Your team can begin with a simple process and evolve it as the project progresses.

3 Argumentation tools into Jazz RTC

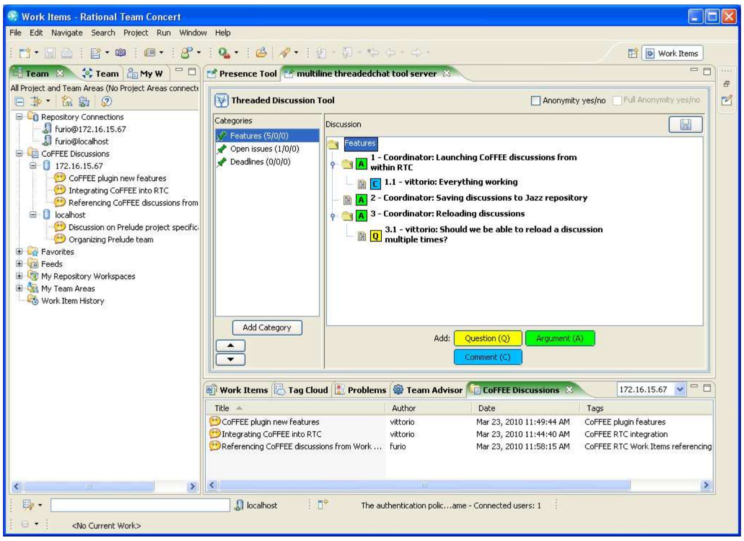
The integration of RTC with CoFFEE grants the possibility of having a strong support to synchronous discussions, involving two or more users, taking advantage of the variety of CoFFEE collaboration tools, and to save the discussion to the Jazz repository. The latter functionality is very important, because it allows to re-analyze the discussion which led to a decision, and also to re-open an old discussion keeping memory of everything was said before. Furthermore, different users might want to access other people's discussions about a problem they are facing, and this could be done with a simple repository search.

Architecture. We have created an eclipse plug-in which allows to use the CoFFEE tools within eclipse. Adding it to RTC, which is an eclipse-based application, is straightforward, but we had to modify our generic CoFFEE plug-in to load/save the discussion from/to the Jazz repository.

To allow the interaction between the CoFFEE plug-in and the Jazz repository, we have implemented a Jazz component, which consists of a service and a client library which communicate to each other via SOAP/XML.

Finally, we had to create a new Jazz *artifact*, to render the CoFFEE discussions in the Team Artifacts view and to allow some common operation such as adding to the Favorites folder and creating references from Work Items.

User Interface. A CoFFEE toolbar has been added to the RTC toolbar to allow the launch of a CoFFEE discussion from within RTC. The CoFFEE tools run inside the user interface of the RTC application. The discussions are saved to the Jazz repository and represented as artifacts: they are visible in the Team Artifacts view in a tree format,



A 1 - Coordinator: Launching CoFFEE discussions from within RTC

C 1.1 - vittorio: Everything working

A 2 - Coordinator: Saving discussions to Jazz repository

A 3 - Coordinator: Reloading discussions

Q 3.1 - vittorio: Should we be able to reload a discussion multiple times?

Title	Author	Date	Tags
CoFFEE plugin new features	vittorio	Mar 23, 2010 11:49:44 AM	CoFFEE plugin features
Integrating CoFFEE into RTC	vittorio	Mar 23, 2010 11:44:40 AM	CoFFEE RTC integration
Referencing CoFFEE discussions from Work …	furio	Mar 23, 2010 11:58:15 AM	CoFFEE RTC Work Items referencing

Fig. 1. Discussion with the Threaded Discussion Tool. The CoFFEE Discussions view is also shown at the bottom and discussions are also shown in the Team Artifacts tree at left.

grouped by repository, and in a dedicated view, in a list format, where further details are shown (author, creation date, and tags, in addiction to the title). Like other artifacts, saved discussions can be added to the Favorites folder, and organized in sub-folders by the user. Other operations allowed on the saved discussions are loading, deleting, and saving to the file system. This allows to create, execute and manage CoFFEE discussions all inside Rational Team Concert. Artifacts can be referenced from inside the RTC Work Items, but an easy support to this feature for CoFFEE discussions is currently under development.

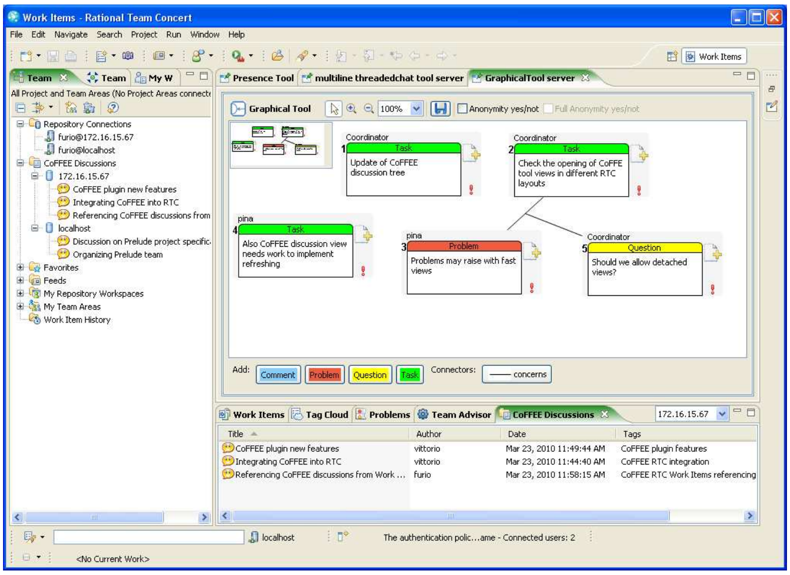
A CoFFEE discussion inside RTC. Let us show a small example of a discussion in a work team using CoFFEE within RTC.

The user Furio wants to start a discussion, so he clicks the CoFFEE toolbar menu and chooses ‘Tools List’; now he can choose to use a dedicated Eclipse perspectivefor the discussion, or to visualize the CoFFEE tools inside the currently active perspective. Then he fills a small form in which he indicates the argument (“Next Jazz-CoFFEE release”), and the discussion starts with Furio as coordinator. He selects the Threaded Discussion Tool as first CoFFEE tool to use, editing or loading the desired configuration for the tool (a default configuration is available as well). In our case, he decides that contributions can be tagged as *Question*, *Argument* or *Comment*.

Other users, Vittorio and Pina, join the discussion by clicking the ’Client Start’ command, and choosing the discussion among the available ones (they see a list with the coordinator’s name and the title for each active discussion). They receive all the tools launched by the coordinator, with all their contents. Now they can interact with the discussion. If the coordinator launches other tools, they automatically appear on the clients’ UI.

The coordinator adds the main discussion topics with ’Add Category’ button (in our example: ’Features’, ’Open issues’, ’Deadlines’), each of which has its own discussion tree. The threaded discussion allows to organize the contributions in the correct logical way, so that every contribution can be appended to the one it refers to. Furthermore, the tags on the contributions give an immediate idea of the contribution intents.

After this preliminary discussion, the coordinator starts a new tool, the Graphical tool, editing the configuration so that the contribution can be tagged as *Task*, *Question*, *Problem*, *Comment*, and with a single type of connector which means *concerns*. In this tool, the discussion can be organized as a concept map. In the example, the team members use the tool to organize the tasks, underlining the possible problems and adding their comments (Figure 2). At the end, the coordinator starts the ’Co-Writer’ tool (a shared editor with turn-taking mechanism) to write down (the draft of) a final document on what to do for the next release (Figure 3).



Coordinator
1 Task
Update of CoFFEE discussion tree

Coordinator
2 Task
Check the opening of CoFFE tool views in different RTC layouts

pina
4 Task
Also CoFFEE discussion view needs work to implement refreshing

pina
3 Problem
Problems may raise with fast views

Coordinator
5 Question
Should we allow detached views?

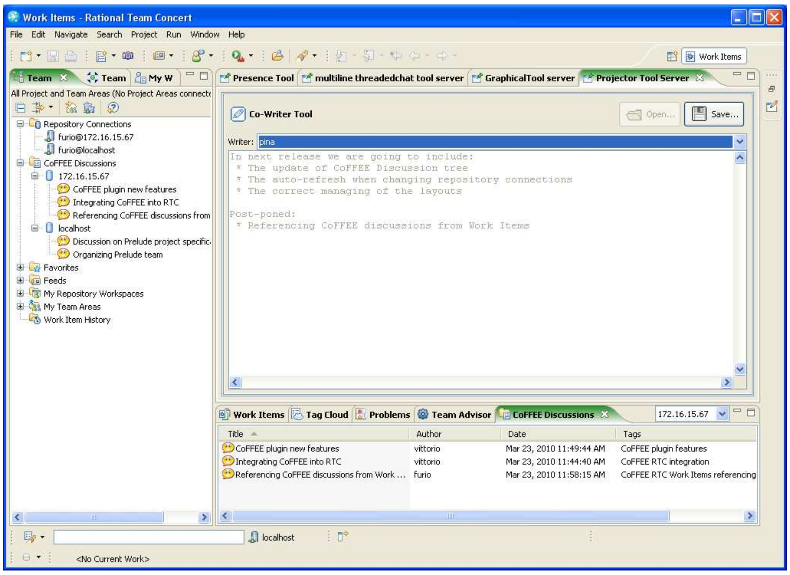
Title	Author	Date	Tags
CoFFEE plugin new features	vittorio	Mar 23, 2010 11:49:44 AM	CoFFEE plugin features
Integrating CoFFEE into RTC	vittorio	Mar 23, 2010 11:44:40 AM	CoFFEE RTC integration
Referencing CoFFEE discussions from Work ...	furio	Mar 23, 2010 11:58:15 AM	CoFFEE RTC Work Items referencing

Fig. 2. Discussion with Graphical Tool.

Finally, the coordinator closes the discussion by clicking the ’Server Stop’ command. The discussion is closed, the CoFFEE tools disappear from the coordinator’s user interface and from the clients’ UI too. Now the coordinator is prompted for saving the discussion: he specifies a title for the discussion (say “Release 1.1”), some tags useful for a successive search, and the repository to save the discussion to.

4 Conclusions and Future Works

There are several tools in CoFFEE that can help the development, besides the discussion tools. For example, the synchronized WWW browser can be used to access public documentation site or support forums, during the discussion, the Shared (programmer)



Writer: pina

```
In next release we are going to include:
 * The update of CoFFEE Discussion tree
 * The auto-refresh when changing repository connections
 * The correct managing of the layouts

Post-poned:
 * Referencing CoFFEE discussions from Work Items
```

Title	Author	Date	Tags
CoFFEE plugin new features	vittorio	Mar 23, 2010 11:49:44 AM	CoFFEE plugin features
Integrating CoFFEE into RTC	vittorio	Mar 23, 2010 11:44:40 AM	CoFFEE RTC integration
Referencing CoFFEE discussions from Work ...	furio	Mar 23, 2010 11:58:15 AM	CoFFEE RTC Work Items referencing

Fig. 3. Using the Co-Writer Tool.

Editor [18] supports collaborative editing and the Positionometer supports voting (also with anonymity included).

Our solution is a significant improvement over the synchronized communication tools currently available in RTC: not only we offer a wide variety of tools, with support to elaborated and structured interactions, but we also do not rely on external servers, as (on the contrary) even the simple chat available in RTC does (since it needs a Jabber or Lotus Sametime server which the clients connect to). But our work offers also a simple platform to develop and include new collaboration tools in RTC: any new CoFFEE tool can be easily included in RTC and simple Eclipse wizards exist (available on CoFFEE Sourceforge website) to support the development of new CoFFEE tools.

Our work has been conducted not without some difficulties: as stated by a group of IBM researchers [19] (from Watson, Haifa and China IBM Research Centers, with experiences also in developing Enseble [13]), the lack of well structured and complete documentation is among the difficulties in extending RTC. It has been sometime very complex and time consuming to identify correct patterns for extending some functionalities in RTC, often solved only by "educated guessing", supported by some experience on the Eclipse platform and a bit of intuition.

Work currently under development includes linking directly from work items toward CoFFEE discussions, so that tracing the process where actions on the projects were planned is more immediate.

Acknowledgments: The authors gratefully acknowledge IBM for partially supporting, with the IBM Jazz Innovation Award 2008, the research described in this paper.

References

1. Aydin, S., Mishra, D.: A tool to enhance cooperation and knowledge transfer among software developers. In: CDVE. (2009) 257–260

2. Sengupta, B., Chandra, S., Sinha, V.: A research agenda for distributed software development. In: ICSE ’06: Proceedings of the 28th international conference on Software engineering, New York, NY, USA, ACM (2006) 731–740
3. Cheng, L.T., de Souza, C.R., Hupfer, S., Patterson, J., Ross, S.: Building collaboration into ides. Queue **1**(9) (2004) 40–50
4. Booch, G.: Introducing collaborative development environments. Technical report, IBM Rational (2006)
5. Lanubile, F.: Collaboration in distributed software development. In: ISSSE. (2008) 174–193
6. Carmel, E., Agarwal, R.: Tactical approaches for alleviating distance in global software development. IEEE Software **18**(2) (2001)
7. Hupfer, S., Cheng, L.T., Ross, S., Patterson, J.: Introducing Collaboration into an Application Development Environment. In: In Proc. of CSCW 04. (2004)
8. CollabVS: Collaborative Development Environment using Visual Studio. http://research.microsoft.com/en-us/projects/collabvs/
9. IBM: Rational team concert. http://www-01.ibm.com/software/awdtools/rtc/ (2010)
10. Jazz Community: Research Projects. `http://jazz.net/community/academic/relatedResearchProjects.jsp` (2010)
11. Calefato, F., Gendarmi, D., Lanubile, F.: Embedding social networking information into jazz to foster group awareness within distributed teams. In: SoSEA ’09: Proceedings of the 2nd international workshop on Social software engineering and applications, New York, NY, USA, ACM (2009) 23–28
12. Calefato, F., Gendarmi, D., Lanubile, F.: Adding Social Awareness to Jazz for Reducing Socio-Cultural Distance within Distributed Development Tools. In: Proc. of the 4th Italian Workshop on Eclipse Technologies, Sept. 28-29, 2009, Bergamo (Italy). (2009) 17–28
13. Xiang, P.F., Ying, A.T.T., Cheng, P., Dang, Y.B., Ehrlich, K., Helander, M.E., Matchen, P.M., Empere, A., Tarr, P.L., Williams, C., Yang, S.X.: Ensemble: a recommendation tool for promoting communication in software teams. In: RSSE ’08: Proceedings of the 2008 international workshop on Recommendation systems for software engineering, New York, NY, USA, ACM (2008) 1–1
14. De Chiara, R., Di Matteo, A., Ilaria Manno, Scarano, V.: CoFFEE: Cooperative Face2Face Educational Environment. In: Proceedings of the 3rd International Conference on Collaborative Computing: Networking, Applications and Worksharing (CollaborateCom 2007), November 12-15, 2007, New York, USA. (2007)
15. Belgiorno, F., De Chiara, R., Manno, I., Overdijk, M., Scarano, V., van Diggelen, W.: Face to face cooperation with CoFFEE. In: Proceedings of 3rd European Conference on Technology-Enhanced Learning (ECTEL 08), September 16-19 2008, Maastricht, The Netherlands. Lecture Notes in Computer Science, Springer-Verlag (2008)
16. LEAD: Lead project: Technology-enhanced learning and problem-solving discussions: Networked learning environments in the classroom, 6th Framework Programme Priority IST. `http://lead2learning.org/` (2010)
17. Belgiorno, F., De Chiara, R., Manno, I., Scarano, V.: A Flexible and Tailorable Architecture for Scripts in F2F Collaboration. In: Proceedings of 3rd European Conference on Technology-Enhanced Learning (ECTEL 08), September 16-19 2008, Maastricht, The Netherlands. Lecture Notes in Computer Science (5192), Berlin, Heidelberg, Springer-Verlag (2008) 401–412
18. Fierro, A., Manno, I., Vitale, P.: Awareness in un Plug-in per Shared Editing. In: Proc. of the 4th Italian Workshop on Eclipse Technologies, Sept. 28-29, 2009, Bergamo (Italy), Eclipse Italian Community (2009)
19. Cheng, P., et al.: Jazz as a research platform: experience from the Software Development Governance Group at IBM Research. In: Proc. of First International Workshop on Infrastructure for Research in Collaborative Software Engineering (IRCoSE) at FSE 2008. (2008)