Visualisations for Team Learning:
Small Teams Working on Long-Term Projects

Judy Kay, School of Information Technologies, University of Sydney, Sydney, NSW 2006, Australia,
judy@it.usyd.edu.au
Kalina Yacef, School of Information Technologies, University of Sydney, Sydney, NSW 2006, Australia,
kalina@it.usyd.edu.au
Peter Reimann, Research Centre on Computer-supported Learning and Cognition (CoCo), University of Sydney, Sydney, NSW 2006, Australia, p.reimann@usyd.edu.au

Abstract: We have developed a set of visualisations mirroring the activity of small teams engaged in a task. These provide a bird's-eye view of what is happening in a small team, giving insights into the way that each individual is contributing to the group and the ways that team members interact with each other. We report on our first experience of using these visualisations for a semester-long software development project course. The study revealed that students, especially those with leadership roles, found the visualizations informative and helpful and that over a third of students modified their behaviour accordingly.

Case Study Context

We are interested in the case of group learning that involves substantial long-term collaboration, over several weeks if not months, where the primary goal is to get a task done (solve a problem, design/develop something) and where the learning effects occur in the context of working on the task. In an educational context, such learning teams can be expected to (a) produce useful artifacts that constitute a contribution to socially shared knowledge (e.g., a problem gets solved), (b) to learn individually—as a side effect—about the domain the problem is contextualized in, and (c) to learn individually—as a side effect—about the team members and to develop knowledge and skill about collaboration management. On the group level, we can expect learning to occur (d) by improved team effectiveness, such as improved coordination of individual activities.

In this paper, we describe our approach to supporting collaboration management for teams that make use of electronic tools like wikis. Support can be established in many forms but, minimally, teams must be provided with the information required for learning along all of the dimensions; in particular, they must be provided with information on team processes, in addition to the task-related information. We have created a set of interrelated visualizations that display the vast amount of information stored in electronic traces such as log files in a format directly addressing team functions (Kay et al., 2006), two of which will be presented here as used in the context of an undergraduate capstone software engineering project based on eXtreme programming (Beck, 2000).

Before describing the abstract form of the visualizations, we introduce the electronic and broader environment that provided the data for our main use of the visualizations. This is a software development project where students work in groups of 5-7 over 13 weeks. Team members tend to focus on the goal of producing a software product that meets their clients' needs, rather than the group management needed to achieve this. Following the Extreme Programming (XP) approach (see http://www.extremeprogramming.org/), students endorse one or more particular XP role such as team leader (who manages the group), tracker (who tracks people's work and ensure that things are progressing as planned), the customer (who is in charge of liaising with the client), the tester (in charge of functional testing), the doomsayer and so on.

To support their tasks and communication, groups are required to use trac, a tool designed for programmers working in teams to build software. It has three, tightly integrated media: (1) a wiki for collaborative editing of web pages for general group communication; (2) an issue tracking system based on so-called tickets, where one team member creates a ticket when a task needs to be done, this is allocated to a team member and, when the task has been completed, the ticket is closed; (3) a browsing interface to a repository based upon the version control system called Subversion (SVN), for storing documents like source code, including any versions.

Visualisations

Figure 1 shows our main visualization, which we call the Wattle Tree. (The Wattle tree is an Australian native plant with fluffy golden-yellow round flowers, similar to this visualization.) Each person in the team appears as a “tree” that climbs up the page over time. The tree starts when the user first does an action on any of the three media considered. The vertical axis shows the day number and date. Wiki-related activity is represented by yellow (bright) “flowers”, circles on the left of the trees. SVN-related activity is similarly represented, as orange (darkish) flowers on the right of the trees. The size of the flower indicates the size of the contribution. Ticket actions are represented by leaves – the green lines: a dark (green) leaf on the left indicates a ticket was open by the user and a (light) green leaf on the right indicates the user closed a ticket. The length of the left leaf is proportional to the time it remained opened. Those still open are shown at a standard, maximal size (e.g. the ones around day 41 in Figure 1).

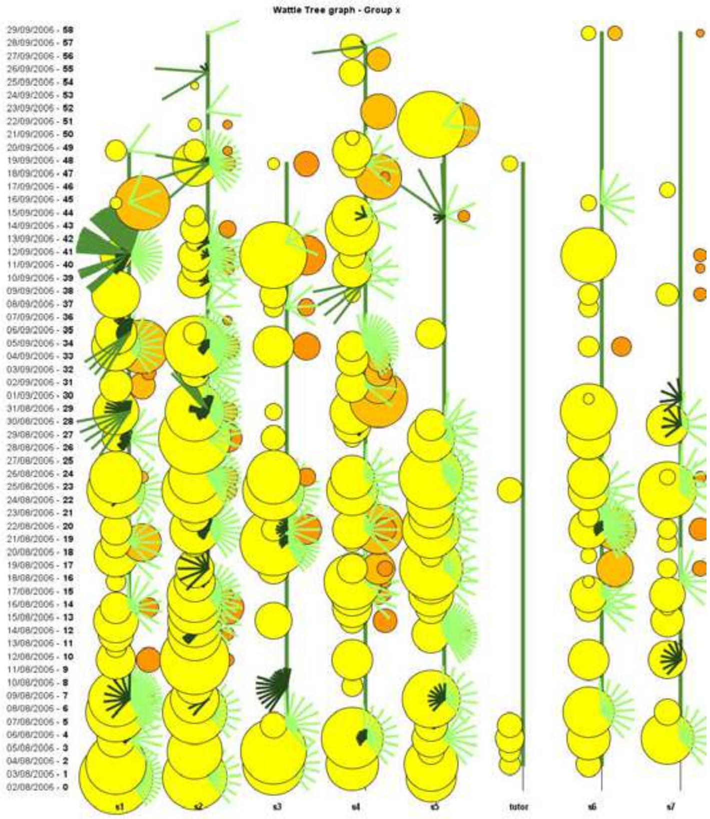

Figure 1.Wattle diagram

Wattle trees do not contain information on who issued tickets to whom, and who contributes to a wiki page. In order to visualize this kind of information, we use what we call an interaction network, inspired by the graphical notations used in Social Network Analysis (Scott, 1991). The network is modeled as a graph based on nodes representing team members, with each team member occupying a fixed position. So, for example, the person at 12 o’clock in Figure 2 and Figure 3 is the same in each of these visualizations. Lines between these nodes indicate interaction between these team members. We define interaction to occur if the pair modifies the same wiki page or SVN file or both perform actions on the same ticket. The width of the edge is proportional to the number of interactions between them. For a given resource, the number of interactions is calculated as n = min(n1, n2) where n1 and n2 are the number of times user1 and user2 modified the resource.

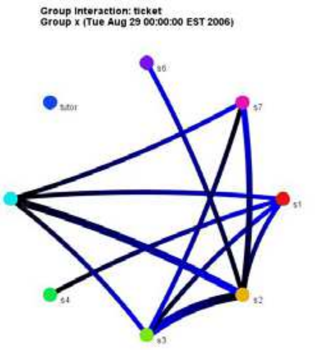
Group Interaction: ticket
Group x (Tue Aug 29 00:00:00 EST 2006)
s6
tutor
s7
s1
s4
s2
s3

Figure 2. Interaction network for tickets

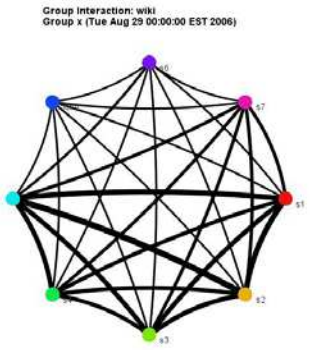

Figure 3. Interaction network for wiki pages

First experiences using the visualisations

We report here experiences from a semester-long project course (capstone project) where teams used trac. There were 7 groups of 5 to 7 students in each team, with a total of 44 students making it to the end. Three of the seven groups showed great enthusiasm for the visualisations and asked to be able to generate them on demand (This is not yet possible). The students indicated that the visualizations were helpful for the tracker (the person who has to ensure that work is progressing as intended) and the manager (who distributes the workload). There has also been spontaneous reference to the visualizations in relation to some difficulties in groups, particularly in the case of seeming occurrences of social loafing, with an individual failing to carry their fair share of work in the group. Students have also stated that the visualisations help individuals to see the amount of work they have contributed to the group, to compare it with that of others and to provide some quantitative measurement for balancing the group workload. Some students explained that they would like to see how the diagrams change after they have contributed a fair amount of work and see how this amount compares with the others. One group mentioned that the lack of contribution from a member showed up on the Wattle Tree. The group liked to see the evidence. The member said he took it as a wake-up call, and intended to participate more. The main negative feedback was related to the fact that the visualizations are based on simple counts of the amount of activity and there is no measure of quality. This is a very valid concern, but we point out that the numbers are interpreted by team members who are very familiar with each others' work.

Conclusions

One of the main contributions of this work is the set of visualizations of activity in small groups working over sustained periods. The Interaction Network visualizations, derived from social network diagrams, are novel in their presentation across the media available in trac. The Wattle trees are new, although they were inspired by Donath (1999). Perhaps the most important contribution of the paper is the insights into how such visualisations are actually used by teams. We found that our teams appear to need to be introduced to these tools. This goes hand in hand with the need to motivate team members to appreciate the importance of collaboration management.

References

Beck, K. (2000) Extreme Programming Explained: Embrace Change. Addison-Wesley Longman Publishing.

Donath, J., K. Karahalios, and F. Vigas, Visualizing conversation. Journal of com-puter-mediated communication, 1999. 4(4). l-75.

Kay, J., Maisonneuve, N., Yacef, K., & Reimann, P. (2006). The Big Five and Visualisations for Team Work Activity. Proceedings of Intelligent Tutoring Systems (ITS06), M. Ikeda, K. D. Ashley & T-W. Chan (eds). Taiwan. Lecture Notes in Computer Science 4053, Springer-Verlag, 197-206.

Scott, J., (1991) Social network analysis: A handbook, London: Sage.

Acknowledgements

This work is carried out under a project funded by the Australian Research Council.