Interaction Rules: their place in collaboration software

Robert Kildare, University of Tasmania School of Computing, Locked Bag 359 Launceston Tasmania 7250 Australia, Email:Robert.Kildare@utas.edu.au

R. N. Williams, University of Tasmania School of Computing, Private Bag 100 Hobart Tasmania 7001 Australia, Email: R.Williams@utas.edu.au

Jacky Hartnett, University of Tasmania School of Computing, Locked Bag 359 Launceston Tasmania 7250 Australia, and Email: J.Hartnett@utas.edu.au

Peter Reimann University of Sydney, Education Building A35, Sydney, NSW 2006, Australia Email: preimann@edfac.usyd.edu.au

Abstract: One major form of social capital that is central to the development of communities is the set of structures devoted to; norms of interaction, making these norms overt and regulating interaction. The creation of these structures is deeply linked to notions of trust, which has been identified as a major factor in the behaviour of successful virtual teams. Because (virtual) teams are complex, the types of interactions that suit a particular team may or may not be predictable. The first author has created Phreda, software that permits group members to create interaction rules in the form of production rules. These rules are then given to an expert system shell that matches these rules against resource use parameters in the group's online collaboration environment. This software as well as results from a first pilot study are described.

Democratic Interaction Rules

What knowledge can a software designer draw upon to understand social interaction and, then, encourage trust and bonding in a virtual environment? Sociological and social psychological analyses help the designer to understand how and why people collaborate. The CSCW and CSCL literature describes the qualities of "successful" teams – those exhibiting optimal performance – as including open communications, conflict management, (Yeatts & Hyten, 1998) trust and risk taking. Trust is both a cause and an effect of healthy team interaction (Fernández, 2004). McGrath's Time, Interaction and Performance theory identifies team behaviour as fulfilling three functions – task or production work, team support and individual support (McGrath, 1991). Yeatts and Hyten call the behavioural patterns norms (Yeatts & Hyten, 1998). The norms apply to behaviours such as attendance, the acceptable level of smalltalk and the level of risk taking. They are often unspoken expectations shared amongst the team. Giddens' structuration theory explains how patterns become established in a community. Structuration is the process of producing and reproducing structures of a social system by 'agency' or use. Structures do not exist without people using them. They are systems of ongoing action being produced and reproduced through time.

These considerations led us to the following design guidelines:

- Since teams are complex systems that develop dynamically over time, it is very hard, if not in principle impossible, to prescribe optimal norms; instead, such norms (collections of interaction rules) need to emerge from the group itself.
- Even if useful for a team, the cognitive load of devising rules and then learning the interface for their implementation would be a major disincentive for team members to engage with the module (Sintchenko & Coiera, 2003). Hence, groups should be 'seeded' (Fischer & Ostwald, 2005) with an initial set of rules and be provided with means to change them as they see fit.
- It is awkward for members of democratic, peer-based groups to "police" the compliance with the rules even if these are self-imposed. The monitoring of compliance should hence be given to an outside "authority", a software component that monitors the interaction rules.

In the next section, the software tool Phreda is described that realizes these design guidelines as a first implementation.

The Software Tool Phreda

Phreda is based on a production rule formalism. Production rules were chosen as the implementation language for interaction rules because they allow the user to attach symbolic labels reflecting the intended meanings to sets of measurable conditions. Measurable conditions are those that can objectively be found in electronically recorded activity traces as stored in a database. The conditions express the state of the collaborative tools in the software, for example the number of times a person logs into the website. The action part of a rule expresses consequences, such as sending messages to specific group members. Actions can also result in predicates describing a group's state at a level higher than the measurables, and the predicates can in turn be used in the condition part of other rules. Allowing team members to freely create, edit and delete rules permits the dynamic expression of meaning and importance to evolve.

For example, the 'give_feedback' rule notifies an individual if they are classifying less than a tenth of their postings as 'feedback' and suggests that they offer more support to their fellow members.

Figure 1 shows how a group member, when logging into to the collaboration environment, sees the consequences of rules firing. These are a number of messages generated by the expert system. The messages are based on the team state - how the group used the collaboration tools available, such as a discussion board and a document upload area.

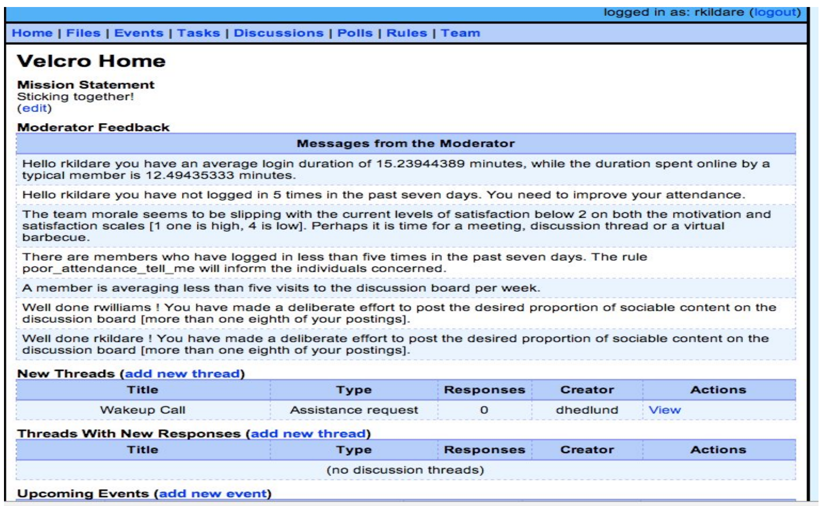

Figure 1: Software home page showing links to tools for managing people, time, artifacts and actions, also the rule management module. The output from the software moderator is displayed under the heading 'Messages from the Moderator' with current discussions, tasks and events below.

Phreda is a composite of a rule editor, a collaboration environment and a server-side rule processor. The collaboration environment contains cut down versions of typical groupware tools and an awareness graphics tool. The rule module is separable and can be used in other projects, with most of the effort being the interface integration of the rule editor. Typically collaboration software has tools to manage people, events, task allocation, communication and storage for personal and team files. These are represented in the trial implementation. Also represented are graphical awareness tools, which are still predominantly in the research domain (Reimann, Kay, Yacef, & Goodyear, 2005). Their effectiveness almost certainly guarantees that the genre will be represented in mainstream software.

Trials to Date

Two teams have used the software to date: a senior secondary team of 6 co-located students and their teacher, and a team of 8 mainly co-located academics Both of the teams were given the opportunity to make changes to the rules or to request that the author make changes. The seed rules were designed to address issues of domination and freeloading and some of the desirable qualities of the communication upon which collaboration depends. Attendance rules were set so that individuals not meeting a benchmark were told so personally, the team told anonymously.

Observations

The team of teenagers has a much larger proportion of 'social' contributions in its discussion board. It is tempting to think that there may be recognisable team types, given the pictures of usage and types of communications. It should be noted that there are 96 individual measures available for team members to use when creating rules, and therefore 96 variables that could define a team's state. There is great potential for theory driven testing.

The first (student) team did spend time looking at the rule module, but neither made, nor requested any changes to the initial seed set. Their teacher observed that the team was helped by the structured format of the system. The inbuilt rules helped them keep - if not to task, then at least regularly logged on and providing messages to each other. The students' questionnaire responses indicated that about half the participants saw value in having the rules.

The academics' team did not request any rule changes either, but made comments that reflected a better understanding of the value of interaction rules. Perhaps the most telling feedback during the four weeks was after members were encouraged to vote their levels of satisfaction and motivation. The rule 'morale_slipping' fired suggesting among other things, a 'virtual barbecue' (details in 'Rules', above). One member commented in the discussion forum: "I must admit to being a little sceptical about the automated rules and the messages they generate. ….However, I now appreciate their value as triggers for human intervention..." This was followed by an attempt by another of the members to organise a face-to-face social event (a non-virtual barbecue). The results of these initial trials of software to support team behaviour when team participants are interacting as a virtual group have been surprisingly encouraging. The trials were of limited duration and size and participants mostly had the option of interacting physically as well. However, the software performed robustly and the size of the trial did not prevent it from constructively affecting behaviour. It seems that the software can not only support research into the performance of general collaborative tasks but also theory driven research, in particular a more rigorous experiment that explores hypotheses related to trust, conflict or performance. There appeared to be some educative value in understanding the importance of interaction rules to the collaborative process as a result of simply having the seed rules present.

As predicted, both teams hesitated to be engaged in the formation and editing of the rules. Cognitive load can be overcome by a rule expert, but heuristic knowledge is scarce. A number of machine learning techniques, however may be useful for ascertaining patterns in rules that correlate with types of teams. A software expert would use the rule sets and the states of many software moderators as the basis for its learning. It would recommend rules. Underlying concepts have been inferred in model-free expert systems in both hierarchic (Richards, 1998) and mesh structures (Suryanto, 2004). Decision trees are also appealing as a means of machine learning symbolic data. The software expert would permit recommendations to evolve, adapting to changes experienced within the team.

References

Fernández, W. D. (2004). Trust and the Trust Placement Process in Metateam Projects. In D. J. Pauleen (Ed.), *Virtual Teams: Projects, Protocols and Processes* (pp. 40-69). Hershey, Pennsylvania: Idea Group Publishing.

Fischer, G., & Ostwald, J. (2005). Knowledge Communication in Design Communities. In R. Bromme, F. W. Hesse & H.Spada (Eds.), *Barriers and Biases in Computer-Mediated Knowledge Communication* (pp. 213-239). New York: Springer.

McGrath, J. E. (1991). Time Interaction and Performance (TIP). A Theory of Groups. *Small Group Research, 22*(2), pp147-174.

Reimann, P., Kay, J., Yacef, K., & Goodyear, P. (2005). *Adaptive Visualisation Support for Self-managed Learning Groups.* Paper presented at the Artificial Intelligence in Education 2005, Amsterdam, Netherlands.

Richards, D. C. (1998). *The Reuse of Knowledge in Ripple Down Rule Knowledge Based Systems.* Unpublished PhD., University of NSW, Sydney Australia.

Sintchenko, V., & Coiera, E. W. (2003). Which clinical decisions benefit from automation? A task complexity approach. *International Journal of Medical Informatics, 70*, 309-316.

Suryanto, H. (2004). *Learning and Discovery in Incremental Knowledge Acquisition: Ripple Down Rules.* Unpublished PhD, University of N.S.W., Sydney, Australia.

Yeatts, D. E., & Hyten, C. (1998). *High Performing Self-Managed Work Teams: A Comparison of Theory and Practice*. Thousand Oaks California: Sage Publications.