Expert Software Support for Ad-hoc Teams - Enabling Online Interaction Rules

Robert Kildare
School of Computing
University of Tasmania
Launceston, Tasmania, Australia
rkildare@utas.edu.au

Ray Williams
School of Computing
University of Tasmania
Sandy Bay, Tasmania, Australia
R.Williams@utas.edu.au

Jacky Hartnett
School of Computing
University of Tasmania
Launceston, Tasmania, Australia
J.Hartnett@utas.edu.au

ABSTRACT

An expert set of interaction rules may help to guide leaderless, transient teams of individuals that work online and asynchronously. Medical and engineering professionals as well as students fall into this category. Sufficient evidence exists to treat such teams as complex systems, to consider team processes as emergent and thus unpredictable. For software to support such teams in a variety of scenarios, the components must cater for the emergence of interaction rules. A software 'Moderator' is proposed which gathers, monitors and executes these rules. The standard AI methodology of expert systems will be used. Symbolic rules link the attribute value pairs that software can monitor with language that is meaningful to the team members. The reuse of rules to form new rules allows for a process of evolution. Multiple Classification Ripple Down Rules (MCRDR) provides a mechanism for team-member control over the evolution of rules. One end result proposed is the collection of the rules developed by similar teams as an expert set of rules for the context in which the teams work. This may fulfil the traditional role of a human expert where one is unlikely to exist.

1. INTRODUCTION

This paper proposes the creation of a software Moderator to support leaderless, ad-hoc online teams. The design of this software is informed by understanding these teams to be complex systems. AI techniques are required to accommodate their unpredictable, emergent behaviour.

Online collaboration between individuals, in order to complete a specific task or to solve a specific problem, is becoming a common occurrence. It often tends to be between professionals who are considered independent. Examples include a number of supervisors combining to oversee a postgraduate student, medical, nursing and paramedical practitioners collaborating for community or client needs and the remote development of software by experts. Consider also students at all educational levels, involved in collaborative learning activities, both formally and informally, with and without instructors or intelligent support, and with varying degrees of face-to-face contact. These ad-hoc, asynchronous, online collaborations take advantage of being able to overcome time zone and scheduling problems.

The software Moderator, it is hypothesised, will provide support for the team by enabling it to create, use and review rules for team interaction, thus educating the users about the nature of effective team interactions as well as supporting the actual operation of the team itself.

In order to understand how to support these loosely coupled teams and evaluate the effectiveness of the support, it is necessary to understand team behaviour. Substantial literature exists on organisations as complex systems [1, 2]and a significant history of research exists on emergent co-operation[3]. The literature implies that teams should be considered complex systems and team processes considered as emergent patterns. Teams have been shown to possess the qualities of complex systems. They are part of and composed of other complex systems, demonstrate non-linear relations between inputs and outputs and display phases of emergent evolutionary behaviour [4]. Baker and Salas suggest that the processes that emerge as teams develop are either task-based processes, team-based processes or a combination of both [5].

Software Support for Collaborative Work teams (CSCW) is readily available. Computer Support for Collaborative Learning (CSCL) has also been an expanding field. Both of these areas have focused predominantly on task or goal support and not on supporting team processes [4]. In the case of the often transient, asynchronous, ad-hoc teams considered in this paper, support for the well-being of the team is arguably more important than providing clever tools for the task. Major risks, including activities such as freeloading [6] and domination [7, 8], may threaten team existence. Having team members learn about team processes can be considered an end in itself, particularly for those about to enter a world of loosely-coupled cyber-collaboration.

The lack of predictability of a complex system and thus the interaction rules generated by a team must somehow be accommodated by the software. The first issue is that the rules of interaction are human concepts which should be represented computationally. The software may set constraints upon just what can be monitored and hence risk binding the rule set to a limited number of options, negating any opportunity for emergence. Hence, the ability for team members to review the rules will be essential.

.Finally, the transient nature of these teams and the variety of contexts in which they occur suggest that expertise in running the teams will be rare. It is perhaps possible to provide a software agent to aggregate interaction rules developed by multiple previous teams, prioritise the rule set and provide suggestions to a particular team as to which rules work best in their situation.

Traditional expert systems and semi-automated methods for harvesting expert knowledge allow these issues to be resolved. The intended application of these systems in the proposed Moderator architecture will be the focus of this paper.

2. DESIGN

The software Moderator is envisaged as a component of a web-enabled, three-tier design intended to facilitate asynchronous teamwork. The client side of the software will contain a number of components for managing communications, file archiving, recording of times worked and indicating personal well-being as well as a tool for mapping task domain concepts. The client will communicate with a database that is accessible to the Moderator. One of the roles of the Moderator will be to provide feedback to the client. Multiple Moderators will then provide rules to be evaluated by an expert software agent.

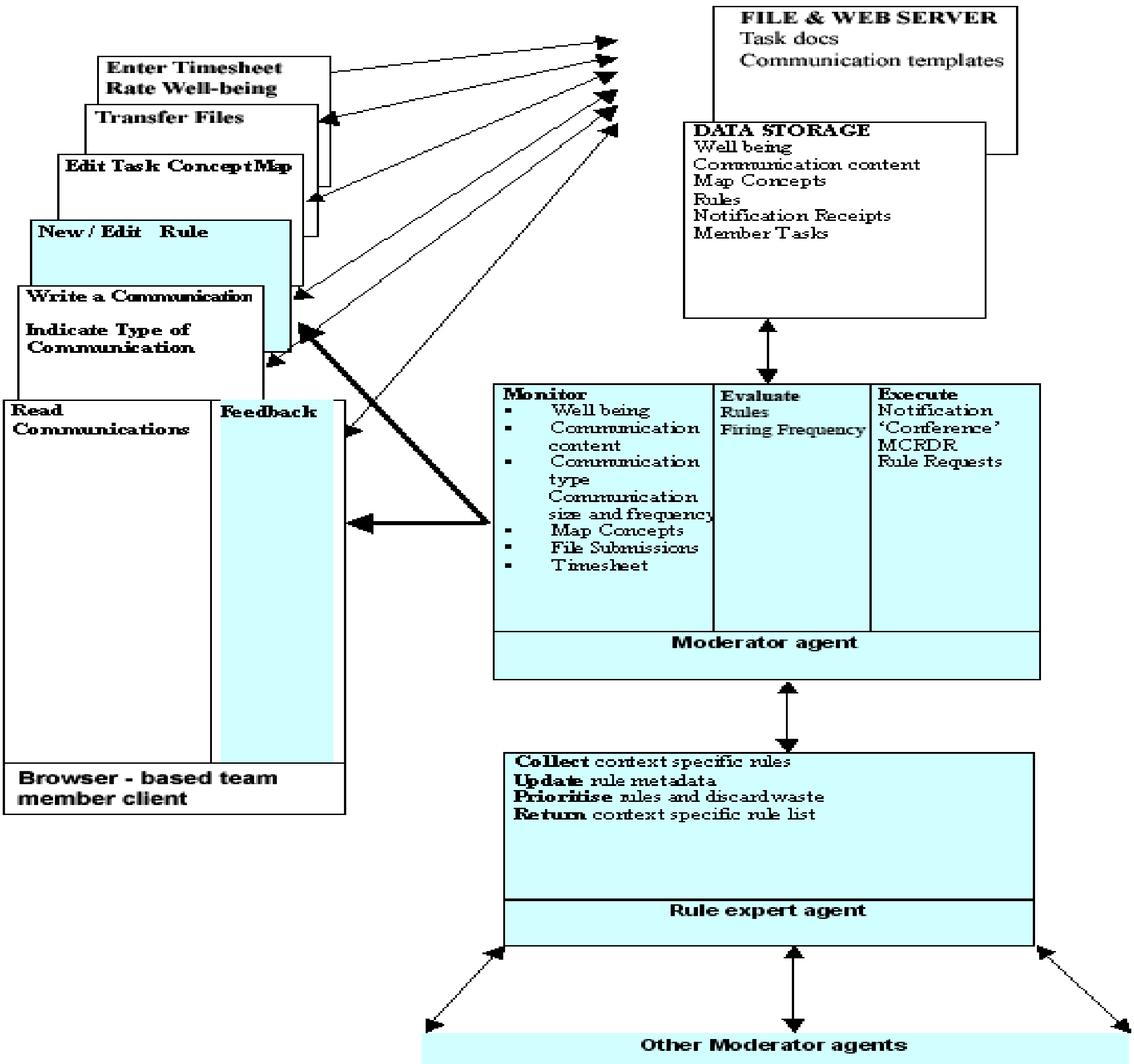

Figure 1: Proposed platform for supporting online asynchronous teams.

2.1. WHAT TO MONITOR

One of the important design questions is to work out how to decide the preconditions that the team members will want to know about, in order to regulate the team processes. The more choice open to members, the greater the potential for rule variety, but the greater the scope for confusion. Design refinement is likely to occur in this area and prototyping is planned. Currently the proposed architecture would permit the recording of the attribute values listed below. The attributes are drawn from quantifiable components recognised to be useful in supporting online interaction. CSCW groupware [9], CSCL domain modelling [10] and "Well-Being" software [11] in particular, have been incorporated into the design. Team members will be expected to use attribute value pairs to create the conditions for interaction rules as discussed in section 2.2.

2.1.1. USERS

The system can identify participants and thus the frequency of contributions and the types of contributions they make. Such contributions would include ideas placed on a concept map, files uploaded to the file server, communications, indications of the well-being of participants and time spent on tasks.

2.1.2. DATES

Significant dates will be identified. Deadlines for submissions and decision-making will be a requirement of most tasks so dates should be associated with all actions, and deadlines should be stored for all subtasks so that comparisons can be made.

2.1.3. FREQUENCIES

In particular the frequency with which a certain rule fires (such as 'Late') may be important. It will also be important for determining whether a particular rule should be edited. The frequency of a particular type of communication or concept contributed may indicate whether an individual is passive, lazy or domineering. Classification of 'types' is discussed below. One position has been put that suggests the frequency of all communications will decrease as the team matures [12]. Aside from any rules team members may wish to create, communications frequency has implications for evaluating the effectiveness of the Moderator as a teaching tool. Reduced frequencies of communication may indicate that the team members are learning to interact. Comparing frequencies of communications with surveys of the team members and other tests, may confirm this position.

2.1.4. SIZES

The size of communications is believed to reduce with team maturity [12]. Again this is important for experimental purposes. Quantity of output may be an issue for team members and therefore file sizes should be available for rule creation.

2.1.5. TIMES

Total times for task completion need to be compared as part of calculations for evaluating team performance and individual performance – needed for measuring the success of the Moderator. It is conceivable that team members may also wish to create rules involving limits to time worked.

2.1.6. TYPES

Types of communications (eg: task content, administration, feedback, assistance, decision and social) could be expressed as frequency distributions for each individual to characterize their contributions. If domain concept nodes are to be given a 'type' property then this could further be used to characterize the contributions of individuals.

The mirroring of online rating surveys of well-being was used by Reimann [11] as a means of informing the team of their motivation levels. Team-members may wish to take some action as a consequence of certain degrees or types of well-being.

2.2. INTERACTION RULES

It is intended that the rules created by the team members take the form of standard production rules [13] . Rule emergence will be possible when team members create "facts" and "hypotheses" from measures of the attributes described above. By ascribing meaning to attribute-value pairs unpredictable outcomes are possible. For example, a frequency value of zero for communication type "off-topic" might be construed as "Optimal performance" or "Antisocial behaviour" depending upon the values of the team. Unpredictable outcomes will be further compounded if participants are given the opportunity to derive further rules from those already created.

The consequences that the system can deliver will be limited to notifications and perhaps alterations to access control lists. It is envisaged that either individual or group notifications will be possible and that the messages (either positive or punitive) will be part of the construction of the rule. It may also be possible to manage the networking necessary to automatically organise an online team conference.

The Java rule engine Jess, created by Friedman-Hill [14], is anticipated to be the major component of the rule creation system and would be integrated into a Servlet environment. One of the benefits of using this inference engine is that it is capable of both forward and backward chaining. The system will have a monitoring task to perform, traditionally carried out using forward chaining. It may be more efficient to search for conditions defined by rules if they are few in number. Jess will permit this issue to be explored further.

2.3. REVIEWING THE RULES

Work by Kang on Multiple Classification Ripple Down Rules [15] demonstrates that it is possible to create a system that obviates the need for a knowledge engineer in the construction of rules. In particular, the MCRDR approach is intended to refine rules that are incorrect. It would seem appropriate that consideration be given to integrating an MCRDR system into both the team member client and the Moderator.

Successful mirroring of the execution of team interaction rules and the ability to refine these rules requires the use of metacognitive learning processes. This approach is often used in educational software [11, 16]. It is hypothesised that the Moderator will act as a means of learning about the rules required for successful online team interaction.

2.4. SOFTWARE EXPERT

It is unlikely that there would be human experts available with appropriate expertise in successful online interaction amongst ad-hoc teams, since the variation amongst team members, between tasks and in team contexts is so great. When there is no overt authority structure in place and this variation exists, no clear patterns of interaction are presented to the team. By contrast, business scenarios have processes repeated and a project leader is usually appointed. Cognito software reflects this with "active methods" for defining interaction and overt roles for a leader [17].

The Prisoner's Dilemma game and the processes of symbiosis versus competition have been the focus of numerous studies in the field of complex systems. These studies attempt to gain some understanding of what underlies decisions to co-operate or compete [3, 18, 19]). An expert in team interaction rules would be required to understand which structures should be developed to promote co-operation. Ostrom makes revealing observations on the manner in which successful policies are arrived at by communities of independent individuals sharing a common resource. Co-operative solutions come from within the community affected rather than being externally imposed and tend to be most durable if they can be freely changed by the participants [6]. If experts in these solutions exist, one would expect them to be local to the scenario in which they find themselves.

In the context of online ad-hoc teams, such as programmers creating software or medical professionals and paramedics preparing a community-based care plan, the participants are likely to find themselves faced with the same task. This may occur for different reasons and with different people, but the accumulation of some degree of experience is likely to occur. A software expert would be one which accumulates a record of interaction rules which have proved successful in that specific scenario. A software expert has the added advantage of being able to monitor multiple parallel teams. It is therefore likely that the software will accumulate more experience (and be able to manage the information more reliably) than a human expert could, should one exist.

The expert agent proposed in this architecture would return recommendations of prior successful rules to subsequent teams. This would perhaps reduce the need for the team to reinvent them. In keeping with Ostrom's above-mentioned observations, the rules would not be imposed and would be available for the participants to alter. Emergence of rules best suited to the specific team would still be possible.

The method of selecting and prioritising the recommended rules raises some interesting issues. Rules that fire the most often are not necessarily good rules – they may just be poorly constructed. Rules that are not broken may not fire, but may be good. Revised rules might be more reliable. The rules themselves will become the data and an appropriate Machine Learning technique may be needed to select appropriate rules.

3. CONCLUSIONS

The understanding that leaderless ad-hoc online teams can be seen as complex systems informs the design of any software intended to support their collaboration. Team processes and outcomes must be considered unpredictable. A Moderator based on expert system techniques has been proposed to facilitate the democratic creation of interaction rules for the team as a means of stabilising a loosely-coupled entity. Expert system methodology provides the structure needed to encode the emergent policies for controlling competitive forces and encouraging co-operation within the team. The MCRDR system allows an interface through which humans can experiment with the creation of these policy rules. Team members using metacognitive processes will learn at the same time as refining the rules. The role of human experts in these ad-hoc scenarios can be fulfilled by a software expert capable of learning successful policies. A number of research questions follow.

From a computer supported collaborative work perspective one must determine whether the software optimizes task performance. Task experts can evaluate the efficiency (value / time) performance of teams with and without the software support.

A more fundamental question for software engineers is whether it is possible for ad-hoc online teams to operate asynchronously without some overt statement of interaction rules. It may be that there are rules found in the communications of those teams not using a Moderator. Running a set of teams with the choice of using the Moderator and a set that must use the Moderator would reveal more about the usefulness of the software from a performance perspective. It may be simply easier to deal with the interaction rules through the communications channels.

The Moderator can be evaluated as a learning tool for immature team members. "Before and after" studies evaluating individual understanding of team processes would use a traditional pre-test / post-test methodology in combination with a qualitative survey. Increases in team maturity are likely to be reflected in reduced need for communications [12]. There may be a decrease in frequency of some types of communication and an increase in frequency of other types. This could be studied if an effective communication classification process can be implemented. The question of whether team maturity correlates with individual learning could also be explored.

Is it legitimate to consider ad-hoc online teams as complex systems? Examination of patterns of change of rule sets for each team will inform understanding of whether ad-hoc online teams behave as complex systems. Data mining a collection of rule sets may indicate underlying patterns. The extent to which the rules are unsuccessfully categorised may indicate the extent to which teams are complex and unpredictable. Further information can be gained by examining the diversity of expert rule sets when the software is used in different contexts. It may be true that there is an identifiable minimum rule set for all loosely-coupled online team scenarios.

4. ACKNOWLEDGEMENTS

Peter Reimann deserves considerable thanks for sparking interest in to the area of collaborative learning and support.

REFERENCES

[1]. Marion, R. and J. Bacon, "Organizational Extinction and Complex Systems." Emergence, Vol. (4): pp. 71-96, 2000.

[2]. Ng, D., "The Social Structure of Organisational Change and Performance." Emergence, Vol. (1): pp. 99-119, 2003.

[3]. Schweitzer, F., L. Behera, and H. Mühlenbein, "Evolution of Cooperation in a Spatial Prisoner's Dilemma." Advances in Complex Systems, Vol. (2-3): pp. 269-299, 2002.

[4]. Kildare, R. "Ad-hoc online teams as complex systems: agents that cater for team interaction rules." in The 7th Asia-Pacific Conference on Complex Systems. Cairns, Australia 2004.

[5]. Baker, D.P. and E. Salas, "Principles for Measuring Teamwork: A Summary Look Toward the Future.," in Team Performance Assessment and Measurement: Theory, Methods and Applications, M.T. Brannick, E. Salas, and C. Prince, Editors. Lawrence Ehrlbaum Associates: New Jersey. pp. 331-355, 1997.

[6]. Ostrom, E., *Governing the Commons. The evolution of institutions for collective action*. New York: Cambridge University Press, 1990.

[7]. Brannick, M.T., E. Salas, and C. Prince, eds. *Team Performance Assessment and Measurement: Methods and Applications*. 1997, Lawrence Ehrlbaum Associates: New Jersey.

[8]. Tompson, G.H., "The Role of Top Management Team Conflict.A Redistribution of Power." Electronic Journal of Radical Organisation Theory, Vol. (2) 1997.

[9]. PHPROJEKT. "PHProjekt- and open source groupware suite," http://www.phprojekt.com/, 2004

[10]. Pinkwart, N. and K. Herrmann. "Checking Conditions for Graph Based Collaborative Modeling Systems." in AIED2003. Sydney Australia: SIT 2003.

[11]. Reimann, P. "How to Support Groups in Learning:More Than Problem Solving.." in AIED2003. Sydney Australia: University of Sydney 2003.

[12]. Kraiger, K. and L.H. Wenzel, "Conceptual Development and Empirical Evaluation of Measures of Shared Mental Models as Indicators of Team Effectiveness.," in Team Performance Assessment and Measurement: Theory, Methods and Applications, M.T. Brannick, E. Salas, and C. Prince, Editors. Lawrence Ehrlbaum Associates: New Jersey. pp. 63-84, 1997.

[13]. Giarratano, J. and G. Riley, *Expert Systems Principles and Programming*. 3rd ed. Boston: PWS Publishing Company, 1998.

[14]. Friedman-Hill, E. "JESS," http://herzberg.ca.sandia.gov/jess/, 2004

[15]. Kang, B., P. Compton, and P. Preston. "Multiple Classification Ripple Down Rules: Evaluation and Possibilities." in 9th AAAI-Sponsored Banff Knowledge Acquisition for Knowledge-Based Systems Workshop. Banff, Canada 1995.

[16]. Jermann, P., et al. "From Mirroring to Guiding: A Review of State of the Art Technology for Supporting Collaborative Learning." in First European Conference on Computer Supported Collaborative Learning. Maastricht The Netherlands 2001.

[17]. GroupSystems. "Cognito Teamwork ActiveMethod," http://www.groupsystems.com/products/teamwork.htm, 2004

[18]. Lopez-Ruiz, R. and D. Fournier-Prunaret. "Three Logistic Models for the Two-Species Interactions: Symbiosis, Predator-Prey and Competition," http://xxx.lanl.gov/PS_cache/nlin/pdf/0406/0406020.pdf, 2004

[19]. Waldrop, M.M., *Complexity. The Emerging Science at the Edge of Order and Chaos.*, New York: Simon and Schuster, 1992.