

An Online Tool for Learning Collaboration and Learning While Collaborating.

Robert Kildare

School of Computing
University of Tasmania
Launceston, Tasmania, Australia

rkildare@utas.edu.au

R. N. 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

Individuals are increasingly required to join as teams to complete online tasks. This impacts education in three ways. Firstly teachers increasingly set collaborative online tasks for students when teaching curriculum. Secondly individuals need to learn online collaboration skills. Finally, collaborative knowledge creation and innovation can occur when team members take risks. Educationally sound software must promote a psychologically secure environment. Software currently available for online learning and collaboration is predominantly task-based and does not support personal and social aspects of team interaction. Computer Supported Collaborative Learning (CSCL) and Computer Supported Collaborative Work (CSCW) communities are actively researching team interaction. Time Interaction and Performance (TIP) theory, while providing some framework for understanding interaction, does not recognise the unpredictability of team processes. This paper describes software which is hypothesised to support the education and performance of online team members. The software captures democratically created symbolic interaction rules and monitors indicators of the team's interaction. If the conditions are met for the rules to fire, the software enacts the rule consequent. The software, based on a model-free expert system, will accommodate emergent team interaction patterns and provide evolutionary, analytical feedback to both team members and researchers. Constructivist principles of activity and metacognition underlie the validity of this as an educational tool.

Keywords: trust, conflict, collaborative learning, knowledge creation, virtual teams, complex systems, expert systems, data mining.

1 Introduction

This paper introduces a prototype module that can be added to online, database-enabled collaboration software. It permits the democratic creation and editing of interaction rules by team members.

The software acts as a team conscience or public service by enacting the consequences specified by the team, should the conditions of a rule be met. The human role most similar is that of a moderator for an online group.

Online teams are becoming increasingly important in our society as participants take advantage of the ability to overcome distance and scheduling barriers. They are now commonplace in our educational institutions and workplaces. Prime examples include inter-professional and intra-professional collaboration in the expanding health sector and in different types of engineering.

Teams can be distinguished from groups by their structure and purpose. Groups may shift in focus and lack internal structure. Both traditional team studies and more recent cross-system analysis (comparing definitions from the fields of insect behaviour, robotics theory and human team studies) concur.

A team is two or more individuals with specific (although not fixed) role assignments who must perform specific tasks and must interact or coordinate to achieve a common goal or outcome. The efforts of the team must amount to more than the sum of its parts (Baker & Salas 1997; Anderson & Franks 2004).

Commonly these teams are composed of independent participants who combine for the purposes of completing a specific task and may not work together again in the same unit. If there is any advantage in terms of efficiency, in overcoming lack of ability, improving fault tolerance, or if it is cheaper to use a team of single taskers than pay for a multitasking individual, then the efforts of the team amount to more than the sum of its parts (Anderson & Franks 2004). It will be difficult to find a team that does not provide at least some of these benefits.

2 Collaborative learning

Both CSCL and CSCW communities recognise the importance of learning about the processes of collaboration, not just learning about the task or problem itself.

2.1 Knowledge and Task

Educators use teams in their classes when fostering learning through collaboration. Teams are primarily concerned with the sharing, acquisition and creation of knowledge and teachers must increasingly prepare students for this task in a loosely coupled cyber-world of collaboration.

Dimitracopoulou (2005) distils the aim of computer-based collaborative educational systems in her literature review:

In all these systems, collaborative learning is viewed as a pedagogical method that can stimulate students to discuss information and problems from different perspectives, to elaborate and refine these in order to re-construct and co-construct (new) knowledge or to solve problems. In such situations, externalisation, articulation, argumentation and negotiation of multiple perspectives are considered the main mechanisms that can promote collaborative learning. (Dimitracopoulou 2005, p. 122)

Problem Based Learning (PBL) uses a common problem as the stimulus for collaboration. Collaboration brings with it advantages of shared cognitive load, and the sharing of expertise (Chernobilsky et al. 2005). Knowledge sharing, its construction and reconstruction, occurs with the social purpose of solving the shared problem. Iteratively internalising the knowledge and making it available socially are the general processes for learning through collaboration (Ertl et al. 2005; Eleuterio et al. 2000). PBL uses scenarios found outside educational institutions as a means of preparing students for tasks they will experience in the field. Learning about knowledge associated with the task can be considered the primary focus or first order (Ulcsak 2000) of collaborative learning.

2.2 Team Processes

Teamwork has also been seen to promote the second order learning of individual responsibility, social awareness and communication skills (Ulcsak 2000). It is in this area that computer support has been lacking - an area vital to the success or failure of the team. Major risks, including activities such as freeloading (Ostrom 1990) and domination (Brannick et al. 1997; Tompson 1997), may threaten team existence. Another source of destructive conflict is interpersonal conflict and its corollary, “misattribution” – misinterpreting a debate about ideas as a personal affront (Crampton 2002). Having team members learn about team processes rather than the primary knowledge issues can be considered an end in itself. It is important to learn collaboration as well as to learn while collaborating.

2.3 CSCL and CSCW perspectives

Given that education does not stop when one leaves school, it is unsurprising that the Computer Supported Collaborative Learning (CSCL) and Work (CSCW) communities share a common desire to facilitate optimum team performance and to maximize knowledge creation, knowledge sharing and its innovative use, as well as to minimize the forces of team instability. These communities draw from relevant studies in social psychology and organizational systems in an attempt to describe and understand teams.

Baker and Salas divide team processes into ‘task-based’ and ‘team-based’ (Baker & Salas 1997). These fit into McGrath’s Time Interaction and Performance (TIP)

theory (McGrath 1991). Task-based processes approximate the processes that occur when a team is carrying out McGrath’s ‘production function’. The processes of task knowledge sharing and development fall into the task-based classification. Team-based processes would be used when fulfilling the ‘member support’ and team ‘well-being’ functions (see table 1). The processes associated with the learning of social awareness, personal responsibility and the associated skills of communication and decision-making belong in the team-based realm.

MODES	FUNCTIONS		
	Production	Well-Being	Member Support
Mode I. Inception	Production Demand/ Opportunity	Interaction Demand/ Opportunity	Inclusion Demand/ Opportunity
Mode II. Problem Solving	Technical Problem Solving	Role Network Definition	Position/ Status Attainments
Mode III. Conflict Resolution	Policy Conflict Resolution	Power/ Payoff Distribution	Contribution/ Payoff relationships
Mode IV. Execution	Performance	Interaction	Participation

Table 1. McGrath’s analytical framework for team performance. Teams perform three functions when in each of four modes of operation. (McGrath 1991)

3 Current Software

Current collaborative software such as project management tools, Gantt charts, content management systems, knowledge mapping tools, process templates (scripts) and communications facilities (forum, video conference, email, etc) are predominantly task-based in their focus. (Plone 2005; PHPROJEKT 2004; Carell et al. 2005; Pinkwart & Herrmann 2003). Krenjins and Kirschner (2002) commented on the lack of attention paid to social interactions when they introduced widgets for providing social affordances. This lack of focus on team-based interaction processes has begun to change. For example, satisfaction and motivation levels were mirrored to team members (Reimann 2003), and knowledge construction networks were analysed to identify passivity in participants. (Aviv et al. 2003).

3.1 Unpredictability

Prescriptive software of either the team or task types can be prone to mismatching the processes and patterns of specific teams. Team members required to use sentence starters when communicating (an attempt to classify types of communications) did not use them as expected (Dimitracopoulou 2005). Ertl, Kopp and Mandl (2005) reference four studies yielding “mixed results” from testing prescriptive knowledge management tools. They attempt to explore prior knowledge as the source of these

“mixed results”. They concluded “that the effects of an individual’s prior knowledge are quite varied in the quality of collaborative knowledge construction and individual learning outcome” (Ertl et al. 2005).

Unpredictability continues to haunt quantitative researchers trying to find meaningful variables upon which to create computerised collaboration scaffolding. Zumbach, et al (2005) note the difficulties confronting software designers. Their work traces collections of team member actions as the basis for defining interactions. Human observers interpret whether these are effective collaboration processes. While their work reveals the value of mirroring to the team their history of communication patterns, problem-solving processes, “meaningful” collaborations and emotional states, there is a major difficulty in automatically recognising meaningful patterns and the meaningful absence of patterns (patterns that should exist in an effective team’s interaction, but are absent). “The dilemma here is that software, so far, has not surpassed humans in their ability to identify meaningful interaction patterns...” (Zumbach et al. 2005). Unpredictability and the identification of key variables confront all disciplines that deal with complex systems.

4 Teams as Complex Systems

Teams have been recognised as complex systems. They are part of and are composed of other complex systems. They demonstrate non-linear relations between inputs and outputs and display phases of emergent evolutionary behaviour. McGrath’s modes II and III (see table 1, above) can be regarded as the phase where patterns emerge. The functions of membership support and maintaining team well-being are interdependent (Arrow et al. 2000). The teams are composed of members (learners) whose tangle of myelin coated neurons, and experiences constitute complex systems. The teams are products of complex contexts such as a health delivery system (Goodchild et al. 2005). Feedback mechanisms (“payoff”) can turn emergence into chaos (Kildare 2004).

Problem solving is the primary activity of teams – members must work out how to complete the task that unites them. Kapur, et al (2005) analyse the problem solving process in genetic evolutionary terms associated with dynamic, open systems. Number, function, fitness and persistence are dealt with in an attempt to describe the evolution of problem solving process. In keeping with the observation by Ostrom (1990) that small units should control their own processes, problem solving was most effective when control was taken by the students using Freinet methodology in Mexican schools (Serrano et al. 2003).

Learning in general, and problem solving in particular, is a continuous, dynamic process distributed in space and time over multiple actors, actions and artefacts, influencing and being influenced by the environment in a complex, adaptive, and iterative manner. (Kapur et al. 2005, p. 252).

It follows that software designs intended to assist the collaboration process must take into consideration the

unpredictable and evolutionary nature of teams. The need for democratic regulation also underpins the software design presented in this paper.

5 Why a democratic regulatory system?

Balancing the levels of trust and risk can be achieved by formalising emergent team norms as regulations. If these regulations are imposed from a corporate policy or by software engineers, rather than democratically by the team members themselves, then the level of complexity of the team is increased. While externally imposed regulation may act to control instability, the regulators become added sources of instability.

5.1 Trust and Risk

As noted above, the area of team well being and the processes that emerge for member support are not frequently addressed. Studies of what makes a successful or an unsuccessful work team do, however, reveal many features whose processes could be categorised as belonging to the team rather than the task domain.

Trust and risk arise as key areas of research in the CSCW community:

trust in a dyadic work relationship is defined as an individual’s willingness to be vulnerable to the actions of the other involved party based on a particular action important to the trustor, irrespective of the trustor’s ability to monitor or control the trustee. (Hung et al. 2004, p. 2).

Fernández (2004) provides an extensive summary of the nature of trust as part of his study of “metateams” – teams that extend across multiple corporations or institutions. In his business scenario virtual teams can replace trust with contracts – they can write the relationship down. They cannot, however, write down everything. Risk is defined as “trusting uncontrollable others” (Fernández 2004). In other words, risk is considered an inherent part of trust.

Trust is bound up with the equitable distribution of rewards that flow from being part of a team, in exchange for the contribution from each member. Members must either trust or regulate that each other will neither dominate nor freeload (Ostrom 1990; Hertel 2004) and thus skew the equitable nature of the distribution of rewards. These rewards can be monetary (Simmel et al. 1978) or more abstract (eg: knowledge, status, influence, etc), as theorised by Max Weber (Abukuma, 2004). Along with writing contracts, some form of regulatory control is also needed.

The need to “write the relationship down” is as unquestionable as the need to implement appropriate controls... We maintain that trust cannot be totally replaced. This is so because its replacement, total control, is a delusion in complex sociotechnical systems; and we know that contracts and specifications are inherently imperfect. ... Therefore writing a relationship down will always involve elements of risk and misinterpretation (Fernández 2004, p. 49).

Fernandez argues from the literature that trust “facilitates open communication, cooperative efforts, reduction of uncertainty, resolution of conflicts, common understanding, and control of transaction costs.” Trust is both cause and effect of co-operation. Trust and mistrust are not absolutes but either end of a continuum where a balance between regulation and trust is struck. His study reveals that both too much and too little trust caused team failure and that feedback mechanisms amplified the mistrust and trust. He argues that the correct balance of trust to regulation can only be understood in hindsight; that it emerges from team interaction (Fernández 2004). This observation suggests the behaviour of a complex system and is important when designing computer support for regulatory processes. Self-controlled governance replaces formal authority-based regulation and the consequent trusting environment enables “open and substantive information exchange” (Neece 2004).

Trust is established early in the formation of a team, especially through face-to-face meetings (Hung et al. 2004) “when there is a common belief that others will make good-faith efforts to behave in accordance with commitments (explicit or implicit) and act honestly in negotiation of those commitments” (Hertel 2004).

The trust found in virtual teams is different from that expressed by teams within a corporation and from those teams which are co-located. The trust relies on loyalty to design, harmonious patterning and creativity; that to which all team members aspire. Other forms of loyalty are either to the regulations and goals of the corporation, or in the case of co-located teams, loyalty to each other (Murphy 2004).

Trust is undermined by an undisciplined, laissez faire attitude to commitments (Hertel 2004). Both Neece and Fernandez recommend education for all team members in the nature and value of trust. Methods include early familiarity-building project meetings, responses to deadlines, acceptance of deliverables, detected contract breaches, or joint conflict management and resolution efforts (Neece 2004) (Fernández 2004). These methods do not necessarily have to occur face-to-face, but they do require an understanding of the team expectations and awareness of breaches of these expectations.

5.2 Educational value of rule setting and mirroring

The creation of interaction rules allows expectations to be expressed and may substitute for some of the face-to-face familiarity building (seen as so important to reinforcing the initial trust of team members). The mirroring of consequences of the rules can give team members awareness of breaches of the team’s norms - if the rule conditions are valid. Checking the validity and editing poorly constructed rules will prompt deeper understanding of expectations. The metacognitive stimulation provided by mirroring is a well-accepted constructivist learning technique used by educational software (Jermann et al. 2001). It is hypothesized that the creation and editing of interaction rules will teach members about the nature of self-regulation and

simultaneously engender trust. The more trusting or “psychologically safe” the environment, the more likely it will be that risks are taken. This will promote a greater degree of knowledge exchange, which is an indicator of team efficacy, and a greater likelihood of knowledge creation and of innovation when generating solutions to problems (Crampton 2002). A tool for learning about collaboration should also improve the learning associated with collaboration.

6 Rules

Production rules allow the user to attach symbolic meanings to sets of measurable conditions. The conditions express the state of the collaborative tools in the software, for example the number of times a person logs into the website. Allowing team members to freely create, edit and delete rules permits the dynamic expression of meaning and importance to evolve.

6.1 A Model-free Expert System

From the observations of successful and unsuccessful teams (Fernández 2004; Crampton 2002; Armstrong & Bill 2002; Hertel 2004; Kimble et al. 2001; Jarvenpaa & Leidner 1999; Hollingshead et al. 2002), one can deduce the likely subject matter of interaction rules. Rules may emerge about punctuality, attendance levels, level of contribution to collective resources, biases in types, frequencies and promptness of communications, the altering of others’ work, secretive use of information, insufficient off-task (social) chat and conflict of a personal nature.

The following rules are a selection of those created while developing the prototype:

IF today’s date is greater than the deadline of task X
THEN task X is “late”.

IF task X is “late” AND X has dependent tasks THEN
notify the team “task X is critically late.”

IF “feedback” communications for Member X are less
than half the team average THEN notify X “Your
feedback is less than half the team average.”

IF login frequency for Member X is less than half the
team average THEN notify the team “Member X
attendance poor.”

It is also worth noting that interaction rules do not necessarily have to be punitive, but can reward positive achievements as well. Any rule monitoring must be constrained by what the system can monitor. Symbolic rules (as used in expert systems) permit the user to attach meanings to these measurables as they see fit by defining terms. “Late” is defined above and then reused as a condition in a subsequent rule. A low level of social communications could be rewarded with the label “efficient” and another team might label that same level of social communication “anti-social”.

A model-free expert system will allow the team to define its own regulatory environment. This is in keeping with the observations of what is best for the success of a small independent unit (Ostrom 1990) and what is best for

maximising creativity (Neece 2004). The ability to delete and edit the interaction rules will allow for the expression of meaningful patterns of interaction to emerge dynamically as required by a complex system.

Perversely, the monitoring of team regulations also threatens trust, once it has developed. Monitoring rules can provide evidence of both renegeing on commitments and/or incongruence of expectations. The effect is the same; it is seen as a failure on the part of a team member to fulfil obligations (Piccoli & Ives 2003). While deliberate renegeing cannot be prevented, the opportunity to expose incongruity between team member's expectations and clarify issues can be provided using the expert systems approach.

6.2 MCRDR and Conflict Resolution

It is possible, if an expert system has no underlying knowledge model, to accept multiple consequents (or classifications) for the same set of conditions. Contradictory rules may fire. Researchers working on the development of the Multiple Classification Ripple Down Rules (MCRDR) approach to knowledge acquisition (Kang et al. 1995) have developed an approach to resolving what are considered undesirable inconsistencies in the rule set. They ask the expert why a given classification is the wrong one for a set of conditions. The expert is given the opportunity to edit the rule by adding further conditions that might lead to a correct classification, or to create a new rule altogether (Kang et al. 1995). In the team scenario, where the members construct the rules, differing expectations will be made overt when rules with the same conditions, but different interpretations, fire. The team is made aware of the situation when it is mirrored to them.

Studies of conflict divide it into two types; personal, which is considered destructive to teamwork and content - based (also termed concept-based or issue-based). The latter form is considered constructive if handled in an open and trusting manner, leading to greater understanding and exchange of views and often resulting in creative resolution of differences (Crampton 2002). The hope is that making overt the differing expectations between team members will create a situation for positive conflict resolution.

7 Software Design

At the most general level, the software envisaged will permit the team members to freely decide on the desirability of certain meaningful interactions. They define the interaction by selecting a set of attribute-value pairs. At the very least the consequence of such a set of pairs existing will be that this precondition is labelled with a meaning given by the user. Other consequences might include sending a notification to the team, sending a notification to an individual member or altering permissions on access control lists.

The moderator software will need to measure the state of the team. Most online task-based collaborative software is either naturally database-centred or can be restructured to feed low level data into one. Specific queries are designed

to draw out information from the database. These queries constitute the team "state". The queries in this prototype are low level facts (eg: user names, web page names, dates), frequencies (such as the total logins per user), averages (eg: the average number of logins for a hypothetical team member) and rates (eg: the average number of logins per week for the team), attempting to measure whatever can be measured. The queries however, can be redefined to draw out higher level abstractions of team state.

The measurables that define the state of the team are then used as the basic tools with which users create the conditions of their rules. Consequents of already existing rules can also be used as conditions. The state of the team is regularly monitored and checked against the rules and any consequents that require further action (such as feedback to the team) are completed.

The moderator software, a Java application which will eventually run as a servlet, is currently able to monitor a MySQL database, translate the results of SQL queries into CLIPS facts and execute prewritten rules applied to these facts. The next stage will involve designing and implementing the online rule-making interface. The entire process of rule construction activity raises a larger educational concern; that of cognitive overload. (Reimann et al. 2005; Sintchenko & Coiera 2003). Just how appealing for the user is the prospect of sifting through sets of team states, creating rules that might prove to be unnecessary. The user will assume at first that the other team members have similar views until this is shown to be incorrect (Menzies et al. 1999). How will team members be taught to carry out this extremely abstract exercise? Further, what is the point of reinventing rules that have been shown by other teams or by research to work? The very real risk is that rule setting, despite its importance, will be seen as just too much to be bothered with, given that there is enough to manage just completing the main task-based activities of the team.

It would seem sensible to seed the process of rule construction with a few examples, and ideally, provide recommendations for teams; rules that the system recognises as having worked for their type of team in the past. These could be used as a start and then modified by the team as their work progresses.

If the team is one created in a learning institution then a human facilitator will be providing the external context and can provide a set of recommendations. So too could a corporate overseer. The problem with prescription, of course, is that one cannot necessarily predict which rules will suit a given team. Experts with heuristic knowledge may be 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 is proposed that would use the rule sets and states of many team moderators as the basis for its learning. Teams may be sorted into types according to their states, or changes in their states. Underlying concepts have been inferred in model-free expert systems in both hierarchic (Richards 1998) and mesh structures

(Suryanto 2004). The software expert will permit recommendations to evolve, thus adapting to changes experienced within the team. At this stage, given the distributed, independent nature of multiple teams it is likely that the proposed software will be implemented as a multi-agent system.

8 Further research

Once the software has been implemented and a number of teams have been involved in trials, it will be possible to evaluate the impact that the software has on overall team stability, performance and the development of individual members. Measures such as outcome evaluation (eg marks for the team assignment), degree of knowledge sharing, efficiencies reflected in communications patterns and the learning achieved by individual team members are all possible candidates for evaluation.

The learning of the specific skills involved in collaboration can be tested by comparison with teams which do not use the moderator software. This comparison would particularly focus on levels of trust and how they change over time. Measures of the degree of cognitive load may help to determine whether interaction regulation is worth the extra load.

Finally, it should be possible to gain further understanding of appropriate rules and the types of teams to which such rules apply. It may also be possible to determine what aspects are and aren't predictable. It may be possible to represent key measurables in visual form, avoiding the need to label with symbols and thus reducing cognitive load.

9 References

- Abukuma, M., *Weberian Sociology of Religion. Max Weber's Texts: Sociology of Community*[Online], Available: http://www.ne.jp/asahi/moriyuki/abukuma/weber/society/soci_comm/index.htm[accessed: August 2004]
- Anderson, C. & Franks, N. R. (2004), Teamwork in Animals, Robots and Humans, *Advances in the Study of Behaviour*, **33**, pp 1-48.
- Armstrong, D. J. & Bill, P. E. (2002), Virtual Proximity, Real Teams, in *Distributed Work*. 187-191, Hinds, P. & Kiesler, S. (eds), Massachusetts Institute of Technology, Cambridge, Massachusetts.
- Arrow, H., McGrath, J. E. & Behrdal, J. (2000), *Small Groups as Complex Systems: Formation, Coordination, Development and Adaptation*, Sage Publications, Thousand Oaks, California.
- Aviv, R., Erlich, Z., Ravid, G. & Geva, A. (2003), Network Analysis of Knowledge Construction in Asynchronous Learning Networks, *Journal of Asynchronous Learning Networks*, **7**, (3), 1-23.
- Baker, D. P. & Salas, E. (1997), Principles for Measuring Teamwork: A Summary Look Toward the Future., in *Team Performance Assessment and Measurement: Theory, Methods and Applications*. 331-355, Brannick, M. T., Salas, E. & Prince, C. (eds), Lawrence Erlbaum Associates, New Jersey.
- Brannick, M. T., Salas, E. & Prince, C. (eds) 1997, *Team Performance Assessment and Measurement: Methods and Applications*, Lawrence Erlbaum Associates, New Jersey.
- Carell, A., Herrmann, T., Kienle, A. & Menold, N. (2005), Improving the Coordination of Collaborative Learning with Process Models, in *Computer Supported Collaborative Learning 2005*, eds Koschmann, T., D.Suthers & T.W.Chan, Lawrence Erlbaum Associates, Taipei, Taiwan 18-27.
- Chernobilsky, E., Nagarajan, A. & E.Hmelo-Silver, C. (2005), Problem-Based Learning Online: Multiple Perspectives on Collaborative Knowledge Construction, in *Computer Supported Collaborative Learning 2005*, eds Koschmann, T., D.Suthers & T.W.Chan, Lawrence Erlbaum Associates, Taipei, Taiwan 53-62.
- Crampton, C. D. (2002), Attribution in Distributed Work Groups, in *Distributed Work*. 191-212, Hinds, P. & Kiesler, S. (eds), Massachusetts Institute of Technology, Cambridge, Massachusetts.
- Dimitracopoulou, A. (2005), Designing Collaborative Learning Systems: Current Trends & Future Research Agenda, in *Computer Supported Collaborative Learning 2005*, eds Koschmann, T., D.Suthers & T.W.Chan, Lawrence Erlbaum Associates, Taipei, Taiwan 115-124.
- Eleuterio, M. A., Bortolozzi, F. & Kaestner, C. A. (2000), The Roles of Ontologies in Collaborative Virtual Learning Environments, in *European Conference on Artificial Intelligence 2000*, ed. Horn, W., I.O.S. Press, Berlin, Germany 31-35.
- Ertl, B., Kopp, B. & Mandl, H. (2005), Effects of an Individual's Prior Knowledge on Collaborative Knowledge Construction and Individual Learning Outcomes in Videoconferencing, in *Computer Supported Collaborative Learning 2005*, eds Koschmann, T., D.Suthers & T.W.Chan, Lawrence Erlbaum Associates, Taipei, Taiwan 145-154.
- Fernández, W. D. (2004), Trust and the Trust Placement Process in Metateam Projects, in *Virtual Teams: Projects, Protocols and Processes*. 40-69, Pauleen, D. J. (ed). Idea Group Publishing, Hershey, Pennsylvania.
- Goodchild, A., Bird, L. & Hoylen, S. (2005), Using complex systems techniques to aid health software design, in *Health Informatics Conference 2005*, eds Chu, S., Grain, H. & Wise, M., Health Informatics Society of Australia, Wellington, New Zealand 019, 1-13.
- Hertel, T. D. (2004), Effective Virtual Teams, in., *Virtual Teams: Projects, Protocols and Processes*. 209-230, Pauleen, D. J. (ed). Idea Group Publishing, Hershey, Pennsylvania.

- Hollingshead, A. B., Fulk, J. & Monge, P. (2002), Fostering Intranet Knowledge Sharing: An Integration of Transactive Memory and Public Goods Approaches, in *Distributed Work*. 335-355, Hinds, P. & Kiesler, S. (eds), Massachusetts Institute of Technology, Cambridge, Massachusetts.
- Hung, Y.-T. C., Dennis, A. R. & Robert, L. (2004), Trust in Virtual Teams: Towards an Integrative Model of Trust Formation, in *37th Hawaii International Conference on System Sciences 2004*, IEEE, Hawaii 1-11.
- Jarvenpaa, S. L. & Leidner, D. E. (1999), Communication and trust in global virtual teams, *Organization Science*, **10**, (6), 791-815.
- Jermann, P., Soller, A., Muehlenbrock, M. & Monés, A. M. (2001), 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 324-331.
- Kang, B., Compton, P. & Preston, P. (1995), Multiple Classification Ripple Down Rules: Evaluation and Possibilities, in *9th AAAI-Sponsored Banff Knowledge Acquisition for Knowledge-Based Systems Workshop*, eds Gaines, B. & Musen, M., Banff, Canada 17.1 - 17.20.
- Kapur, M., Voiklis, J. & Kinzer, C. K. (2005), Problem Solving as a Complex, Evolutionary Activity: A Methodological Framework for Analyzing Problem-Solving Processes in a Computer-Supported Collaborative Environment, in *Computer Supported Collaborative Learning 2005*, eds Koschmann, T., D.Suthers & T.W.Chan, Lawrence Erlbaum Associates, Taipei, Taiwan 252-261.
- Kildare, R. (2004), 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.
- Kimble, C., Li, F. & Blanchflower, A. (2001), Overcoming the Barriers to Virtual Team Working Through Communities of Practice, *Management Science*, **Research Paper**.
- Krenjins, K. & Kirschner, P. A. (2002), Group awareness widgets for enhancing social interaction in computer-supported collaborative learning environments: design and implementation, *Frontiers in Education*, **1**, 14-20.
- McGrath, J. E. (1991), Time Interaction and Performance (TIP). A Theory of Groups., *Small Group Research*, **22**, (2), pp147-174.
- Menzies, T., Cukic, B. & Coiera, E. (1999), Smaller, Faster Agent Dialogues via Conversational Probing, in *Sixteenth National Conference on Artificial Intelligence (AIII-99)*, Orlando, Florida.
- Murphy, P. (2004), Trust Rationality and the Virtual Team, in *Virtual Teams: Projects, Protocols and Processes*. 317-343, David J, P. (ed). Idea Group Publishing, Hershey, Pennsylvania.
- Neece, O. E. (2004), Factors Contributing to Knowledge Sharing and Communication in a Large Virtual Work Group, in *Virtual Teams: Projects, Protocols and Processes*. 280-316, Pauleen, D. J. (ed). Idea Group Publishing, Hershey, Pennsylvania.
- Ostrom, E. (1990), *Governing the Commons. The evolution of institutions for collective action*, Cambridge University Press, New York.
- PHPROJEKT, *PHPProjekt- and open source groupware suite* [Online], Available: <http://www.phprojekt.com/>.
- Piccoli, G. & Ives, B. (2003), Trust and the Unintended Effects of Behavior Control in Virtual Teams, *MIS Quarterly*, **27**, (3), 365-395.
- Pinkwart, N. & Herrmann, K. (2003), Checking Conditions for Graph Based Collaborative Modeling Systems, in *AIED2003*, eds Aleven, V., Hoppe, U., Kay, J., Mizogucci, R., Pain, H., Verdejo, F. & Yacef, K., SIT, Sydney Australia 33-35.
- Plone, *Plone Content Management System* [Online], Plone Foundation, Available: <http://plone.org/> [August, 2005].
- Reimann, P. (2003), How to Support Groups in Learning: More Than Problem Solving., in *AIED2003*, eds Aleven, V., Hoppe, U., Kay, J., Mizogucci, R., Pain, H., Verdejo, F. & Yacef, K., University of Sydney, Sydney Australia 3-16.
- Reimann, P., Kay, J., Yacef, K. & Goodyear, P. (2005), Adaptive Visualisation Support for Self-managed Learning Groups, in *Artificial Intelligence in Education 2005*, I.O.S.Press, Amsterdam, Netherlands.
- Richards, D. C. (1998), The Reuse of Knowledge in Ripple Down Rule Knowledge Based Systems, PhD., University of NSW.
- Serrano, M. T. E., Cantu, A. G. & Vila, I. M. (2003), Problem-solving: Evaluative study of three pedagogical approaches in Mexican Schools, *Electronic Journal of Research in Educational Psychology*, **2**, (1), 79-96.
- Simmel, G., Bottomore, T. & Frisby, D. (1978), *The Philosophy of Money*, Routledge & Keegan Paul Ltd, Boston, Massachusetts.
- 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, PhD, University of N.S.W.
- Tompson, G. H. (1997), The Role of Top Management Team Conflict. A Redistribution of Power., *Electronic Journal of Radical Organisation Theory*, **2**.

- Ulicsak, M. H. (2000), Analysis and Modelling of Collaborative Learning Interactions: Research Summary, in *European Conference on Artificial Intelligence 2000*, ed. Horn, W., I.O.S. Press, Berlin, Germany 9-17.
- Zumbach, J., Schönemann, J. & Reimann, P. (2005), Analyzing and Supporting Collaboration in Cooperative Computer-Mediated Communication, in *Computer Supported Collaborative Learning 2005*, eds Koschmann, T., D.Suthers & T.W.Chan, Lawrence Erlbaum Associates, Taipei, Taiwan 758-767.