Human-Computer-Human Interaction

Trust in CSCW

Steve Jones and Steve Marsh

Keywords: computer supported cooperative work, groupware, trust

Introduction

Computer Supported Cooperative Work (CSCW) (Ellis *et al.*, 1991) is ostensibly concerned with supporting the activities of work groups through the use of computer technology. However, to date, CSCW systems (groupware) have emphasised technological issues of support at the expense of social issues such as relationships, roles and social protocols.

We postulate that this situation has arisen because the majority of groupware designers are technologists who have both the experience and tools to develop new and effective hardware and software. Unfortunately they do not have tools or experience to effectively analyse and provide support for social facets of group working. Multidisciplinary development teams may contain group work experts, but common languages and vocabulary for precise communication regarding social and relationship aspects of systems are lacking. Groupware designers and developers also require tools to embed their considerations of social issues in systems and then to analyse those systems and the work of the groups which use them.

We have attempted to ameliorate this situation by developing a formal notation of the trust that is present between individuals in collaborative activities. The notation can be used in the representation and consideration of social relationships in the context of CSCW.

We suggest that trust is a key factor in the efficacy of both intra-group and inter-group activities, and that it can be formalised and then exploited in the design and analysis of CSCW systems. We call our formal description TRUST in order to differentiate it from wider definitions. Potential uses of TRUST in a group work context include the following:

- it can be used as a tool for the discussion of the design of CSCW systems;
- it can be embedded in computer systems to mediate cooperative computer based activities;
- it can be used to record and analyze group activity;
- it provides a tool for the discussion and clarification of trust, and its role in group activities.

The development of the formalism addresses the need for support beyond technical issues for designers involved in the development of multi-user-centered systems.

Describing TRUST

Trust is a common phenomenon that has been extensively discussed in the literature of sociology, social psychology and philosophy. Its importance in societies and interpersonal relationships is often highlighted. Golembiewski and McConkie (1975) have stated "Perhaps there is no single variable which so thoroughly influences interpersonal and group behaviour as does trust...". Luhmann (1979) has argued that without it we would not be able to face the complexities of the world, because it enables us to reason sensibly about the possibilities of everyday life. It has been suggested that society would collapse if trust was not present (Bok, 1978; Lagenspetz, 1992). Proposed benefits of trust include better accomplishments in task performance (Golembiewski & McConkie, 1975), greater and more healthy personal development and the ability to cooperate (Argyle, 1991; Deutsch, 1962).

Little investigation has been carried out into the role of trust in a computational context beyond security issues (such as Reiter, 1996), or the human-machine relationship (such as Muir, 1987 and Arion *et al.*, 1994).

We have drawn on related studies of trust in the development of our notation. From Deutsch (1962) and Zeckhauser (1990) we adopt the notion of utility playing a role in trusting behaviour and that the utility of a work context is directly related to the likelihood of an individual cooperating with others in that context. We also adopt Deutsch's view that individuals assume similar behaviour in others. We take Luhmann's (1979) view that trust is concerned with managing complexity and risk. From Barber (1983) we adopt the view that trust facilitates expectations about future behaviour of others, implying a history of trust and use of that history to reason about future actions. We diverge from Barber's view that trust in an individual can not be generalised across contexts through our use of a finer grain representation of trust. A key influence is Gambetta's (1990) use of probabilistic values to represent levels of trust. We adopt this approach although with an amendment to provide a stronger intuitive link between values and their semantics.

Our building blocks of TRUST and the social aspects of group work are as follows:

People. Individuals and groups are the entities whose activities are mediated by a TRUST framework. We represent individuals by $a, b, c, \ldots, z, a', \ldots, z', a'', \ldots, z''$ and so on. Individuals are members of the set of all individuals (everyone), and may be collected into groups. We represent groups of individuals $G_1, G_2, \ldots, G_n \subset A$.

Contexts. Interactions between individuals have a context. The behaviour of an individual may vary greatly from context to context. Our view of contexts is simply as tasks that individuals may undertake collaboratively. We represent contexts $\alpha, \ldots, \omega, \alpha', \ldots, \omega', \alpha'', \ldots \in C$. α_x is context α from *x's* point of view, β_y is context β from *y's*[†]. If a set of people form a group G_1 with a fixed set of contexts, that set of contexts is notated C_{G_1} .

Basic trust. A group member has a disposition which we call basic trust. It may be derived from previous experiences, and may change as a result of new experiences. T_x represents the basic trust of individual *x* and has a value in the range [-1,+1].

Knowledge. A representation of knowledge of others provides an indication as to whether context specific information is available to interacting group members, in order to form expectations regarding behaviour. $K_x(y)$ represents the fact that *x* has met *y* at some time and that *x* can remember it. $\neg K_x(y)$ represents the opposite.

General trust. Given that two individuals know each other, they will have a general notion of how much they trust each other, regardless of the context. We term this general trust. $T_x(y)$ represents the amount of trust that *x* has in *y* and has a value in the interval [-1,+1]. It is not relative to any specific context.

Contextual trust. When individuals have met each other in a specific context they have contextual trust. To describe the trust between *x* and *y* in context α we use $T_x(y, \alpha)$ which takes a value in the interval [-1,+1].

Importance. Interaction in any given context will have some level of importance attached to it by the individuals involved. It may be important for an individual to get a task done, to get it done correctly, not done and so on. We represent the importance of a context α for *x* with $I_x(\alpha)$. It has a value over the interval [0,+1].

Utility. Individuals expect some return from cooperation. We call the expected return *utility* and represent the utility *x* gains from context α , $U_x(\alpha)$ which has values over the interval [-1,+1].

These building blocks provide a basic framework for abstract discussion about relationships in group work which are mediated by trust. This provides CSCW system designers with a vocabulary for discussion, and a notation for description.

Manipulating TRUST

To this point the formalism allows us to represent and quantify facets of group members and relationships between them. We can also represent and quantify some aspects of the group work contexts that members may find themselves in.

The definition so far is purely representative. In order for TRUST to be used in mediating or analysing group work it must describe how members determine whether or not to cooperate with each other. This is achieved by indicating how group members determine contextual trust and the threshold above which they will cooperate in a given context.

Determining Contextual Trust

In the case $\neg K_x(y)$ the contextual trust is determined by the basic trust of *x* (T_x), modified by the utility and importance of the context from the point of view of *x*.

$$T_x(y, \alpha) = U_x(\alpha) \times I_x(\alpha) \times T_x$$

In the case $K_x(y)$ but $\neg K_x(y, \alpha)$ the importance and utility of the context also impact on the contextual trust. However, in this case, more specific previous experience of the trustee is available in the form of general trust. Therefore contextual trust is determined by the general trust of *x* in *y* ($T_x(y)$), mod-

[†] In the formulae that follow, the subscript is often dropped, because it is evident which agent is involved.

ified by the utility and importance of the context from the point of view of *x*.

$$T_x(y, \alpha) = U_x(\alpha) \times I_x(\alpha) \times T_x(y)$$

In the case $K_x(y, \alpha)$ the trust level for the trustee in the current context is estimated from previous levels of trust in the trustee in the same context. Importance and utility are constant for an individual with respect to a given context, although they will be different for different contexts. Therefore, given that they have partly determined past trust levels with the trustee in the current context they are not required here. The estimate is notated $\overline{T_x(y, \alpha)}$, therefore

$$T_x(y, \alpha) = \overline{T_x(y, \alpha)}$$

We do not specify how this estimation takes place. One approach may be to derive the estimate dependent on the disposition of the truster. An optimistic truster may take the maximum of those values, a pessimistic truster the minimum, or a realist the mean.

Determining the Cooperation Threshold

The threshold above which an agent will cooperate with a trustee in a given context is notated $\mathrm{Cooperation\_Threshold}_x(\alpha)$. So

$$T_x(y, \alpha) > \mathrm{Cooperation\_Threshold}_x(\alpha) \Rightarrow \mathrm{Will\_Cooperate}(x, y)$$

Three factors determine the cooperation threshold. As the risk that the truster perceives in given context increases so should the threshold. The risk is moderated by the perceived competence of the trustee; the higher the competence in the current context, the lower the threshold. The importance of the current context also moderates the threshold. Therefore

$$\mathrm{Cooperation\_Threshold}_x(\alpha) = \frac{\mathrm{Perceived\_Risk}_x(\alpha)}{\mathrm{Perceived\_Competence}_x(y, \alpha) \times I_x(\alpha)}$$

We define both $\mathrm{Perceived\_Risk}$ and $\mathrm{Perceived\_Competence}$ to have values in the range [0,1] but are continuing to develop full definitions of how they are derived.

Updating TRUST

So far we have provided a notation to describe under which circumstances cooperation between individuals may or may not take place. However, the formalism is, as it stands, static. Values will remain constant and behaviour will not adapt to changing circumstances. We therefore introduce dynamic behaviour which provides for the update of values as a result of interactions between individuals.

An interaction between group members *x* and *y* may result in revisions to basic, general and contextual trust values for the two members. Therefore, we must extend the formalism to incorporate such modifications. The level of modification of trust values is an issue. We contend that after an interaction between group members, contextual trust is modified the most as the truster can consider with certainty the most recent behaviour of the trustee in the current context. General trust is modified less than contextual trust because the interaction was in only one of perhaps several contexts. The smallest modification is made to basic trust because the interaction was with one of perhaps several individuals in one of perhaps several contexts.

Members of a group are considered to have a level to which they are prepared to modify trust values. This is determined by a member's own disposition and by the disposition of the group as a whole. Here we take the view that as basic trust increases so will the level of modification, and as basic trust decreases, so will the level of modification. This models a group member's trust that trustee's behaviour was consistent and predictable.

Exploiting TRUST

The formalism described above provides a notation which may support CSCW systems designers in describing

- group members and the relationships between them;
- contexts in which group members may find themselves;
- under what circumstances group members will cooperate with each other

and we further outline how it may be extended to determine

- how group members' outlooks may change as a result of the actions of themselves and others.

As such, it is a tool that allows concise and unambiguous descriptions of the social relationships in group work associated with the use of a CSCW system.

The formalism has been developed with the goal that it is implementable in computer software and that designers may move from its descriptive use to a direct representation in a programming language. We have developed two software implementations to act as testbeds for TRUST. One uses a conventional imperative programming language (C) in a Unix environment. The other uses an event based scripting language (HyperTalk) in an Apple Macintosh environment.

Trust may be used to mediate (or appropriately constrain) group activities. Some CSCW systems such as ACE (Dykstra and Carasik, 1991), Grove (Ellis *et al*, 1991) and PREP (Neuwirth *et al*, 1993) have provided little or no constraint on user activity. In these cases the onus is on users to develop, enforce and maintain social protocols. Others such as Quilt (Leland *et al*, 1988) and Coordinator (Winograd, 1987) have imposed strict constraints based on theories of group activity or intuitions about roles and constraints. As contextual trust and cooperation threshold values are subjective and dynamic, TRUST has the potential to appropriately constrain activity between (and including) the extremes of no constraint and complete constraint.

One use that we described for TRUST was that of a tool for analysis of group work. We have established that it is implementable in software through our prototype implementations. Although TRUST may be used as a tool for imposing appropriate constraints it may be 'deactivated' in a system so that it does not actively constrain the activities of group members. In this case it could continue to record events as if it were active, to reveal the evolution of relationships, member behaviour in given contexts and so on. Its subjectivity and fine grain detail facilitates the capture of rich data about the group process.

Further Development of TRUST

Adoption of Discrete Positive Values

The use of negative TRUST values is effective in the description of an individual's view of the environment. However, when any two of $U_x(\alpha)$, $I_x(\alpha)$ and T_x are positive in

$$T_x(y, \alpha) = U_x(\alpha) \times I_x(\alpha) \times T_x$$

a small change from positive to negative in the third has a major effect on the outcome. Also two negative values and one positive on the right hand side of this equation produce a positive result. For this reason we will consider the adoption of positive values only.

Although real values provide subtle expressivity they may be unnecessarily fine grain. We will investigate the use of a finite set of rational number values that a TRUST variable can take.

Abstraction for Individuals and Contexts

It is unlikely that designers will be able to describe specific individual group members in advance. They may, however, be in a position to describe roles which may be adopted by, or allocated to group members. We will extend the formalism to allow such abstraction which will also support designers in describing constraints on activities associated with roles and how members in those roles may react to the behaviour of others in the group.

Extending the Formalism to Support Dissemination of Knowledge

So far, the formalism that we have presented allows an individual *x* to determine general and contextual trust for individuals which are not yet known. This is achieved with reference to the basic and general trust assigned to *x*. A more sophisticated, and perhaps more realistic, technique for determining trust in such a context is dissemination of knowledge. Perception of others in human relationships is determined, to some extent, by third-party perceptions in addition to personal experience. This approach can be introduced into an extended formalism.

Extending the Formalism to Support Inter-group Trust

As it stands, the formalism addresses only intra-group relationships. Group working is not only concerned with interactions within a single group, but also with inter-group interactions and individual to group interactions. Johnson & Kerridge (1992) have described why it can be just as desirable to support inter-group interactions as intra-group interactions. Often groups must work together to undertake large tasks. Just as optimal cooperation between individual group members can improve both the product and the process of the group work, so can optimal cooperation between groups. We may view groups as meta-individuals, whose attributes are derived from an amalgamation of the attributes of its members. The formalism will be extended to encompass this notion.

Introducing Heterogeneous Agents

The current formalism considers people involved in group work. This is useful for describing social relationships and can be translated into software to represent people and relationships. However, in CSCW systems, collaboration is not just person-to-person, but takes place via computer programs and data. We will consider the extension of the formalism to introduce a definition of an agent which will encompass people and the software and data that they use.

Summary

CSCW necessarily involves Human-Computer Interaction – it is concerned with supporting human-human interaction using computer software and hardware. In this paper we have emphasised the human-human aspects of CSCW such as the relationships between group members. We have provided a framework for representing, discussing and reasoning about group activity which includes a notation for trusting behaviour. This may be used by CSCW system designers and in the computer element of human-computer-human interaction to appropriately mediate or record group activity.

References

Argyle, M. (1991). *Cooperation; the Basis of Sociability.* Routledge, London.

Arion, M., Numan, H., Pitariu, H. and Jorna, R. (1994). Placing trust in human-computer interaction. *Proceedings of ECCE 7. Seventh European Conference on Cognitive Ergonomics,* Bonn, Germany, pp. 353-365.

Barber, B. (1983). *Logic and Limits of Trust.* Rutgers University Press, New Jersey.

Bok, S. (1978). *Lying: Moral Choice in Public and Private Life.* Pantheon Books: New York.

Deutsch, M. (1962). Cooperation and trust: some theoretical notes. In *Nebraska Symposium on Motivation* edited by M. R. Jones. Nebraska University Press.

Dykstra, E.A. and Carasik, R.P. (1991). Structure and Support in Cooperative Environments: the Amsterdam Conversation Environment. *International Journal of Man-Machine Studies* 34:419-434.

Ellis, C.A., Gibbs, S.J. and Rein, G.L. (1991) Groupware: Some Issues and Experiences. *Communications of the ACM,* 34(1):38-58.

Gambetta, D. (1990). *Chap. 2 of:* Gambetta, D (ed), *Trust,* Blackwell: Oxford.

Golembiewski, R.T., and McConkie, M. (1975). The centrality of interpersonal trust in group processes. In *Theories of Group Processes,* edited by Cooper, C.L., Wiley, pp. 131-185.

Johnson, K. and Kerridge, S. (1992) *Proceedings of the Third Belief Representation and Agent Architectures Workshop (BRAA '92).* Tech. Report 6/92, University of Durham, School of Engineering and Computer Science.

Lagenspetz, O. (1992). Legitimacy and Trust. *Philosophical Investigations*, 15(1):1-21.

Leland, M.D.P., Fish, R.S. and Kraut,R.E. (1988). Collaborative document production using Quilt. *Proceedings of the 2nd conference on Computer Supported Cooperative Work (CSCW '88)*. Portland, Oregon. September 1988, pages 206-215.

Luhmann, N. (1979). *Trust and Power.* Wiley.

Muir, B.M. (1987). Trust between humans and machines, and the design of decision systems. *International Journal of Man-Machine Studies*, (5-6):527-539.

Neuwirth, C.M., Kaufer, D.S., Chandhock, R. and Morris, J.H. (1993) Issues in the design of computer support for co-authoring and commenting. Pages 537-549 of *Readings in Groupware and Computer Supported Cooperative Work: Assisting Human-Human Collaboration*. Morgan Kaufmann.

Reiter, M.K. (1996). Distributing trust with the Rampart Toolkit. *Communications of the ACM*, 39(4):71-74.

Winograd, T. (1987) A Language/Action Perspective on the Design of Cooperative Work. *Human-Computer Interaction* 3(1), 3-30.

Zeckhauser, R.J. and Vicusi, W.K. (1990). Risk Within Reason. *Science*, 248:559-564.

About the Authors

Steve Jones is a Lecturer in Computer Science at the University of Waikato, New Zealand. His interests include Group Support Systems, hypermedia visualisation and navigation tools and visual query languages.

Steve Marsh is a Research Fellow in the Interactive Information Group at the National Research Council of Canada. His interests include information retrieval and provision which makes use of intelligent agents.

Authors' Addresses

Steve Jones. Department of Computer Science, University of Waikato, Private Bag 3105, Hamilton, New Zealand.
stevej@cs.waikato.ac.nz, +64 7 838 4490.

Steve Marsh. National Research Council, Institute for Information Technology, Building M-50, Montreal Road, Ottawa. K1A 0R6, Canada. steve@ai.iit.nrc.ca, +1 613 993 8553.