Working Paper Series
ISSN 1170-487X

A Dynamic and Flexible Representation of Social Relationships in CSCW

by Steve Jones and Steve Marsh

Working Paper 97/1
January 1997

Department of Computer Science
The University of Waikato
Private Bag 3105
Hamilton, New Zealand

A Dynamic and Flexible Representation of Social Relationships in CSCW

Steve Jones
Department of Computer Science
University of Waikato
Private Bag 3105
Hamilton
New Zealand

Steve Marsh
Interactive Information Group
National Research Council
Institute for Information Technology
Building M-50, Montreal Road
Ottawa
Canada

CSCW system designers lack effective support in addressing the social issues and interpersonal relationships which are linked with the use of CSCW systems. We present a formal description of trust to support CSCW system designers in considering the social aspects of group work, embedding those considerations in systems and analysing computer supported group processes.

We argue that trust is a critical aspect in group work, and describe what we consider to be the building blocks of trust. We then present a formal notation for the building blocks, their use in reasoning about social interactions and how they are amended over time.

We then consider how the formalism may be used in practice, and present some insights from initial analysis of the behaviour of the formalism. This is followed by a description of possible amendments and extensions to the formalism. We conclude that it is possible to formalise a notion of trust and to model the formalisation by a computational mechanism.

Introduction

In this paper we present 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 Computer Supported Cooperative Work (CSCW).

To date, most CSCW systems have been promoted on the strength of the technological advances that they incorporate, yet the challenge for groupware designers is twofold. They must provide appropriate technology *and* they must provide *appropriate* support for the social aspects of group activities. Designers lack effective tools to clearly describe social relationships within group work, to embed such descriptions in computer systems and to analyse the systems that they develop.

Here, we propose a tool to ameliorate this situation. 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 present our definition of trust and a formal description of it. We call this TRUST in order to differentiate it from wider definitions. We consider the potential uses of TRUST in a group work context. They include:

- it can be embedded in computer systems to mediate cooperative computer based activities;
- it can be used as a tool for the discussion of the design of CSCW systems;
- it can be used to record and analyse 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 and technological issues for designers involved in the development of *multi-user-centered* systems.

Defining TRUST

Trust is a common phenomemon that has been extensively discussed in the literature of sociology, social psychology and philosophy. 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 possiblities of everyday life. Golembiewski and McConkie (1975) have stated "Perhaps there is no single variable which so thoroughly influences interpersonal and group behaviour as does trust...". 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 peformance (Golembiewski & McConkie, 1975), greater and more healthy personal development and the ability to cooperate (Argyle, 1991; Deutsch, 1962).

However, any consideration of trust with respect to CSCW has had a narrow focus, with emphasis on security and privacy issues (Reiter, 1996) rather than its more pervasive role as a mediator of social interactions. Unfortunately, trust is so pervasive that it has many definitions. These are described in detail elsewhere (Marsh, 1994a), and have been integrated into the formalism that we present here.

We do not claim to hold the definitive view of trust and we describe a formalism that represents our view of trust (although it is flexible enough to describe other views within the constraints of the building blocks below). Consequently we shall refer to our formalism as TRUST to differentiate is form the wider notion of trust that we all have, and to some extent share.

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

Agents. Agents are the entities whose activities are mediated by a TRUST framework. Conventionally, social or psychological discussions of trust consider agents to be human and we follow this view here. Agents (people) may be collected into groups, and an agent may be a member of zero or more groups.

Situations. Interactions between agents have a context. The behaviour of a single agent may vary greatly from context to context. We call the contexts *situations*, which describe the tasks that agents may undertake collaboratively.

Basic trust. An agent in a group has a *disposition* derived from its previous experiences, which may change as a result of new experiences. We call this *basic trust.* It allows agents to make judgements in cases where the current situation or other agents are not known (Boon & Holmes, 1991).

Knowledge. Agents which have interacted with each other in the past are said to 'know' each other, and agents which have not interacted with each other in the past do not 'know' each other. A representation of knowledge provides an indication as to whether situation specific information is available to interacting agents.

General trust. Given that two agents know each other, they will have a general notion of how much they trust each other, regardless of the situation. We term this *general trust.*

Situational trust. If an agent has had an interaction with another agent, it has more information available to guide its actions. We call this information about interaction with agents in particular situations *situational trust.*

Importance. Interaction in any given context will have some level of importance attached to it by the agents involved. Agents are more likely to enter into interactions which are important than those that aren't.

Utility. Agents expect some return from an interaction. The expected return may be a loss, profit or something intangible. We call the expected return *utility.* Importance and utility are linked, but are distinct.

In the following section we present a formalism which consists of representation for the facets of trust descibed above, and formulae for manipulation of values assigned to those facets.

A formalism for TRUST

We have adopted simple probabilistic methods for describing TRUST . The formalism uses simple linear equations in order to estimate TRUST values. This has the advantage that quantitative data provides *sensitivity* in the formalism–small differences in individual values may contribute to a large difference in the outcome, and sensitivity can be amended by changing the grain of representation.

Representing the building blocks of TRUST

The following notations are summarised in Table I.

Agents and Situations

We represent agents (people) by $a, b, c, \ldots, z, a', \ldots, z', a'', \ldots, z''$ and so on. Individual agents are members of the set of all agents (everyone), $\mathcal{A}$. We represent *societies* or *groups* of agents, defined as a a number of agents (greater than 1) which is collected together according to some metric. We represent groups with $\mathcal{G}_1$, $\mathcal{G}_2 \ldots, \mathcal{G}_n \subset \mathcal{A}$.

We represent situations with Greek letters, with subscripts for the agent concerned since situations are agent-centered, or subjective. α_x is situation α

Description	Representation	Value Range
Agents	$a, b, c, \ldots \in \mathcal{A}$	
Situations	$\alpha, \beta, \ldots$	
Set of all agents	$\mathcal{A}$	
Groups of agents	$\mathcal{G}_1, \mathcal{G}_2 \ldots$	
	$\mathcal{G}_n \subset \mathcal{A}$	
Knowledge (e.g., x knows y)	$K_x(y)$	True/False
Importance (e.g., of α to x)	$I_x(\alpha)$	$[0, +1]$
Utility (e.g., of α to x)	$U_x(\alpha)$	$[-1, +1]$
Basic Trust (e.g., of x)	T_x	$[-1, +1]$
General Trust (e.g., of x in y)	$T_x(y)$	$[-1, +1]$
Situational Trust (e.g., of x in y for α)	$T_x(y, \alpha)$	$[-1, +1]$

Table I. Summary of the basic TRUST notation.

from x's point of view, β_y is situation β from y's[1]. Situations are represented $\alpha, \ldots, \omega, \alpha', \ldots, \omega', \alpha''', \ldots \in S$ and so on. Situations are taken to be members of the set of all possible situations in the world (a very large and open ended set!) notated S. If a set of agents exist in a group ($\mathcal{G}_1$) with a fixed set of situations, that set of situations is notated $S_{\mathcal{G}_1}$.

Knowledge of other agents

For agents, the concept of knowledge of other agents is important. Considerations of trust imply considerations of knowledge of the trustee, at least after the first interaction between two agents. $K_x(y)$ represents the fact that x has met y at some time and that x can remember it. $\neg K_x(y)$ represents the opposite. Thus, it is a boolean concept with a true or false value.

Basic trust

Agents are considered to be trusting entities. As such, they will have a 'basic' trust (Boon & Holmes, 1991; Govier, 1992). This is derived from past experience in all situations. T_x represents the basic trust of agent x and has a value in the range $[-1, +1]$. It is *not* the amount of trust an agent has in any other agent, or situation, or the environment; it is simply representative of the general trusting *disposition* (Boon & Holmes, 1991) of the agent.

General trust

To describe the relationship between agents $x, y \in \mathcal{A}$ we use $T_x(y)$ which has a value in the interval $[-1, +1]$. The value represents the amount of trust x has in y and is not relative to any *specific* situation (see below). A value of 0

[1] In the formulæ that follow, the subscript is often dropped. This is because it is evident which agent is doing the considering.

means x has no, or neutral, trust in y. A value of -1 would represent complete distrust and $+1$ represents complete trust.

Situational trust

Agents are based in situations. A situation means something different to each agent experiencing it. Thus, we have a representation for the amount of trust an agent has in another in a given situation. To describe the trust between x and y in situation α we use $T_x(y, \alpha)$[2]. Once again, this takes a value in the interval $[-1, +1]$.

Utility

For the amount of utility x gains from situation α, we write $U_x(\alpha)$ which has values over the interval $[-1, +1]$. The values can be normalised over this range. We take utility to be based in expected utility theory, with the overall utility of a particular situation, over all outcomes, as being the mean of these utilities of the outcomes (Zeckhauser and Viscusi, 1990).

Importance

We represent the importance of a situation α for agent x with $I_x(\alpha)$. It has a value over the interval $[0, +1]$. Negative importance is not considered here. This is because for the purposes of this work, we take situations of negative importance to be the opposites of situations of positive importance — "it is important that this is *not* done."

Manipulating facets of TRUST

To this point the formalism allows us to represent and quantify facets of group members (agents) and relationships between them. We can also represent and quantify some aspects of the group work situations that members may find themselves in. It is worth emphasising that these representations allow group members to hold differing views of each other and situations.

In order for TRUST to be used in mediating or analysing group work it must be extended to describe how members determine whether and to what extent to cooperative with each other. This is achieved by indicating how group members determine situational trust and the threshold above which they will cooperate in a given situation.

Determining situational trust

There are three states for truster x to consider in determining situational trust: the trustee y is not known at all, trustee y is known but not in the current situation, the trustee y is known in the current situation.

[2] Here, we drop the x from α_x, since the whole value is clearly taken from x's point of view.

In the first case ($\neg K_x(y)$) the situational trust is determined by the basic trust of x (T_x), modified by the utility and importance of the situation from the point of view of x. When $\neg K_x(y)$

$$T_x(y, \alpha) = U_x(\alpha) \times I_x(\alpha) \times T_x \quad (1)$$

In the second case ($K_x(y)$ but $\neg K_x(y, \alpha)$) the situational trust is determined by the general trust of x ($T_x(y)$), modified by the utility and importance of the situation from the point of view of x. When $\neg K_x(y, \alpha)$ and $K_x(y)$

$$T_x(y, \alpha) = U_x(\alpha) \times I_x(\alpha) \times T_x(y) \quad (2)$$

In the third case ($K_x(y, \alpha)$) the trust level for the trustee in the current situation is estimated from previous levels of trust in the trustee in the same situation. This estimate is derived 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 (Marsh, 1994b). This estimated trust is notated $T_x\widehat{(y, \alpha)}$. When $K_x(y, \alpha)$

$$T_x(y, \alpha) = T_x\widehat{(y, \alpha)} \quad (3)$$

Determining the cooperation threshold

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

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

The cooperation threshold is determined by the risk of cooperation in the current situation to the truster, the importance of the situation to the truster, and the perceived competence of the trustee in the current situation. Both Perceived_Risk and Perceived_Competence have values in the range [0,1]. The threshold is determined by

$$\text{Cooperation\_Threshold}_x(\alpha) = \frac{\text{Perceived\_Risk}_x(\alpha)}{\text{Perceived\_Competence}_x(y, \alpha)} \times I_x(\alpha) \quad (4)$$

In this case we consider that the more important the situation is, the more trust is required to cooperate. An alternative view may be that a high level of importance indicates a need to cooperate, implying a lower cooperation threshold. This second view can be represented by

$$\text{Cooperation\_Threshold}_x(\alpha) = \frac{\text{Perceived\_Risk}_x(\alpha)}{\text{Perceived\_Competence}_x(y, \alpha) \times I_x(\alpha)} \quad (5)$$

Determining perceived risk

In estimating the risk involved in cooperating in a situation α, agent x considers the perceived cost and the perceived benefit, notated $\text{Cost}_x(\alpha)$ and $\text{Benefit}_x(\alpha)$. Cost and benefit have values in the range [0,1]. The difference between cost and benefit, normalised to the range [0,1] gives the perceived risk:

$$\text{Perceived\_Risk}_x(\alpha) = \frac{1 + \text{Cost}_x(\alpha) - \text{Benefit}_x(\alpha)}{2} \tag{6}$$

Determining perceived competence

Perceived competence is determined in the context of the group in which the two members exist.

There are three cases to consider when determining the perceived competence of a truster in a trustee. In the first case, $\neg K_x(y)$ we use the disposition of the truster (basic trust, T_x) modified by the importance of the situation to x. When $\neg K_x(y)$

$$\text{Perceived\_Competence}_x(y, \alpha) = T_x \times I_x(\alpha) \tag{7}$$

In the second case ($K_x(y)$ but $\neg K_x(y, \alpha)$) the truster can consider the experienced competence of the trustee in all situations where the trustee is known. The competence of the agent under consideration is compared to the competence of the most competent agent in the group, giving a comparative value. When $K_x(y)$ but $\neg K_x(y, \alpha)$

$$\text{Perceived\_Competence}_x(y, \alpha) = \frac{\text{Average\_Competence}_y(S_{\mathcal{G}_1})}{\text{Highest\_Average\_Competence}_{\mathcal{G}_1}(S_{\mathcal{G}_1})} \tag{8}$$

In the third case ($K_x(y, \alpha)$), the truster considers the highest average fitness within the society in the current situation, and compares it to the average fitness of the trustee in the current situation. When $K_x(y, \alpha)$

$$\text{Perceived\_Competence}_x(y, \alpha) = \frac{\text{Average\_Fitness}_y(\alpha)}{\text{Highest\_Average\_Fitness}_{\mathcal{G}_1}(\alpha)} \tag{9}$$

Updating trust

An interaction between group members x and y may result in revisions to basic, general and situational trust values for the two agents. Therefore, we must extend the formalism to incorporate modifications to trust values.

After an interaction between group members situational trust is modified the most as the truster can consider with certainty the most recent behaviour of the trustee in the current situation. General trust is modified less than situational trust because the interaction was in only one of perhaps several situations. The smallest modification is made to basic trust because the interaction was with one of perhaps several agents in one of perhaps several situations.

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.

We notate the level of modification adopted across a group $\mathcal{G}_1$ by $M_{\mathcal{G}_1}$. The modification to situational trust is notated $M_{T_x(y,\alpha)}$. The modification to general trust is notated $M_{T_x(y)}$ and the modification to basic trust is notated M_{T_x}. So, if trustee y cooperates with truster x in situation α

$$T_x(y,\alpha) = T_x(y,\alpha) + M_{T_x(y,\alpha)}$$

$$T_x(y) = T_x(y) + M_{T_x(y)}$$

$$T_x = T_x + M_{T_x}$$

If trustee y does not cooperate with truster x in situation α

$$T_x(y,\alpha) = T_x(y,\alpha) - M_{T_x(y,\alpha)}$$

$$T_x(y) = T_x(y) - M_{T_x(y)}$$

$$T_x = T_x - M_{T_x}$$

In each case

$$M_{T_x} < M_{T_x(y)} < M_{T_x(y,\alpha)}$$

And the modification values are determined by

$$M_{T_x(y,\alpha)} = \frac{1+T_x}{2} \times M_{\mathcal{G}_1} \tag{10}$$

$$M_{T_x(y)} = \frac{\frac{1+T_x}{2}}{|S_{\mathcal{G}_1}|} \times M_{\mathcal{G}_1} \tag{11}$$

$$M_{T_x} = \frac{\frac{1+T_x}{2}}{|\mathcal{G}_1| \times |S_{\mathcal{G}_1}|} \times M_{\mathcal{G}_1} \tag{12}$$

Note that T_x is normalised to a value in [0,1].
The modifier $M_{\mathcal{G}_1}$ is determined by

$$M_{\mathcal{G}_1} = \frac{1+T_{\mathcal{G}_1}}{2} \tag{13}$$

where $T_{\mathcal{G}_1}$ is average basic trust of all agents in the group, determined by

$$T_{\mathcal{G}_1} = \frac{1}{|\mathcal{G}_1|} \sum_{w \in \mathcal{G}_1} T_w \tag{14}$$

Exploiting a formal representation of trust

Using TRUST in system design

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

- group members and the relationships between them;
- situations in which group members may find themselves;
- under what circumstances group members will cooperate with each other;
- 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. Obviously it is not possible for designers to describe individual group members in advance. They can, however, describe roles which may be adopted by, or allocated to group members. They can also describe constraints on activities associated with those roles and how members in those roles may react to the behaviour of others in the group. They can describe how far and at what rate members may diverge from their allocated roles.

Embedding TRUST in CSCW systems

The formalism has been developed with the goal that it is implementable in computer software. Designers may move from its descriptive use to to a direct representation in a programming language.

Once the formalism has been integrated into a software tool the values describing the group must be initialised. Several approaches may be taken to this. First, default values may be used. Member and situation attribute values may be set to zero, for example, indicating neither trust nor distrust at the outset of the group activity. Alternatively identified roles may have default sets of values to be used. Second, group members may assign values themselves to the attributes under their control such as their disposition, views of others and the importance of certain situations. Third, values may be initialised from historical TRUST data generated during members' use of this and other tools which incorporate TRUST . A mixture of these approaches may also be used.

The same piece of software is therefore configurable for use by multiple groups.

Mediating group interaction with TRUST

Groups may proceed with their activities mediated (or appropriately constrained) by TRUST . Some CSCW systems such as ACE (Dykstra & Carasik, 1991), GROVE (Ellis *et al.*, 1991) and PREP (Neuwirth *et al.*, 1993) have provided little or no constraint on user activity. Others such as QUILT (Leland *et al.*, 1988) have imposed strict constraints. As situational 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.

Analysing group interaction with TRUST

TRUST may be 'deactivated' in a system so that it does not actively constrain the activities of group members. However, it could continue to record events as if it were active, to reveal the evolution of relationships, member behaviour in given situations and so on. Its subjectivity and fine grain detail facilitates the capture of rich data about the group process.

Analysing TRUST

In order to analyse the formalism we have developed two software implmentations of TRUST. One uses the C programming language in a UNIX environment, and the other uses HyperCard (programmed in HyperTalk) (Apple Computer, Inc, 1987) in an Apple Macintosh environment. Through analysis of the formalism over repeated trials we have found that

- it is sufficiently abstract to enable implementation in both a conventional procedural language and an event based scripting language;
- during execution, the combination of zero as a meaningful value for attributes of trust and the use of division can lead to 'divide by zero' errors. Although ∞ is valid in the formalism, some alternative must be found in the implementation;
- the behaviour of the formalism is replicable. A fixed set of inputs (agents, their attributes, situations and values), and an ordered set of interactions result in the same outputs over multiple trials. Whether the outputs are sensible or not, they are not chaotic, which allows observation of the behaviour of the formalism with systematic manipulation of its attributes;
- the ordering of interactions is important in groups of more than two members. Experience determines future behaviour and so an interaction between x and y and then between x and z will produce a different state from x and z and then x and y although the same set of interactions has occured;
- although negative values seem intuitively useful to convey distrust they present a problem in some parts of the formalism. For example, in equa-

tion 1, to determine situational trust a single negative value will give a negative result, but two negative values give a positive result;

- the use of real values may produce results outside the allowable range of values. For example, in equation 4 perceived risk of 0.5, perceived competence of 0.1 and importance of 1 results in a cooperation threshold of 5.

Future work

Iterative analysis and amendment

In the future, immediate work will be concerned with addressing the problems described in the previous section. Next we will carry out analysis of the formalism using the two implementations mentioned earlier. Our main concern will be to identify the comparative importance of each attribute of trust in determining the behaviour of agents which exploit the formalism. This will, however, present some problems for more complex environments (more than two agents, more than a single situation) than those considered so far. In theory, given that all values are real, there are an infinite number of simulations to run for each attribute of trust which we make the independent variable. This ideal situation may be approximated by introducing a finite number of rational number values that a variable can take.

Extending the formalism to support dissemination of knowledge

So far, the formalism that we have presented allows an agent x to determine general and situational trust for agents 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 situation 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. (Johnson & Kerridge, 1992) has 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, or meta-agents, 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 agents to be 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 definition of agent to encompass the software and data used by and belonging to group members.

Conclusions

We have shown that formalising a notion of trust and modelling that formalistion by a computational mechanism is clearly possible. We have also argued that it is necessary in order to provide more effective support for the work of CSCW system designers.

The formalism is concise, has descriptive power and is easily modifiable and extensible, but we have also identified shortcomings which must be addressed in future work.

We believe that by using TRUST we can model extant CSCW systems along the continuum of constraint, from minimally constraining systems to those that impose role based constraints. It has the advantage that systems which use it are not fixed at a point on the constraint continuum. It allows for the application of appropriate and dynamic constraints.

The idea of persistent records of TRUST and relationships, that can be exploited across task and groups is attractive, and leads us to consider how TRUST might be used across multiple CSCW systems, including those in which it is not embedded. However, we must also consider the privacy and ethical issues of using it as a group work analysis tool, given that it records details of group members' work and relationships.

At the very least we provide a notation to be used in discussions regarding the nature of trust and its role in CSCW.

Acknowledgments

The first author carried out part of this work while on sabbatical leave supported by the Department of Mathematical and Computer Sciences, University of Abertay Dundee. The work has been undertaken in part by the first author at the Department of Computing Science, University of Stirling and the Department of Computer Science, University of Canterbury, New Zealand. The visit to the University of Canterbury was partially supported by the Carnegie Trust for the Universities of Scotland. Thanks to Steve Reeves for comments on earlier drafts of this paper.

References

Apple Computer, Inc. 1987. *Apple Macintosh HyperCard user's guide.* Apple Computer, Inc.

Argyle, Michael. 1991. *Cooperation: The Basis of Sociability.* London: Routledge.

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

Boon, Susan D., & Holmes, John G. 1991. "The dynamics of interpersonal trust: resolving uncertainty in the face of risk". *Pages 190–211 of:* Hinde, Robert A., & Groebel, Jo (eds), *Cooperation and Prosocial Behaviour.* Cambridge University Press.

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

Dykstra, EA, & Carasik, RP. 1991. "Structure and support in cooperative environments: the Amsterdam Conversation Environment". *International Journal of Man-Machine Studies,* **34**, 419–434.

Ellis, CA, Gibbs, SJ, & Rein, GL. 1991. "Groupware: some issues and experiences". *Communications of the ACM,* **34**(1), 38–58.

Golembiewski, Robert T., & McConkie, Mark. 1975. "The Centrality of Interpersonal Trust in Group Processes". *Chap. 7, pages 131–185 of:* Cooper, Cary L. (ed), *Theories of Group Processes.* Wiley.

Govier, Trudy. 1992. "Trust, Distrust, and Feminist Theory". *Hypatia,* **7**(1), 16–33.

Johnson, Kevin, & Kerridge, Simon. 1992. *Proceedings of the third Belief Representation and Agent Architectures Workshop (BRAA '92)*. Tech. rept. 6/92. University of Durham, School of Engineering and Computer Science.

Lagenspetz, Olli. 1992. "Legitimacy and Trust". *Philosophical Investigations*, **15**(1), 1–21.

Leland, MDP, Fish, RS, & Kraut, RE. 1988. "Collaborative document production using Quilt". *Pages 206–215 of: Proceedings of the 2nd conference on Computer Supported Cooperative Work (CSCW '88)*. Portland, Oregon. September 1988.

Luhmann, Niklas. 1979. *Trust and Power*. Chichester: Wiley.

Marsh, S. 1994a. *Formalising trust as a computational concept*. Ph.D. thesis, University of Stirling, Scotland.

Marsh, Stephen. 1994b. "Optimism and pessimism in trust". *In:* Ramirez, J (ed), *Proceedings Iberoamerican Conference on Artificial Intelligence/National Conference on Artificial Intelligence (IBERAMIA94/CNAISE94)*. McGraw-Hill.

Neuwirth, CM, Kaufer, DS, Chandhock, R, & Morris, JH. 1993. "Issues in the design of computer support for co-authoring and commenting". *Pages 537–549 of:* Baecker, RM (ed), *Readings in groupware and computer supported cooperative work: assisting human-human collaboration*. Morgan Kaufmann.

Reiter, MK. 1996. "Distributing trust with the Rampart Toolkit". *Communications of the ACM*, **39**(4), 71–74.

Zeckhauser, Richard J., & Viscusi, W. Kip. 1990. "Risk within Reason". *Science*, **248**(May 4th), 559–564.