Trust: A Distributed Cognition Approach[1]

Sanjay Chandrasekharan

Center for Adaptive Behavior and Cognition,
Max Planck Institute for Human Development,
Lentzeallee 94, D-14195,
Berlin, Germany

schandra@mpib-berlin.mpg.de

Cognitive Science Ph.D. Program,
Institute of Interdisciplinary Studies,
2210, Dunton Tower, Carleton University,
Ottawa, Canada

schandr2@chat.carleton.ca

Summary

We outline the notion of trust and two formal models developed to solve the trust problem in multi-agent systems. The limitations of these models, and their similarities with classical AI models (which stress centrally-stored representations of the world) are examined. We then consider trust as a Distributed Cognition problem, and suggest an agent design framework, inspired by Distributed Cognition. A distributed model of trust is then developed, extending work by Bacharach and Gambetta on trust in signs, and based on Zahavi's Handicap Principle (a theory of animal signaling that emphasizes the role of costs in ensuring signal reliability). We apply this model to agent systems to suggest a programming language that can act as an institution to partially solve the trust problem.

Keywords: Trust, Multi-Agent Systems, Agent Design, Distributed Cognition, Handicap Principle

[1] Carleton University Cognitive Science Technical Report 2002-12
URL http://www.carleton.ca/iis/TechReports

Comedian Steven Wright points to a glass of water, and says: “I mixed this myself. Two parts H, one part O. I don' t trust anybody!”

Wright shows brilliantly how far we can go without the notion of trust – not very far. Put bluntly, people and societies cannot function without trust, whether we are dealing with objects, people or processes. On the Internet, which at one level is a huge society, trust is not just an issue of theoretical importance -- it is an issue that affects the Net here and now, especially with the development of mobile agent search tools like Gossip and file-sharing tools like Gnutella and Pointera[2]. Trust is also a major issue for auction sites like E-bay, which has trust ratings and decided some time ago to block agent programs from accessing the site’s contents.

The trust issues raised by agent technologies and peer-to-peer systems are largely security-related, but the trust problem has other facets as well, particularly ones relating to competence and learning in multi-agent systems. In this paper, we will focus on trust in delegation in multi-agent systems. A trust mechanism is needed for agent frameworks, since agents are conceived and developed as autonomous entities. This results in a situation where there is no guarantee for the performance of an agent system created out of random agents. The system is open to security and competency threats. Currently, it is assumed that all agents are benevolent, an assumption that is quite unjustified (Marsh, 1994), particularly if agents are to be deployed in a wide scale.

[2] Gossip: http://www.tryllian.com
Gnutella: http://gnutella.wego.com
Pointera: http://www.pointera.com

The trust problem in multi-agent systems takes various forms. However, we consider the following to be the central question: how can an agent trust another agent, if that agent is unknown? There are three general forms of the problem, based on the task the agents need to perform[3]. They are:

- *Trust for Security*: Agent B asks Agent A for access to parts of A's system. To give access, Agent A needs to know whether Agent B is malevolent or benevolent — that is, will B harm A's system?

- *Trust for Learning*: Agent A needs to learn X from Agent B. For this, A needs to know whether B's beliefs about X is true and justified, and relevant to A's functional role. That is, is B's knowledge of X correct and relevant?

- *Trust for Delegation*: Agent A needs to delegate a task to Agent B. For this, A needs to know whether B has the competence to do the task, and whether B will in fact do the task, given suitable external conditions (Castelfranchi, 1999).

We consider the solution to the problem of trust in agent systems as: *designing agents that can be trusted, and communicating this design to the agents that do the trusting*. The problem of designing an agent that can be trusted is different for each of the above situations.

Most current models of trust focus on the first or the last of the above categories. They also focus on the *trusting agent* (the agent that trusts, trustor), rather than on the *agent that is trusted* (trustee). In this paper we will sketch two such approaches, and argue that the models of trust developed by these approaches are limited and they do not provide a framework for developing solutions to the problem of trust. In particular, we will argue that the approaches focus too much on the trustor, ignoring the role of the trustee, the environment, and communication, in trust formation. The layout of the paper is as follows: In Section 1, we present definitions and some categories of trust, and an overview of concepts closely related to it. In Section 2 we sketch two dominant formal models of trust, and our reasons for thinking why they are limited. In Section 3, we relate these models to models of Artificial Intelligence (AI), and draw parallels between these models and classical, head-centered approaches to AI. Arguments are presented as to why a distributed model of trust is needed, using a framework that treats classical AI models and situated AI models as two ends of a continuum. We then consider the trust problem as a Distributed Cognition problem, and suggest an AI methodology that is inspired by Distributed Cognition. Section 4 goes back to trust, and sketches an alternate model of trust, based on the relationship between trust and representation. The crucial role of communication in trust is explored. In section 5, we use the methodology developed in section 3 to understand institutional signs, using work in animal signaling by Zahavi (1997) and its extension into trust by Bacharach and Gambetta (2000). In Section 6, we combine the idea of institutions with our alternative narrative on trust and suggest a programming language that can solve the trust problem in agent systems partially, by

[3] All three cases presuppose possible cooperation between agents.

acting as an institution. We end with the limitations of our methodology and areas of future work.

All along the paper, we will take a human-centered approach to the problem of trust — that is, start with the analysis of the human version of the problem and then look at how it relates to artificial agents. We believe this approach is needed for two reasons. One, in the limiting case of Distributed AI, the human agent is part of a diverse agent society. Two, we think that the following argument is applicable to the trust problem: *Agents do not harm agents, people harm agents.* As Khare and Rifkin (1997) point out in the context of computer security, the trust problem between computers bottom out to a trust problem between humans. Similarly, we think the trust problem between agents, too, bottom out to a trust problem between humans. People are ultimately accountable for an agent's actions.

We start with some definitions and categories of trust.

1. What is Trust?

Gambetta (1990) gives the following definition of trust, which is commonly accepted, judging by the frequency of its appearance in the literature:

…trust, (or symmetrically, distrust) is a particular level of the subjective probability with which an agent will perform a particular action, both before he can monitor such action

(or independently of his capacity to monitor it) and in a context in which it affects his own action.

The term "subjective probability" is important in the above definition, because it points to a certain amount of arbitrariness in the trust metric. Thus, trust is not something that can be captured fully using objective measures. Trust is not an objective property, but a subjective degree of belief about others' competence and disposition. So, as Dunn (1990) points out, "however indispensable trust may be as a device for coping with the freedom of others, it is a device with a permanent and built-in possibility of failure".

Marsh (1994) stresses the point made by Luhmann (1990), that "trusting a person means the trustor takes a chance that the trustee will not behave in a way that is damaging to the trustor, given that choice". In general, "trust …presupposes a situation of risk."

Marsh (1994), integrating various views, argues that trust is:

- A means of understanding and adapting to the complexity of the environment
- A means of providing added robustness to independent agents
- A useful judgement in the light of experience of the behaviour of others
- Applicable to inanimate others (including artificial agents)

Kinds of Trust

The label "Trust" is quite amorphous, and is applied to a range of phenomena, involving objects, processes, and people. Three general types of trust have been identified.

Dispositional trust describes an internal state of the trustor, a basic trusting attitude. This is "a sense of basic trust, which is a pervasive attitude towards oneself and the world" (Abdul-rahman, 2000). This trust is extremely open-ended and independent of any party or context. Dispositional trust has been further divided into two – type A concerns the trustor's belief on others' benevolence, type B is the "disposition that irrespective of the potential trustee's benevolence, a more positive outcome can be persuaded by acting "as if" we trusted her". (McKnight et al, quoted in Abdul -rahman, 2000).

Impersonal trust refers to trust on perceived properties or reliance on the system or institution within which the trust exists. An example is the monetary system (Abdul-rahman, 2000). This can also be seen as dispositional trust directed towards an inanimate system. Impersonal trust is related to the notion of trust involved in learning, where Person *A*, while learning something from Person *B*, trusts that the facts s/he learned are true. Part of this trust is based on experience, part of it on institutional settings. This is also related to the developmental aspects of trust – how trust develops in infants and children — discussed in detail by Lagenspetz (1992) and Hertzberg (1988).

Interpersonal trust refers to the trust one agent has on another agent directly. This can be seen as dispositional trust directed towards an animate system. This trust is agent and

context specific. For instance, Person *A* might trust Person *B* in the context of fixing a furnace, but not for fixing a car.

Sometimes the word "trust" is used interchangeably with "faith". In this sense, "I trust him", implies that "I have an unjustified (perhaps unjus tifiable) belief that he will do the right thing" (Castelfranchi, 1999). This is closely connected to dispositional trust and another notion of trust, where Person *A* has known Person *B* for a long time, and has interacted with him/her extensively. *A* now trusts *B* in a "non-specified" manner. That is, the set of situations for which *A* trusts *B* is an open set. The responsibilities of *B* are not set out in advance, and sometimes *B* may not even know what his/her responsibilities are. This is a complicated notion and we will call this *Open Trust*. This is person-specific and not as all- enveloping as dispositional trust.

Finally, the word "trust", as used in common parlance, implies something people have inside them — a fluctuating internal state, a sort of meter that goes up or down, depending on the situation and people involved. There is talk about "trust levels". Trust is also considered to have qualia, a phenomenal feeling of trusting or being trusted, associated with it. A breach of trust results in emotional changes, interestingly, *both* in the trustor and the trustee. It is also interesting to note that "the loss or pain attendant to unfulfillment of the trust is sometimes seen as greater than the reward or pleasure deriving from fulfilled trust" (Golembiewski et al, quoted in Marsh, 1994).

Characteristics of Trust

As pointed out earlier, trust is not an objective property, but a subjective degree of belief about a person, process or object. The degree can vary from complete trust to complete distrust. There is also a situation where a person does not have an opinion on the trustworthiness of another person— i.e. the person is ignorant of the other person's trustworthiness.

Trust is not a blind guess or a game of chance, even though it involves a decision taken on the face of uncertainty. Trust involves a decision taken in anticipation of a positive outcome, and the decision is based on the knowledge and experiences of the trustor. It is this knowledge and experience that makes trust more than a blind guess. Abdul-rahman (2000) points out that trust reasoning is inductive. It is also dynamic and non-monotonic – additional evidence or experience at a later time may increase or decrease our degree of trust in a person.

However, Lagenspetz (1992) gives a counterexample to the role of induction in trust. A squirrel comes to a tree every morning. The regularity of the arrival of the squirrel does not lead to the establishment of trust between an observer and the squirrel. According to Lagenspetz, *a trustful relation can develop only from a shared life*, not merely from observation. For Lagenspetz, trust is a two-way relationship, and needs equal contributions from the trustor and a trustee. Most current models ignore the role of the trustee in the trust relationship.

An important feature of trust is that it cannot be brought about by will. The statement "trust me" does not work unless trust is present in the first place (Marsh, 1994). "I cannot will myself to believe that X is my friend, I can only believe that he is." Lagenspetz (1992) argues that trust is innate in children, and hence it is a fundamental mental state, and related to rationality and doubt. Drawing on Wittgenstein, he says that "one must begin somewhere, begin with not-doubting. This is not hasty and excusable, but is part of the process of judgement." Doubt comes after belief. All judgements must be seen in the context of an initial belief. One must first have faith to be able to lose it later.

Trust also has the interesting property that the seeking of evidence for trust affects the evidence. Once distrust sets in, it is difficult to know if such distrust is justified, because such experiments will not be carried out. Trust is thus capable of spiraling dramatically downwards (Marsh, 1994). On the other side, it is also capable of spiraling dramatically upwards, and can be self-reinforcing. Trust also has the property that it grows with use, and decays with disuse. Dasgupta (1990) points out that this property of trust makes it similar to other moral resources.

Trust is closely related to confidence as well. The difference between the two is that trust presupposes an element of risk, while confidence does not (Luhmann, 1990). In a situation of confidence, alternatives are not considered. As Marsh (1994) points out, leaving the house without a gun every morning shows confidence in not needing to use the gun. However, leaving the house every morning without a gun, after considering the probability of having to use the gun that day, shows trust.

Formally, the trust relation is not transitive. That is, if *A* trusts *B* and *B* trusts *C* for a certain action *X*, it does not necessarily follow that *A* will trust *C* for the same action. However, trust is considered weakly transitive, in the following sense: there is a probability that *A* might trust *C*, if *B* recommends *C* to *A* for action *X*[4].

The notion of *reputation* (Abdul-rahman, 2000, Dasgupta, 1990) uses this weak transitivity of trust. Dasgupta (1990) considers reputation as a capital asset, and observes that, like trust, it is acquired slowly and destroyed quickly. Trust and reputation are complementary; one builds the other.

Dasgupta also considers the relation of trust to various institutional mechanisms that societies use to guarantee trust -- like punishment (or ostracizing) for breach of trust, the threat of such an action, the enforcement of it, etc. It is interesting to note here that all these mechanisms are based on qualia. The punishment and ostracizing threats work because the feeling of being in jail or being ostracized (shame) does not *feel* good to human agents. The threats, as they stand, will not work for, say, a robot, or any such artificial agent.

Given these characteristics of trust, let us consider two formal models put forward to capture trust.

[4] It is not clear that what is transferred here is trust. It could be *B*'s experience with *C*, which *A* uses as one of the parameters to arrive at his own trust level.

2. Formal Models

For exposition, we will ignore most of the other notions of trust and focus on trust that involves some amount of formal decision-making on the part of the trustor, and delegation of responsibility to the trustee. We will call this case *reliance*, which is a subset of the general category of interpersonal trust. The following is a non-exhaustive description of such a situation involving trust[5]:

- There is a situation α.

- There are two entities involved, the trustor (x) and the trustee (y). The trustor and the trustee can be organizations or groups.

- There is an action to be performed.

- The trustor is dependent on the trustee(s) to execute the action.

- The execution of the action affects the trustor, but not *necessarily* the trustee. (However, in most real-life trust situations, the execution of the action affects the trustee *positively*.)

[5] A variation of this theme is involved in Prisoner's Dilemma (PD) situations, which have been studied extensively in game theory and the evolution of cooperation (Axelrod, 1994). In PD, the trustor and the trustee have equal control over the action, and each can execute the action. There is no act of delegation. We will ignore work on PD in this analysis because most of the work in PD does not talk about delegation and trust. The focus in PD is on the evolution of cooperation, given a set of variables.

- The trustor has control over the allocation of action. In the sense that s/he can decide between a set of trustees (A, B C…) to execute the action. S/he may also have control over the extent of the action to be allocated. That is, the action could be allocated partially or fully to particular trustees.

- Once the action is delegated, the trustor has minimal control over the action.

- The trustor executes a speech act (a paper or oral contract), delegating the action to the trustee.

- The trustee performs the action.

- The trustee returns control to the trustor once the action is performed.

Like Prisoner's Dilemma games, this situation can be a one-time process, or part of a long-term interaction. As in PD, the long-term interaction scenario brings in a "shadow of the future" to the calculation of trust.

Model I

Marsh (1994) suggests a series of formalisms for capturing *x*'s trust in *y*, including formalisms for situations involving memory and reciprocation. The basic, and the most important, among the formalisms are listed below. They suffice for our discussion.

$$\mathbf{T}_x\,(y, \alpha) = \mathbf{U}_x(\alpha) \times \mathbf{I}_x(\alpha) \times \hat{\mathbf{T}}_x\,(y)$$

Where $T_x\,(y, \alpha)$ is the trust *x* has in *y* in situation α,

$U_x(\alpha)$ is the utility *x* gains from situation α,

$I_x(\alpha)$ is the importance of situation α for agent *x*,

and $\hat{T}_x\,(y)$ is an estimate of *general trust*, the amount *x* trusts *y*. This is *x*'s estimate after taking into account all possible *relevant* (our emphasis) data with respect to $T_x\,(y, \alpha)$ in the past. According to Marsh, this is different from *basic trust*, which is a disposition to trust. Basic trust is not directed to any particular agent or situation, but is a state derived from past experiences. General trust is a token of basic trust — basic trust directed towards a particular agent.

Marsh argues that this trust needs to be part of the formula to calculate trust. This is because to assess the situational trust *x* has in *y* in situation α, *x* has to consider the knowledge *x* has of other trust situations involving *y*. This information is considered to be embedded in the general trust values *x* has for *y*. **(**For simplicity, we will use ψ to denote this component of the formula from here on.)

The second important formalism by Marsh suggests a cooperation threshold, above which the agent will decide to cooperate.

$$\textbf{Cooperation\_Threshold}_x(\alpha) = \frac{\textbf{Perceived\_Risk}_x(\alpha)}{\textbf{Perceived\_Competence}_x(y,\alpha) + \psi} \times \mathbf{I}_x(\alpha)$$

The terms of the formula are self-explanatory. The cooperation threshold is considered to be a "subjective measure, tempered by objective beliefs". Marsh considers different variations of the formula, including ones where the competence of the trustee is not known in advance.

Limitations

- The model considers the trustee to be a passive entity. All the action happens at the trustor's end. She takes the decision to delegate after a series of calculations, based on experience and other such factors. However, in the real world, the trustee is never passive. She constantly sends out signals to the trustor, either positive or negative.

- The role of the environment is not captured. Though trust is considered to be 'situated', the notion of situatedness here is a limited one — used in the sense that the trust decision happens in a 'situation' and can vary across 'situations'. A situation is considered as something like a "box" or a "framew ork" within which the trusting decision is made. Also, for the model to work, situations need to be identifiable as similar or dissimilar. This situatedness is different from the larger notion of 'being in the world' (Clark, 1997), where events are continu ous. In the real world, a situation is

not a slice of time and space, but a broader intermingling of contexts.

To understand this notion better, think of A considering entering the taxi of B, who has just come out from a bar. Suppose that A has been driven safely by B for a number of times before, and B does not show any outward symptoms of being drunk. Should, or would, A trust B in this particular situation? Most likely not, even though B probably meets all the criteria set out in the model. It is interesting to note here that there is a possibility of B not being drunk, maybe s/he went in to check on a friend. But, in spite of that, A probably would not trust B. What complicates the trust situation here is the *context*, which is external to the driving/driven situation, and part of a larger worldview.

- As observed earlier, trust is closely connected to reputation and social institutions. The role of these institutions is assumed, particularly in the Perceived_Competence variable, but the roles are not captured formally by the model. The Perceived_Competence variable is too broadly defined, and the mechanisms and the parameters that govern the perception are not specified.

- There is considered to be a distinguishable and independent state of mind called trust. However, there is a set of mental states (like beliefs) that contribute to trust (Castelfranchi,1999). The role played by these background beliefs in trust is not explored. Particularly, how beliefs are revised after a trusting (or symmetrically, non-trusting) decision.

- The crucial role played by communication in trust is not captured.

- The notion of *relevant* information is a bottleneck. How does the agent know what is relevant? This leads to the frame problem.

Model II

The second formal model, suggested by Castelfranchi (1999), is more cognitively oriented. The model considers the role of beliefs early on, and makes the basic assumption *that only an agent with goals and beliefs can trust*. The model considers trust to be a "cluster" mental state. Trust is thus compositional, and is made up of some basic ingredient beliefs. The degree of trust is based on the "strength" of its component beliefs.

One of the interesting suggestions put forward by the model is that *trust is the mental counterpart of delegation*. The model is thus delegation-driven. Delegation is an action; a set of beliefs contributes to the action of delegation, and once an action is delegated, this cluster of beliefs makes up trust.

Castelfranchi points out that the decision to delegate has no degrees, it is an either/or decision. However, beliefs have degrees. So trust has degrees as well. Essentially, the action of delegation arises when cumulative degrees of belief reach a threshold. This is quite similar to the idea of the cooperation threshold suggested by Marsh (1994).

Another important point made by Castelfranchi (1999) relates to the notion of social relations. When *x* delegates a task *q* to *y*, the action of delegation creates a social relation between *x*, and *y* for the action *q*. This relation "binds" the agents and creates memory. This is important, and we will return to this point later.

Castelfranchi breaks up trust into the following beliefs.

Competence belief: *x* should believe that *y* can do action α

Disposition belief: *x* should believe that *y* is willing to do α

Dependence belief: *x* believes it has to rely on *y* (strong dependence) or *x* believes it is good to rely on *y* (weak dependence)

There are other beliefs that contribute to the decision, which are related, but not entirely independent of the above beliefs[6].

Fulfillment belief: *x* believes that goal *g* will be achieved (thanks to *y* in this case)

Willingness belief: *x* has to believe that *y* has decided and intends to do action α

Persistence belief: *x* believes that *y* is stable in his intentions, and will persist with α

[6] This is work in progress.

Self-confidence belief: *x* believes that *y* knows that *y* can do α

Castelfranchi introduces some predicates to formalise his notion of trust, including some ad-hoc predicates like WillDo and Persist. Thus, formally,

$$\textbf{Trust (X, Y, t)} = \textbf{Goal}_x t \wedge \textbf{B}_x\textbf{PracPoss}_y(\alpha, g) \wedge \textbf{B}_x\textbf{Prefer}_x(\textbf{Done}_y(\alpha, g), \textbf{Done}_x(\alpha, g)) \wedge (\textbf{B}_x(\textbf{intend}_y(\alpha, g) \wedge \textbf{Persist}_y(\alpha, g)) \textbf{ or } (\textbf{Goal}_x(\textbf{Intend}_y(\alpha, g) \wedge \textbf{Persist}_y(\alpha, g)))$$

The predicates are self-explanatory. The model suggests that given these beliefs, *x* will usually trust *y*.

Limitations

One of the problems with the formulation is the broad scope of the competence belief. For instance, it is not clear why a "fulfillment belief" is needed, given the "competence" and "disposition" beliefs. If the "fulfillment belief" refers to the nature of the task (whether it can be done or not), then the "competence" belief has to be defined more narrowly. A broad definition of competence covers fulfillment as well. The way the beliefs are defined currently, the combination of "competence" and "disposition" exhausts "fulfillment". There are some other redundancies as well.

Another problem is the definition of trust, as being strictly relative to a goal. This is not entirely true. In the case of *Open Trust* (see Section 1), an agent trusts another for an open set of tasks. No goals are specified in advance in such a case.

The model also depends on the modeling of the trustee's (*y*'s) mental state to get to a trust metric. All the beliefs that *x* has relate to *y*'s mental state. How *x* arrives at a belief about *y* is not specified. The role of the environment is considered only marginally, as facilitating or negatively affecting the execution of the action.

Besides these limitations, we think most of the problems pointed out in the Marsh model are applicable to this model as well, including the large focus on the internal state of the trustor. The model also ignores communication, which is a crucial component of any trusting decision.

Let us now stand back a little from these models and look at how they relate to higher models of cognition.

3. Trust, AI, Distributed Cognition

We think the two formal models of trust presented above are in many ways similar to the "head-centered" approach to intelligence suggested by traditional models of Artificial Intelligence. For instance, trust is considered as "being in the head" of the trustor — it is a mental state of the trusting agent. The agent creates or computes a central representation of the trust metric from inputs from the environment, compares the value with a built-in threshold, and depending on the output, executes an action. The general picture is the traditional AI one, where a centrally stored, idealised, representation computes inputs from the environment and executes actions based on these computations.

As observed by Brooks (1997), to port to a system, this notion of intelligence needs an objective world model provided by the programmer. This model would then be compared against 'situations' and agents in the world. This is the "Blocks World" or the "Toy World" approach to Artificial Intelligence. It presupposes a world that matches the objective world model stored within the agent. As Brooks (1997) has convincingly argued, this is not a robust way of building intelligence, because the world does not come in readymade templates.

Both the trust models sketched above follow the traditional AI picture. The models analyse the trust problem by breaking it up into subparts. The lashing back of these parts is considered to produce an evaluation of trust in any given environment. There are assumed to be template situations and competencies. The evaluation of trust is very similar to the idea of pattern-matching in traditional AI, where a programmed pattern is compared to the input from the environment, and if they match, an action is executed. In the case of trust, the environment includes another agent, which brings in the perceived state of his mind as a variable. This means that for modeling trust, an "objective model" of the participating agent's mental state has to be provided by the programmer.

This model of trust, like the model of intelligence, has limited applicability when the environment and the other agent do not come with any structure that the program has information about. For situations that can be compared to a stored pattern, the model works. However, there is a wide range of trusting situations, and there is no guarantee

that the idealised patterns provided by the programmer can cover all those situations. Ideally, every situation and agent needs to be assessed before the trust decision can be taken. Some situations need less assessment, because the situation is similar to ones the agent has experienced before. Here the agent can use the built-in representation, to which the situation can be compared. However, many of the trusting situations would be new and different from previously experienced ones. There is no *necessary* structure to a trusting situation, which the agent can compare to an internal model to arrive at a trusting decision.

What does a non-artificial agent do when it is faced with a situation that does not follow a stored pattern it has extracted from the environment? It senses the environment. This is what Brooks' robots try to emulate. They make up for the lack of central representation by sensing, by querying the world often, and base their decisions on the information they gain from the environment at run-time.

When an agent is faced with a situation (world) that it cannot compare with an internalised structure to base a decision on, the agent has to do a large amount of querying, and use that information to arrive at decisions. Intelligence, thus, involves a trade-off between what is stored inside the agent and what is stored in the world. We have argued this in detail elsewhere (Chandrasekharan and Esfandiari, 2000).

The following diagram illustrates how the designer's assumptions about structure in the world is related to the sensing (or querying) that the agent has to perform. The case on the

left shows classical, head-centered agent design. The one on the right is the Brooks' sense-react model.

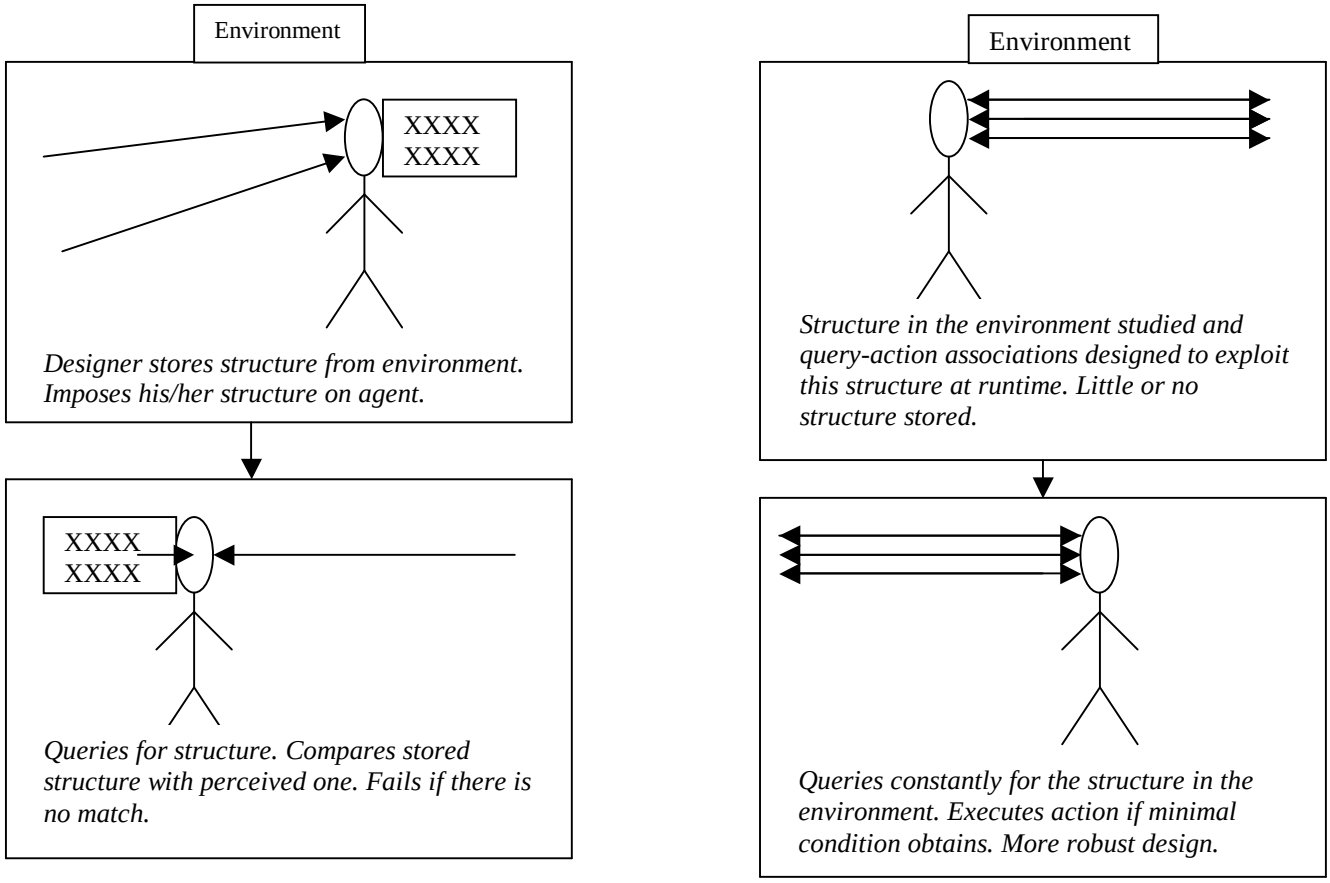


Figure 1

In simple terms, the more structure the agent (or programmer) has extracted and internalized from the world, the less the agent needs to query. All the agent needs to do is look for a stored structure in the world. This model has the problem that if the stored structure does not obtain, there is no action.

When little or no structure is internalised during the design, the agent needs to query the world a lot more to compensate. When there are a significant number of queries, the

world can be considered as contributing to the decision. The locus of intelligence cannot be strictly determined in this case, and we are licensed to talk about emergence. Thus, the classical AI model of comparison with an internally stored structure and the situated AI model of emergence are two ends of a design continuum. We capture this relation in the following diagram.

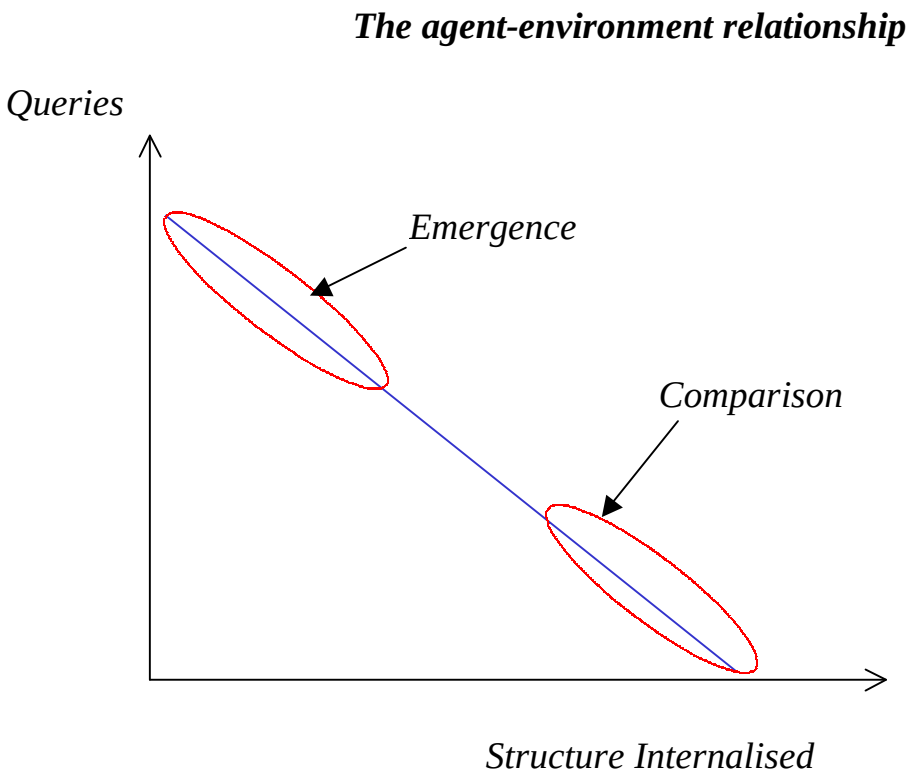


Figure 2

We believe that there is a lack of readymade structure in most trust situations. So the emergence model is a better one for solutions to the trust problem. In such a model, the agent will depend on *content-checking* queries (communication), unlike sonar queries like in the case of Brooks' robots, to find out about the nature of the trustee. This communication aspect makes the human situation more an emergent phenomenon than a

simple pattern-matching one. In most human trust situations, there is a series of communicative interactions that happen between the trustor and the trustee. Moreover, the trustee takes an effort to heighten the trustor's trust in him/her, and a positive decision by the trustor usually results in a positive outcome for the honest trustee. There is negotiation and dialogue between the trustor and trustee, and the "situation" varies depending on this negotiation. The evaluation of trust is thus a dynamically changing function, depending heavily on the reactions of the trustee to the trustor's queries. It is not a simple calculation or comparison that the trustor performs, based on his/her beliefs[7].

There's an important point to note here: the agents involved in the trust relationship communicate, and they use representations to do this. Therefore, they are situated not just in the world, and their actions are not representation-free, as argued by Brooks. They also share a social framework — they are *socially situated*. As Lagenspetz (1992) observed, *a trustful relation can develop only from a shared life*, not merely from observation. Castelfranchi (1999)'s observation is also relevant here: when *x* delegates a task *q* to *y*, the action of delegation creates a *social relation* between *x* and *y* for the action *q*. This relation "binds" the agents and *creates memory*. The creation of memory creates representations as well. Therefore, this kind of situatedness is a level above the being-in the-world kind of situatedness that Brooks refers to. We refer to this kind of situatedness, involving a shared life and representations, as *social situatedness*. Most importantly,

[7] Interestingly, trust spans both ends of the graph. A trust decision can be based on simple comparison as well. Like in the case of catching a bus. You trust the bus to take you to the destination, without resorting to too much querying. This is because the bus comes with a structure — a color, a board, a number. So the bus-catching situation today is just matched to the bus-catching situations in the past. In the same way, a trust situation similar to previously experienced ones will result in a simple comparison reaction. Experience, thus, brings a situation down to the structure-comparison level from the emergence level in the graph.

s*ocial situatedness involves trade in representations*[8]. This explicit acceptance of representations makes an environment-oriented approach to trust different from situated AI as advocated by Brooks. We think Distributed Cognition (Hutchins, 1995; Hollan et al, 2000; Kirsh 2001) is a much better candidate to accommodate an AI model of trust that focuses on the environment, while accepting the role of representation in cognition (For a systematic treatment of the distinctions between distributed and situated cognition, see Nardi, 1996 and Susi, 2001).

Distributed Cognition (DC) considers intelligence to be spread out among other agents and functional contexts, and it emphasizes both representations and the role of the environment. Cognitive processes are considered as distributed across the members of a social group, and the functioning of the cognitive system involves coordination between internal and external structure. Processes are also distributed across time, so earlier events can influence later events. The primary unit of analysis in the DC framework is a distributed socio-technical system, which consists of people working together and the artifacts they use. Individuals and artifacts are described as nodes, or agents, in this complex cognitive system. Behavior results from the interaction between external and internal representational structures.

The distributed cognition approach assumes that cognitive systems consisting of more than one individual have different cognitive properties from the cognitive properties of individuals that participate in such systems. The analysis of one individual's cognition in

isolation will not provide us with an understanding the system. If the task is collaborative, as in most trust situations, individuals working together will possess different kinds of knowledge. The individuals will therefore engage in interactions that will allow them to pool the various resources to accomplish the task. Since knowledge is shared by the participants, communicative practices that exploit this shared knowledge can be used, like having a shared information structure like a speed bug in a cockpit (Hutchins, 1995). Also, the distributed access of information in the system results in the coordination of expectations, and this becomes the basis of coordinated action.

This stance of an extended mind and the focus on *both* internal and external representational structures makes distributed cognition an ideal framework to explore the trust problem. This is because the environment involved in an inter-agent trust decision is one of representations, and more than one agent and (possibly) artifacts are involved. Also, as Khare and Rifkin (1997) point out, any trust decision involving artificial agents bottoms out to a trust decision involving humans. So the problem-space of trust is a distributed socio-technical system, consisting of people and artifacts (agents).

However, the analysis here will not follow the traditional distributed cognition methodology, which describes, through direct observation, how human agents create and interact with external structure and artifacts. The analysis here is more prescriptive, and will consider the role of the environment in different Artificial Intelligence frameworks and suggest that one of them, where external structures are actively created for artificial

[8] For details see Chandrasekharan and Esfandiari, (2000). 'Representation' is used here in the minimal sense, as something standing in for something else.

agents, is the best one to pursue to solve the trust problem. The creation of structure in the environment, or adapting the environment to the agent, has been explored within Distributed Cognition by Kirsh (1996), and to some extent Hutchins (1995). Kirsh's analysis considers how animals change their environment to make tasks easier. He identifies two kinds of structure animals create in the environment, physical and informational. An example of physical structure is the use of tools by animals, like the Caledonian crows using twigs to probe out insects from the ground. The crows even redesign their tools, by making probes out of twigs bitten from living trees, and they fashion at least two different set of probes, one hook-shaped and the other pointed. An example of informational structure would be people reorganizing their cards in a game of gin rummy. In this case, the player is using the cards to encode his plans externally. The cards "tell" the player what he needs to do, he doesn't have to remember it. The gin rummy algorithm is distributed across the player and the card set. The action of sorting the card set reorganizes the environment for "mental rather than physical savings". Kirsh (1994) terms these kind of actions "epistemic actions" as different from "pragmatic actions". Epistemic action changes knowledge states, pragmatic action changes the state of the world. According to Kirsh, the second kind of structures created in the environment, informational structure, furthers "cognitive congeniality" and is usually created only by higher animals.

We disagree with the second half of Kirsh's claim. We consider signaling, a very important aspect of animal life (cutting across biological niches) as an instance of changing the informational structure of the environment to further "cognitive

congeniality". A simple thought experiment illustrates this. Consider the peacock's tail, the paradigmatic instance of a signal. The tail's function is to allow female peacocks (peahens) to make a mating judgment, by selecting the most-healthy male. The tail reliably describes the inner state of the peacock, that it is healthy (and therefore has good genes). The signal is reliable because it pays only a peacock with enough resources to produce a flamboyant tail. If you are a sickly male, you cannot spend resources to produce ornaments. The health of the peacock is directly encoded in the tail; the peacock carries its internal attributes on its tail, so to speak.

To see the cognitive efficiency of this mechanism, imagine the peahen having to make a decision without the existence of such a direct and reliable signal. The peahen will need to have a knowledgebase of how the internal state, of health, can be inferred from behavioral and other cues. Let's say that "*good dancing*", "*lengthy chase of prey*", "*long flights*" (peacocks fly short distances), "*tough beak*" and "*good claws*" are cues for the health of a peacock. To arrive at a decision using these cues, first the peahen will need to "know" these cues, and that some combinations of them implies that the male is healthy.

Armed with this knowledge, the female has to sample males for an extended period of time, and go through a lengthy sorting process based on the cues (rank each male on each of these cues: good, bad, okay). Then it has to compare the different results, keeping all of them in memory, to arrive at an optimal mating decision. This is a computationally intensive process. The tail allows the female peacock to shortcut all this computation, and go directly to the most-healthy male in a lot. Reliable self-description, like the peacock's

tail, is one of nature's ways of avoiding long-winded sorting and inference. The self-description allows the peahen to have a single, chunked, cue, which it can compare with other similar ones to arrive at a decision. A signal provides a standardized way of arriving at a decision, with the least amount of computation.

However, note that the signal provides cognitive congeniality to the receiver, and not to the sender. The sender, the peacock, gains because he has an interest in being selected for mating. Kirsh's analysis considers how individual organisms change the environment for their own cognitive congeniality, and his claim about higher animals is probably justified in that context, because very few animals create information structures for reducing their own cognitive complexity.

The reduction of others' cognitive complexity using signals is so common that it can be considered one of the building blocks of nature. Signaling exists at all levels of nature, from single celled bacteria to plants, crickets, gazelles and humans. Surprisingly, this basic structure of cognition, where the information structure of the environment is changed to facilitate later iterations of a task, has received very little attention from Artificial Intelligence. Many papers have considered the role of stigmergy, a coordination mechanism where the action of one individual in a colony triggers the next action by others (Susi, 2001). Stigmergy is a form of indirect communication, and has been a favoured mechanism for situated AI because it avoids the creation of explicit representations. Signaling, on the other hand, is closer to being a representation, and therefore more useful in understanding the trust problem. In the following section we

look at how signaling can be incorporated into Agent Design. We do not believe in the strong version of AI, and just consider agent architectures as design methodologies here, after Bryson (2000).

Agent Design

We categorize Agent Design into four frameworks. We illustrate these four frameworks using the problem of giving physically handicapped people access to buildings. There are four general ways of solving this problem.

- **Case I**: The first one involves not incorporating detailed environment structure into the design, and building an all-powerful vehicle, which can fly, climb stairs, detect curbs etc.

- **Case II**: The second one involves studying the environment carefully and using that information to build the vehicle. For instance, the vehicle will take into account the existence of curbs, small stairs and elevators, so it will have the capacity to raise itself to the curb, a couple of stairs, or into an elevator.

- **Case III**: The third one involves changing the environment. For instance, building ramps and special doors so that a simple vehicle can have maximum access. This is the most elegant solution, and the most widely used one.

- **Case IV**: The fourth one is similar to the first one, but here the environment is all-powerful instead of the vehicle. The environment becomes "smart", and the building detects all physically handicapped people, and glides a ramp down to them, or lifts them up etc.

The first approach is similar to the traditional AI one, which ignores the structure provided by specific environments during design, and tries to load every possible environment on to the agent, as centrally stored representations. The agent starts from these representations and tries to map the world on to this structure.

The second approach is similar to the Brooks' one, which recognizes the role of the environment, and exploits structure existing in the environment while building the agent. Notice that the environment is not changed here. This is a passive design approach, where the environment is considered a given.

In the third approach, the designer actively intervenes in the environment and gives structure to it, so that the agent can function better. This is Active Design, or agent-environment co-design. The idea is to split the intelligence load — part to the agent, part to the world. This is agent design guided by the principle of Distributed Cognition, where part of the computation is hived off to the world. Kirsh terms this kind of "using the world to compute" Active Redesign. This design principle underlies many techniques to minimize complexity. A good example at Kirsh's information level (the cognitive congeniality level) is bar coding. Without bar coding, the machine in the mall would have

to resort to a phenomenal amount of querying to identify a product. With bar coding, it becomes a simple affair. We consider the same principle to be at work in the Semantic Web enterprise. The effort is to change the world (the Web) so that software agents can function effectively in it. The Active Design principle can also be seen to be at work in the Auto-ID effort, where products are provided with Radio-frequency Identification (RFID) tags, which can be detected by RFID readers. Such tagged objects can be easily recognized by agents fitted with RFID readers, like robots in a recycling plant. At Kirsh's physical level, the Active Design principle can be found in the building of roads for wheeled vehicles. Without roads, the vehicles will have a hard time, or all vehicles will need to have tank wheels. With roads the movement is a lot easier for average vehicles.

The Active Design approach is at work at the social level as well, especially in instances involving Trust. We actively create structure in the environment to help people make trust decisions. Formal structure crated for trust includes credit ratings, identities, uniforms, badges, degrees, etc. These structures serve as reliable signals for people to make trust decisions. Less reliable, and more informal, structure we create includes the way we dress, the way we talk etc.

The fourth approach is the ubiquitous/pervasive computing idea. This is an extreme version of the active design approach. Real design can be seen as a combination of two or more of these approaches. The following illustration captures the four approaches.

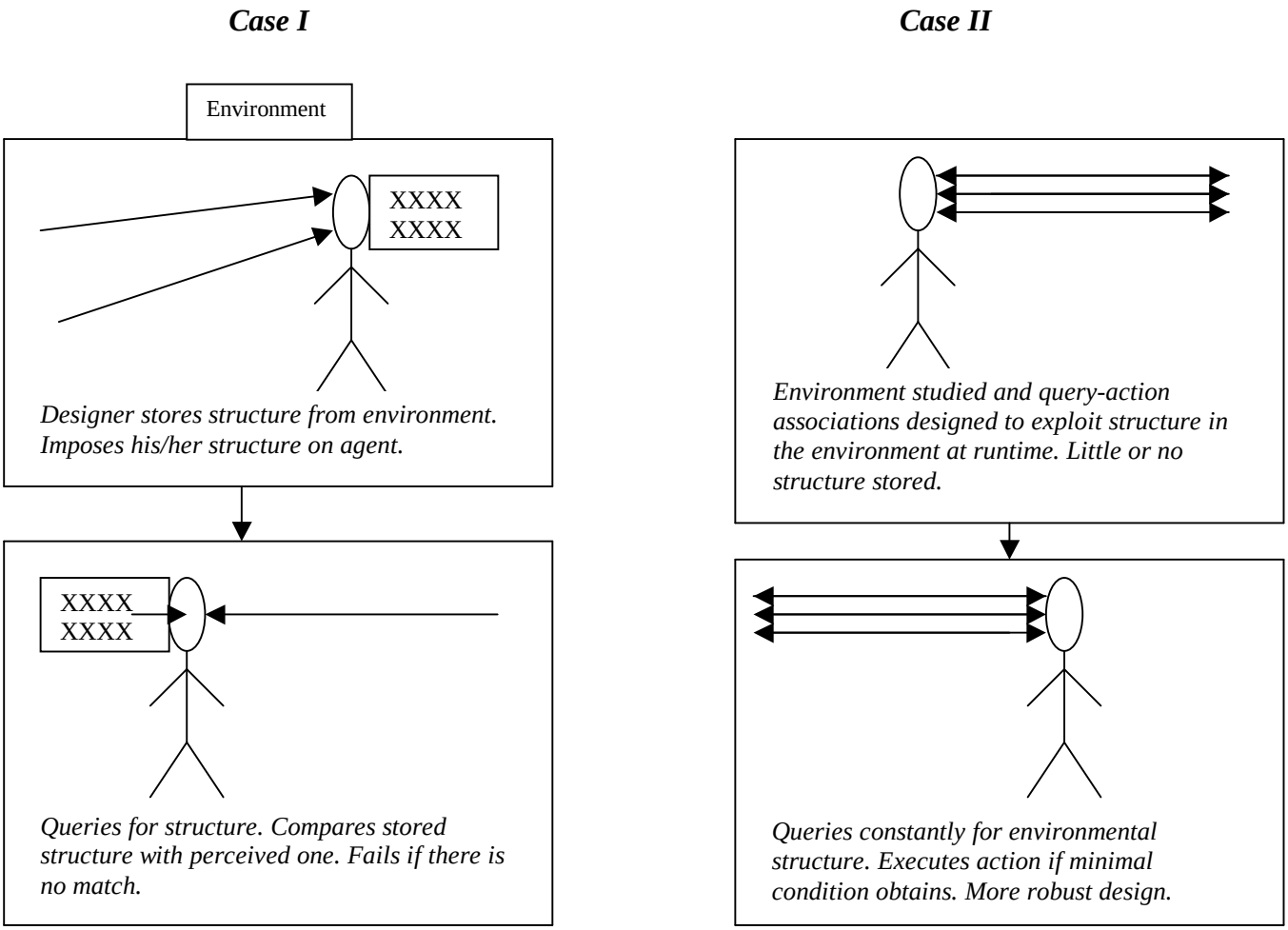


Figure 3

The four approaches to agent design are illustrated above. In the first two cases the environment is considered as a given, and the designer makes no changes to the environment. This is passive design. In the third case, the designer actively intervenes in the environment and gives structure to it, so that the agent can function better in it. The agent only queries for the structure provided by the designer. This is active design, or agent -environment co-design, where the knowledge is split equally between the agent and the environment. The agent and the environment evolve together. In the fourth case, it is the environment that is designed, and the agent is assumed to have minimal capabilities.

As illustrated by the examples, the third approach is the most elegant one — change the world, redesign it, so that a minimally complex agent can work effectively in that world. We will apply this design approach to the trust problem in agent systems.

4. Trust and Representation

In a trusting situation, the trust decision is arrived at through queries, directed at a social environment, an environment made of representations and other agents. By analogy with Brooks' robots — which navigate by "bumping" into objects — agents in a social environment navigate by "bumping" into representations. Instead of sensing the environment by bouncing sound waves off objects (using sonar), these agents bounce representations (mostly linguistic structures) off each other. This kind of sensing is at a level above the kind of sensing done by Brooks' robots, because there is an added level of semantic interpretation involved in this kind of sensing.

We believe that this representational querying process in fundamental to the understanding of the trust problem, because it is a built-in condition of a trusting situation that it *demands* representational querying. To understand the relation between trust and representation, we break the notion of trust differently from the categories of trust laid out in the beginning of the paper. Minimally, trust can be broken up into four categories, depending on the types of entities it is directed to. The entities are: objects, patterns/processes, animals and people.

Objects: Consider an iron block. You trust the block not to move unless an unbalanced force acts on it. The block is inanimate, and has a purely deterministic behaviour pattern, which you know. Hence your trust in the behaviour of the block is complete. Notice that this trust is based on experience of the behaviour of the block (or similar objects) you have encountered before. It is a straightforward inductive relation you have established. Notice also that the laws of physics are not what the trust decision is based on, it is based on an inductive relation established through experience. A villager sitting on an iron block in rural Africa is using this inductive relation, not the laws of physics.

Patterns/Processes: Now consider the sailor's knot, a mechanism some people trust their life with. For them, the trust in the knot, a pattern, is almost complete. But the trust can be affected by the material used to create the pattern. For instance, it would be better if the knot is made using a nylon rope than an ordinary rope. Therefore, you do not trust a knot just from the behaviour of knots you have encountered before. The trust depends on at least one more factor, namely the material that instantiates the pattern. The pattern is at a level of abstraction one order removed from the object. Correspondingly, the trust in the behaviour of the pattern is also lower than trust in the behaviour of the object. The inductive relation here is not as straightforward as in the object case.

Animals: Now consider a strange dog that wags its tail at you. You are almost sure that it will not bite. Why? Because you know that in most animals, a direct correlation exists between an internal state and its external expression[9]. That is, you can infer, with a high degree of accuracy, the dog's internal state (friendliness) from the external expression of

it (wagging tail). The dog will not bite when its tail is wagging. This correlation is one of the factors that make animal signaling reliable. However, there is still a certain amount of uncertainty in trusting the dog. It is not as trusted as the block or the knot. The dog is autonomous, and it has a variable (or indeterminate) internal state, which can only be *inferred* from its external expression. Notice that the inductive relation has been complicated significantly here. We will explore this point in detail below.

People: Now consider a strange human who smiles at you. You probably will not trust the person as much as you trust the dog. Because human beings, unlike most animals, have the following unique capacity: we can have one internal state, and express a very different one. In other words, the internal state and its representation need not be directly correlated. Unlike the case of the dog, we have a gap between our representations and our internal states. We call this the representational gap[10].

Now, given this observation, we would like to make the following claim*: inter-agent trust involves making sure that there is no representational gap.* That is, making sure that the external expressions of people correlate with their (postulated/suggested) internal states. Since this is not something that can be done with hundred percent accuracy, there is an element of risk in every trusting decision.

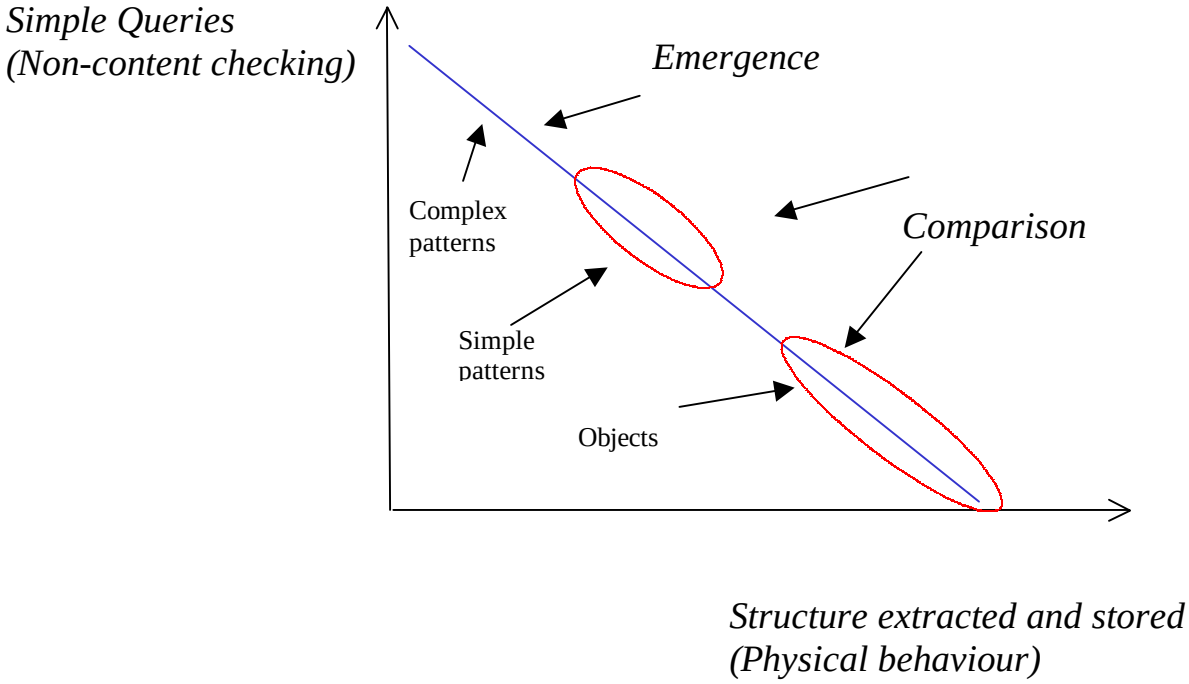
[9] Deception is widespread among animals and plants, but the majority of animal behavior is honest.
[10] The external expression can span two domains, language and action. Thus, we can have one internal state, a different linguistic expression of it, and a totally different action. Or two of them could match. Sometimes all of them match.
.

Notice that two things are happening in the evolution of trust from objects to people. One, there is a gradual loss of inductive power as we move from cases involving objects to cases involving people. Two, you move a level higher from direct observation, to observation of representations.

You can have almost absolute trust on objects based on previous experiences with them. It is almost the same with patterns, with a condition added, namely the pattern is instantiated by a material, or a similar one, encountered before. With reference to the graph presented earlier, experiences with objects allow us to categorise such experiences into the comparison level of our graph (see Figure 4 below).

Experiences with patterns are a notch higher, because more queries are needed, about the nature of the material that instantiates the pattern. However, in the case of animals and humans, which are autonomous agents, the querying becomes categorically different. In the case of animals, we have to make inferences about how an external expression correlates with the internal state of the animal. That is, we cannot just expect the dog not to bite because the dog has not bit before. We have to look for an indication, a representation of the dog's behaviour, namely the wagging of the dog's tail. Notice the second thing that has happened. We have moved one level upward from the pattern: the induction is not about the dog's past behaviour, but about the *correlation of a representation* with its past behaviour.

The agent-environment relationship for trust (the object-pattern case)



The agent-environment relationship for trust (the animal-human case)

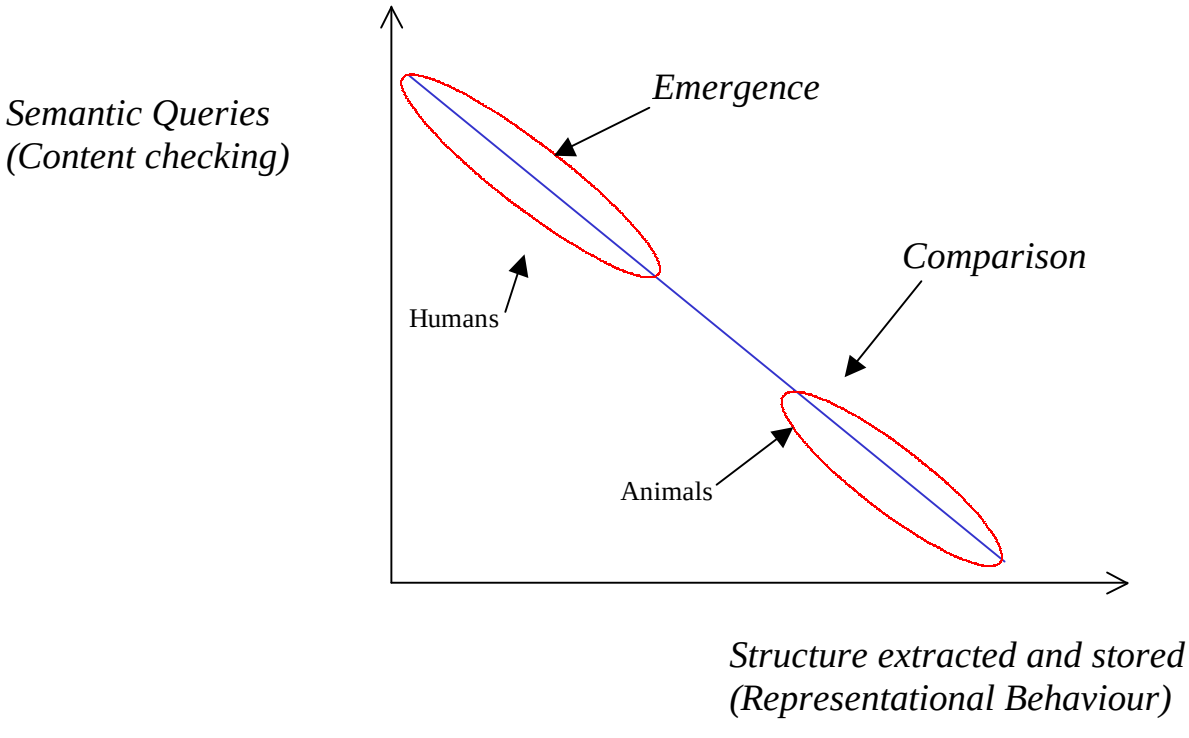


Figure 4

In other words, the experience we base our trust decision on is a second order one. The querying we do to arrive at a trust decision in the animal case is very different from the object case; it is a semantic query, a query that matches representations. We need two different graphs to represent the object-pattern case and the animal-human case (see figure below).

In the case of people, the inductive relation loses its power almost totally. Unlike animals, there is no *necessary* correlation between people's representations and their behaviour. The dog's language is a protocol. The dog cannot but do what it indicates. Human beings are not like that. Our language is ad-hoc, and does not necessarily indicate our internal states. This ad-hoc nature is one of the reasons why natural language is extremely powerful as a representational structure. But it gains this expressive power by sacrificing on reliability.

Since people have the capacity to de-couple their internal state from their external expressions, the representations used by a person may not indicate the person's internal state. This situation, thus, falls into the category of little stored internal structure in the graph. So, the agent has to resort to a significant amount of queries.

However, if our language and behaviour were like the dog's, with a reliable structure, a direct one-to-one correspondence between internal state and external expression, we do not have to run elaborate inference procedures to trust people. The dog language and behaviour is like a protocol, where the expressions have specific and particular meanings.

For systems that work using protocols, there is no need for extensive querying or trust calculation. All we need to do is compare *representational* patterns, like the peacock mentioned earlier. For trust situations involving agents with such a link between internal structure and external representation, the calculation of the trust metric is essentially a process of verifying the extent of the link between an agent's external expressions and her internal state, using queries.

Notice that the queries themselves have to be representational, because what is being checked is the validity of a representation[11]. Thus, communication (representational querying) is a central mechanism for arriving at trust decisions. In summary, *arriving at a trusting decision is the ruling out of the representational gap using representational querying.*

In the next section, we apply this result to the trust problem in agents and suggest a solution.

5. Creating Environment Structure for Trust

To exploit the Active Design strategy, the solution to the trust problem in multi-agent systems needs a transfer of part of the trust-decision process to the environment, which is largely constituted by the trustee, the agent that is trusted. So the designer has to actively intervene in the trust environment and provide structure for the trustor, the trusting agent,

[11] This is perhaps a bit controversial. For instance, we could verify the validity of an agent's representation by the number of times his actions and representations correlated previously. However, we consider this verification as involving representational querying, because this assessment is different from a sonar input. No sonar input can inform us about the extent to which someone keeps his word.

to exploit. Since we have identified the trust-decision as one of bridging the representational gap, the new environment structure should work in bridging the representational gap. With reference to the graph charted in Figure 4, this involves creating exploitable semantic structures in the environment for the trustor agent. These structures should allow the trusting agent to bridge the representational gap[12].

As we observed before, such created informational structures in the environment, which give a "leg-up" for agents while making decisions, is found commonly in nature, in animal signaling. In his seminal book on animal signaling, *The Handicap Principle*, Zahavi (1997) argues that the only way such information structures (signals) can be reliable is by the sender agent paying a cost for "announcing" the signal. Essentially, only costly signals are reliable, "cheap" signals are not reliable. The paradigmatic example of a costly signal is again the peacock's tail. The tail is a signal of genetic quality of the peacock, and peahens take mating decisions based on the quality of a male's tail. The tail is a reliable signal of genetic quality because it imposes a handicap on the peacock – the tail is costly to produce in terms of energy, and it is a hindrance while escaping from predators. This means the peacock that has a good tail is healthy enough to produce such a tail and escape with it from predators. Possible deceivers – unhealthy peacocks – cannot produce a well-formed tail and cannot escape with it, which means a good tail advertises genetic quality very reliably. The cost of carrying the tail ensures its reliability as a signal for peahens' mate choice decisions.

[12] Note that we are not ruling out the trusting agent's contribution to solving the problem. As pointed out earlier, one of the ways the trusting agent can bridge the representational gap is through extensive querying for verification. Developing a mechanism for such trust-creating-queries would involve focusing on the agent-side of the problem. We focus more on the environment, or the trustee-side, in this paper.

The Handicap Principle essentially states that costly signals are reliable. If a signal is not costly, then it *may not be* reliable, because a cheater can fake the signal easily. But this does not mean that reliability is ensured *only* by the sender incurring a cost. Reliability can also be ensured by passing the cost on to possible cheaters, by making cheating risky. An example (a controversial one) is the black bibs of a species of sparrow, which signals aggressiveness. It is easy to fake, but fakers can be caught easily by picking a fight. Once caught, fakers are severely punished by the receivers. Note that there is a cost involved here as well, but for cheaters. The risk involved in cheating makes the signal too costly to mimic. This combination of verification and punishment ensures reliability of the bib as a signal. Since the purpose of a signal is to allow a receiver to take quick decisions, the sender may decide not to invest in a signal if the receiver has other means of ensuring reliability in real time. However, in general, a costly signal for the sender advertises reliability best.

Most existing techniques for trust decision-making in artificial systems also use this cost principle. One of the common ways suggested to design trustable agents is to have a third party certify that in his/her experience the agent's internal states correlate with its representations. This is the general approach of certification using authentication servers etc. Note that the Handicap Principle is at work here. The certification is a cost incurred by the agent that is trusted, the trustee agent, who takes an effort to invest in the certification. Also note that bridging the representational gap is what the certification does, because certification essentially declares that what the agent says is what the agent does. It is easy to see why the certification has to be done by a trusted third party. The

agent itself saying that its representation correlates with its internal states will not do, because the agent is using a representation to say that, and the doubt is in the authenticity of *just that* representation. The third party authentication route can itself take a number of forms, depending on the task the agent performs. This includes recommendation mechanisms (Abdul-Rahman, 2000), points awarded by servers visited, certification by authentication servers etc.

Another way to decrease the representational gap is to check the competence, and verify the claims of the agent that is trusted, using a sandbox model. This is the most common route in trust issues involving security. However, this is not very useful in the learning and delegation cases of trust, because they both require almost-full transfer of control. A technique that mixes the Handicap Principle with the sand-box model is the method of proof-carrying code, illustrated by Necula and Lee (1998). In this method, an easily-checkable proof is added to the code. The checking of the proof proves that the execution of the code does not violate the safety policy of the receiving system. Notice that the proof is a cost incurred by the trustee agent. Currently proof-carrying code works only in cases of trust-for-safety, for instance in proving that the incoming agent does not violate data-access policies, resource-usage bounds etc. No proofs exist for the delegation and learning cases of trust.

We explore here a close cousin of the proof-carrying code approach as a way of bridging the representational gap in delegation situations — an *institutional sign*-based approach. Institutional signs are structures we create in the environment for agents to make quick

and dirty trust decisions. The role of signs in trust decisions is explored in detail by Bacharach and Gambetta (2000). Their approach begins with game theory, and identify trust as a particular belief, which arises in games with a particular pay-off structure. So the primary problem of trust is the problem the trustor faces in answering the question "can I trust this person to do X?" This problem is identified as a problem of uncertainty about the payoffs of the trustee. A trustee's "all-in" pay-offs is allowed to be different from her raw payoffs. If the trustee goes by raw payoffs, she will maximize her interests, and will not do X. But if she goes by her all-in payoffs, she will do X. Importantly, the trustee's decision to do X, i.e. go by her all-in payoffs, is considered to be induced by "trustworthy-making" qualities, like virtues, internalised norms, character features etc.

However, these qualities, like the health attribute of the peacock, are unobservable. Now, given that the trustors know of these trustworthy-making qualities (what the authors call *t-krypta)* are unobservable, the trustors look for mediating *signs* of these qualities in a trustee. However, since the trustor will be proceeding in this way, and given the pay-off structure, there is a motive for an opportunistic trustee to "mimic" – to emit signs of trustworthy qualities (what the authors call *manifesta*). This creates a secondary problem of trust, the problem of mimicry: the trustor must judge whether apparent signs of trustworthiness are themselves to be trusted. Notice that this is nothing but the representational gap sketched above.

This secondary problem of trust is then analyzed in terms of signaling theory, as it is used in economics, biology and game theory. The main result of signaling theory is that for a

certain class of games (games with three kinds of agents: ones with the krypta, ones without the krypta, and receivers), there is an equilibrium in which at least some truth is transmitted, provided that among the possible signals is one, *S*, which is cheap enough to emit for the signalers who have the krypta (K), but costly enough to emit for those who do not have the krypta. A smile is an instance of what won't count as an *S*, because the cost of emitting a smile is the same for both agents who have K and those who don't have K. As the authors observe, benevolent uncles smile to show that they are benevolently disposed; wicked uncles smile to seem to be benevolently disposed.

However, there is a category of signs that are costly to mimic by trustees who do not possess them, *and all of them involve the identity of the trustee* (emphasis mine). The authors note that the secondary problem of trust (the problem of mimicry in their terms, the problem of the representational gap in our terms) is often soluble by assessing the reliability of identifying marks. This is the case where the trustor can indirectly establish whether the trustee has trustworthy-making qualities, by establishing whether she is the same person, or among a category of persons, who proved to be trustworthy in the past.

These identifying marks could be anything, from folk associations of trust with skin colour, gender, height and physical characteristics (untrustworthy: black people, females, short people, redheads, people from the city, people from the country — this shows that the problem of stereotypes and prejudice is a facet of the trust problem) to the grades you got in college (academic prowess demonstrates high productivity). However, there is a class of identifying marks that are *created* in the environment specifically to solve the

trust problem. These are institutional signs. A good example of an institutional sign is the policeman's uniform. We will trust a person with a gun in the marketplace if he is wearing a policeman's uniform. If he is not wearing the uniform, we will need to look extensively for other cues to decide whether the person is trustable.

The interesting point here is that from the active design perspective sketched in section 3, the uniform, like proofs in code and XML metatags, is just an instance of adding information structure to the environment to help the agent – an instance of the agent-environment co-design approach. The uniform is "created", not computed. The uniform comes out of an active effort to change the environment to solve the trust problem. It allows agents to minimise the queries needed to arrive at a trust decision. All institutional mechanisms for advertising competence are designed to do this. Here the uniform is *created* by an institution, namely the government, and it *guarantees* that persons wearing it are trustable. It is this institutional guarantee that allows us to make a quick decision about trusting the policeman. Notice that the institutional guarantee is based on cost for the sender as well as cost for possible mimics. The policeman has invested effort in getting the uniform, and there is a very high cost associated with misusing the uniform, so the payoff for a deceiver is not enough to mimic the uniform.

How can we apply this model of institutional signs to the trust problem in agents? By creating a costly institutional guarantee like the one provided by the uniform, obviously. But what sort of guarantee, and what sort of institution? If we look back at the evolution of trust from objects to people, the optimum case for an autonomous agent is the animal

one – where the representations are as close to the internal state as possible. In other words, the representations generated by an agent need to be like a protocol, and should act like the peacock's tail, reflecting accurately the internal state of an agent. If we can create and institutionalise a protocol that guarantees that no representational gap exists in an agent's speech acts, we can have a quick and dirty way of assessing an agent's trust value.

One possible way to do this seems to be a programming language that guarantees that an agent's speech acts reflect its internal states reliably. In other words, a programming language that acts as an institution.

6. Identity as Handicap

Before we turn to the details of this programming language, let us briefly return to the trust problems we are trying to address. We classified the trust problem in agents systems into three.

- *Trust for Security*: Agent B asks Agent A for access to parts of A's system. To give access, Agent A needs to know whether Agent B is malevolent or benevolent — that is, will B harm A's system?

- *Trust for Learning*: Agent A needs to learn X from Agent B. For this, A needs to know whether B's beliefs about X is true and justified, and relevant to A's functional role. That is, is B's knowledge of X correct and relevant?

- *Trust for Delegation*: Agent A needs to delegate a task to Agent B. For this, A needs to know whether B has the competence to do the task, and whether B will in fact do the task, given appropriate external conditions.

We have argued that the trust problem in humans has its root in the perceived gap between the representation the agent uses to communicate its internal states, and the agent's actual internal states. Any agent design for building trustable agents essentially is aiming to do just one thing: decrease the representational gap[13]. The actual mechanisms of implementing this design will vary for each of the above three problems. We will take the case of trust in delegation to argue our case. The results can be applied partly to the other cases, with some variations.

Remember that what we are trying to create is an institutional structure, a structure that guarantees that the representations generated by an agent are like a protocol, and act like the peacock's tail, reflecting accurately the internal state of an agent. In agent systems, this means creating Agent communication Languages (ACLs) that reflect accurately the internal state of the agent. For this, we will need to combine the ACL with the programming language used to build the agent. Because *only* this will allow us to bring the content of the agent's communication as close to the agent's internal state as possible.

[13]This is true of interface agents as well. A human will trust an interface agent only if the agent provides the user with ways to reduce the representational gap – either by induction through extensive interaction, or through secondary sources like experts and reviews, or by being able to know, and change, the internal

An objection could be raised here: this approach will make the agent system more like an object-oriented system. And this is against the spirit of the agent paradigm, because the paradigm seeks to allow any agent, created using any language, to work with each other. Thus, in a sense, the entity-independence of the ACL is a central tenet of the agent paradigm. Linking an ACL to a programming language would make it a non-agent system.

We think this need not be so. Consider an imaginary language, TrustR, which allows you to create agents with specified roles. That is, the language allows you to create buyer agents, seller agents, ontology mapping agents (agents that can map a domain, from say the Cyc ontology to the Ontolingua one) etc. These agents have predefined functionalities or competencies. That is, the buyer agent has the competence to buy, the seller agent has the competence to sell, etc. The agents can advertise their roles using a performative called *claim*. A *claim* is a new performative[14], but with the added condition that it is binding on the agent (we detail the mechanism for binding later).

Now, suppose TrustR allows the agents to use any ACL to communicate with other agents, but specifies that *the content of the claims should always accurately reflect the functionality of the agent and nothing else*. In other words, the syntax of the claim can be any ACL, but the semantics should reflect the agent's functionality, which is specified by the programming language. For instance, if we create a priest agent using TrustR, the

---

state of the agent. This relates to Schneiderman's (1997) arguments about trusting an interface agent. We consider the trust problem in the case of the interface agent a special case of the delegation problem.

[14]This assumes the inclusion of a new performative "daim", and the message parameter 'role", into the ACL.

agent can only claim that it is a priest, and claim only the competence of a priest, and not the competence of, say, a buyer. However, the agent can use any ACL to communicate this competence to another agent. The ACL is still entity-independent, but the content of the ACL is entity-dependent.

We think such a programming language, a virtual institution of sorts that create reliable roles for agents, would solve some of the trust problems involved in delegation. In particular, it will solve the representational gap problem involved in assessing the competence of another agent. The trade off is that we can work only with roles specified by the language. However, we think this is a good place to start, and different roles can be developed depending on user requirements.

There is still a problem remaining, though. How can we ensure that the claim made by an agent is reliable? That is, how can we make sure that the agent will do only what it claims it will do, and if not, how can we take action against the agent? For instance, we can have a situation were someone builds an agent that does not fulfill the tasks assigned to it, say selling. The agent can take your money and not deliver the promised good. We believe this problem can be solved using the Handicap Principle as well. For the claims to be reliable, the claims should be costly for the person making the claim. To build this cost into the language, we need to establish an *identity* for the agent.

We draw on Khare and Rifkin (1997) in this conclusion: "Trust is a faith placed in humans, even though sometimes this faith may be manifested in such a way as to appear

that we are only trusting some device." The authors also observe that "(only) people have persistent identities as names that are meaningful in the real world", which is why "you can sue only people, not computers".

To make the claim made by an agent binding and therefore reliable, the agent should have an identity. Since only humans can have meaningful identities, this means all agents should be traceable back to the programmer who created the agent[15]. This can only be done using a mechanism that links the copy of the compiler used to create the agent to a machine, and thereby to the machine's owner[16]. At the very least, if the machine's owner creates an incompetent or a malevolent agent, this would result in the blacklisting and blocking of all agents created by that particular machine. This mechanism (which requires the incorporation of the message parameter "compilerID" in the ACL) will give us a direct link to the person behind an agent, bringing in a lot more accountability to agent systems. For the reliable programmers, like the healthy peacocks, this provides an opportunity to advertise their quality to interested receivers. The compiler ID is a handicap the programmer takes on for advertising the quality of his code. Note also that this is in line with trust in the human situation, where the honest trustee stands to gain by being trusted. This "trust-through-identity" principle is used by some on-line auction sites as well.

[15]This proposal has generated some amount of consternation, but we think the suggestion is fair. Professionals like architects and doctors have identities and can be held accountable for their work. They are also held in high esteem for their good work because they have identities. We think programming as a profession should move to the same model as well. Professionals should be able to advertise their quality reliably.

[16]A similar mechanism, a certifying compiler for Java, has been developed by Necula. See http://www.cs.berkeley.edu/~necula/touchstone.html

Now, there could be another question raised: if the language functions like we suggest, allowing only existing competences to be advertised, where is the trust evaluation? We believe the language-as-handicap guarantees only part of the trust puzzle, because the roles and competences of an agent can be specified differently. The agents still need to communicate, and use verification mechanisms, to arrive at a trust decision. To see why the language is only part of the puzzle, consider the roles as uniforms used by humans. The uniform is an institutional mechanism *designed* by humans to speed up trust decisions. So it is similar to the roles we suggest. However, the fact that someone wears a policeman's uniform does not automatically make him capable of everything a typical policeman is capable of doing. But in certain contexts, the uniform is pretty much all you need to make a trust decision.

Roles, Competency-mapping and Ontologies

In this section, we describe briefly the outline of the modules that make up our proposed trust language, which is an application of the agent-environment co-design model. This is work in progress. We consider roles as "internalised" competencies in specified domains (ontologies). When the agent carries part of the ontology within, or can access an ontology, the ontology becomes part of the agent's role. In other words, the role of an agent can be "unpacked" into competencies and domains.

The roles of the agent are created using a technique called competency-mapping, used widely by human resource managers. Every task is mapped on to a set of competences,

and the competences are defined for ontologies (domains in the case of humans). So, in the human case, if the task is web design, the competences would be web programming and design, and the domain would be the Web and its technologies. Applied to agents, if the task is mapping terms from one ontology to another, the competence would be mapping, and the domains would be Ontology1 and Ontology2. The role of an agent, thus, is a high-level description of what the agent can do. If the role of the agent is "buyer", the agent has the competence of buying, in a specified domain.

We briefly sketch below the role of an agent and how it is created. Consider an agent that can buy phones. It needs to have the competence to buy, and access to the phone ontology. For this agent to be trustable, according to our model, two things should occur: one, the production language should not allow the agent to express anything other than the buying role and the phone domain using an ACL. Two, if the agent somehow misrepresents its competency, the agent should be traceable to a specific compiler in a specific machine.

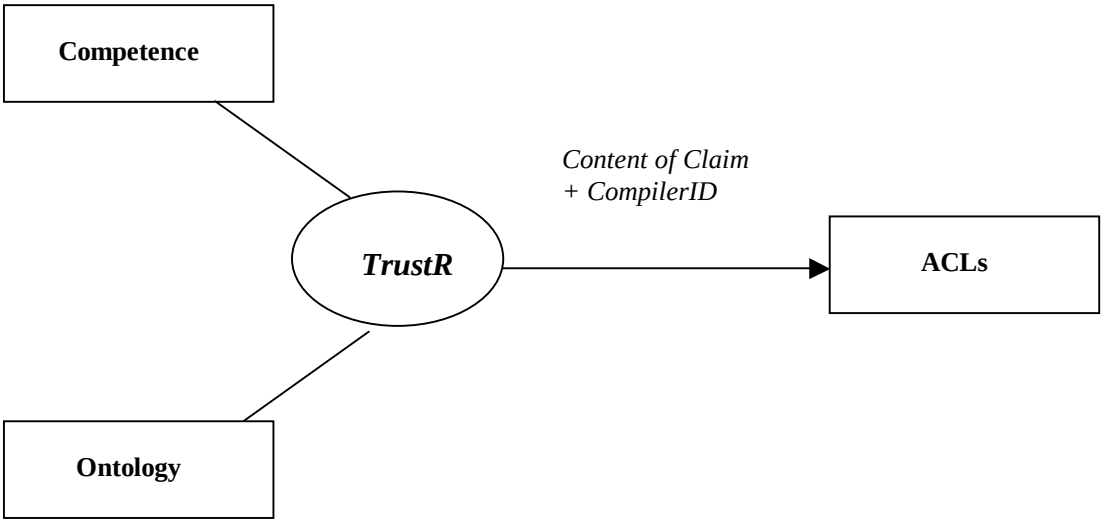


Figure 5

For these objectives to be achieved, the competence (buying), the domain ontology (phones), and the ACL message need to brought together by the compiler. We sketch this process below.

Depending on the ACL selected by the programmer, the compiler allows the agent to make the competence claim in the syntax of that ACL. For instance, if the ACL is KQML, the claim would be:

(claim
:receiver agent b
:sender agent a
:role buyer
:language TrustR
:compilerID 1234
:ontology phone)

The language part, besides specifying the production language, provides a trust value as well. The compiler ID allows the binding of the performative to a compiler and machine.

Summary, Limitations and Future Work

We have tried to do two things in this paper: one, develop a Distributed Cognition approach to the trust problem. Two, in parallel, develop a design methodology to building agents based on Distributed Cognition — the agent-environment co-design model. The approach advocates the designer actively hiving off part of the cognitive load to the environment, and making the environment do some work for the agent. We have applied our design approach to the trust problem, and have suggested that a programming language can act as an institution — by providing the agent that is trusted (which is part of the environment in the trusting decision) with an ID and a pre-defined set of competencies. The ID and the set of competencies function like an institutional uniform (say a policeman's uniform) used by human agents, and helps the trusting agent to arrive at its decision faster.

Our approach is not fool-proof, and has quite a few problems associated with it, the major one being the restrictions the language places on the roles of agents. For instance, the notion of an agent being a "free-agent", an agent that can do any task that comes up, is sacrificed. But we think this problem can be solved to an extent by allowing agents to have multiple roles. The claims made by the roles would depend on the roles the agent intends to take, and the claims would be binding.

Another problem is that our approach is not easily applicable to learning agents. This is because learning involves change of competence, and hence, a change of roles. Thus, learning would result in the agent having a different competence to communicate. Therefore, the production language will have to allow for changes to be made to the claims. Changes in claims will weaken the guarantee provided by the language and bring up the trust problem again.

One solution to this problem is to have fixed skills (like mapping). Learning will then involve changing the domain, the ontologies. However, this approach fails if the learning involves adding changes to the ontologies. A way out of this is to distribute an ontology across agents. We are trying to develop a hierarchy of agents, so that each agent is "responsible" for a layer of the ontology, and is allowed to update only that layer of the ontology. This will allow us to localize and minimize adverse effects of ontology updation through learning.

Our approach only partially tackles the original trust problem in learning: how can an agent trust another agent for learning something? The language-based approach gives a foothold here, because the agent can at least know, or verify, the validity of the information presented. But the relevance of the information to the agent's functional role still has to be computed. This is an area of future work.

Acknowledgements

I would like to acknowledge the help of Dr. Stephen Marsh, Dr. Babak Esfandiari and Dr. Andrew Brook in developing the ideas presented in this paper.

References

ABDUL-RAHMAN A. & HAILES S. (2000). Supporting Trust in Virtual Communities. *Proceedings of the Hawai International Conference on System Sciences*, Maui, Hawaii.

AGRE, P. and HORSWILL, I.(1997) "Lifeworld Analysis." Journal of Artificial Intelligence Research, 6, 111-145

AXELROD R. (1994). *The evolution of cooperation.* New York: Basic Books.

BACHARACH M., & GAMBETTA D. (2001). Trust in Signs. In KAREN COOK, Ed. *Social Structure and Trust.* New York, NY: Russell Sage Foundation.

BRADBURY, J.W. AND VEHRENCAMP, S.L. (1998). *Principles of Animal Communication.* Sunderland, Mass: Sinauer Associates.

BRADSHAW J. Ed. (1997). *Software Agents.* Menlo Park, CA: AAAI Press.

BROOKS R. (1997). Intelligence without representation. In HAUGELAND, J., Ed. *Mind Design II.* Cambridge, Mass: MIT Press.

BRYSON, J. (2000). Cross-paradigm analysis of autonomous agent architecture, *Journal of Experimental and Theoretical Artificial Intelligence*, 12 (2), 165 – 189.

CASTELFRANCHI C. & FALCONE R. (1999). The Dynamics of Trust: from beliefs to action. In the *Proceedings of the Autonomous Agents workshop on deception, fraud and trust in agent societies.* Seattle.

CHANDRASEKHARAN S. & ESFANDIARI, B. (2000). Software Agents and Situatedness: Being Where?. In the *Proceedings of the Eleventh Mid-west conference on Artificial Intelligence and Cognitive Science.* Menlo Park, CA: AAAI Press.

CLARK A. (1997). *Being There: putting brain, body, and world together again,* Cambridge, Mass.: MIT Press.

DASGUPTA, P., (1990). Trust as a Commodity. In GAMBETTA, D., Ed. *Trust: Making and Making and Breaking Cooperative Relations.* Oxford: Basil Blackwell.

DUNN J. (1990). Trust and Political Agency. In GAMBETTA, D., Ed., *Trust: Making and Making and Breaking Cooperative Relations.* Oxford: Basil Blackwell.

ESFANDIARI, B., CHANDRASEKHARAN, S. (2001) "On how agents make friends: mechanisms for trust acquisition," 4th Workshop on Deception, Fraud and Trust in Agent Societies, Montreal, Canada.

GAMBETTA D. (1990). Can we trust Trust? In GAMBETTA, D., Ed., *Trust: Making and Making and Breaking Cooperative Relations.* Oxford: Basil Blackwell.

GANZORALI A. et al (1999). The social and institutional context of trust in electronic commerce. In the *Proceedings of the Autonomous Agents workshop on deception, fraud and trust in agent societies.* Seattle.

GERSHENFELD, N. (1999). *When things start to think*. New York: Henry Holt and Company.

GRIFFITHS, N. AND LUCK, M. (1999). Cooperative Plan Selection Through Trust. In GARIJO, F. J., AND BOMAN, M., Eds. *Multi-Agent System Engineering - Proceedings of the Ninth European Workshop on Modelling Autonomous Agents in a Multi-Agent World, Lecture Notes in Artificial Intelligence,* 1647, pp. 162-174, Springer-Verlag.

HERTZBERG, L. (1988). On the attitude of trust. *Inquiry*, 31 (3), 307-322.

HUTCHINS, E. (1995). How a cockpit remembers its speeds. *Cognitive Science,* 19, 265-288.

HOLLAN, J.D., HUTCHINS, E.L., KIRSH, D. (forthcoming). Distributed cognition: A new theoretical foundation for human-computer interaction research. *ACM Transactions on Human-Computer Interaction,* in press.

JONES J., AND FIROZABADI, B.S. (1999). On the Characterisation of a trusting agent – aspects of a formal approach. In the *Proceedings of the Formal models for Electronic Commerce (FMEC) workshop*.

KHARE, R., RIFKIN A. (1997). Weaving a Web of Trust. *World Wide Web Journal*, 2(3), 77-112.

KIRSH, D. (1990). When is information explicitly represented? In P. HANSON, Ed. *Information, language, and cognition*. (Volume I of The Vancouver Studies in Cognitive Science, 340-365) Vancouver, BC: University of British Columbia Press.

KIRSH, D., & MAGLIO, P. (1994). On distinguishing epistemic from pragmatic action. *Cognitive Science,* 18, 513-549.

KIRSH, D. (1995). The Intelligent Use of Space. *Artificial Intelligence,* 73, 31-68

KIRSH, D. (1996). Adapting the Environment Instead of Oneself. *Adaptive Behavior*, 4 (3/4), 415-452.

KIRSH, D. (1999). Distributed Cognition, Coordination and Environment Design. *Proceedings of the European conference on Cognitive Science.*

KIRSH, D. (2001). The Context of Work, *Human Computer Interaction*, (forthcoming).

KRAUSS, M. R., AND FUSSEL R. S. (1991). Constructing Shared Communicative Environments. In RESNICK. B. L. et al, Ed. *Perspectives on Socially Shared Cognition.* Washington DC: American Psychological Association.

LAGENSPETZ O. (1992). Legitimacy and Trust. *Philosophical Investigations*, 15:1

LUHMANN, N. (1979). *Trust and Power.* Chichester: Wiley.

LUHMANN, N. (1990). Familiarity, confidence, trust: problems and alternatives. In GAMBETTA, D., Ed., *Trust: Making and Making and Breaking Cooperative Relations.* Oxford: Basil Blackwell.

MARSH, S. (1994). Formalising Trust as a computational concept, Ph.D. Thesis, Department of Computing science and Mathematics, Stirling: University of Stirling.

MISZTAL, B. (1996). *Trust in Modern Societies.* Cambridge, Mass.: Polity Press.

NARDI, B.A. (1996b). Studying context: a comparison of activity theory, situated action models, and distributed cognition. In B.A. NARDI, Ed. *Context and consciousness. Activity theory and human-computer interaction.* (pp. 69-102). Cambridge, Mass.: MIT press.

NECULA, G.C. and LEE, P. (1998). Safe untrusted agents using proof-carrying code, in VIGNA, G., Ed. *Mobile agents and security*, LNCS 1419, pp.61-91, Berlin: Springer-Verlag

REED, E. S. (1996). *Encountering the World: Toward an Ecological Psychology*. New York: Oxford University Press.

RIELY, J. AND HENNESSY M. (1998). Trust and Partial typing in open systems of Mobile agents, Internal Report, School of Cognitive and Computing Sciences, University of Sussex.

SCHNEIDERMAN B. (1997). Direct manipulation versus Agents: paths to Predictable, Controllable, and Comprehensible Interfaces. In J. BRADSHAW, Ed. *Software Agents*, Menlo Park, CA: AAAI Press.

SOWA J. F. (2000). *Knowledge Representation: logical, philosophical and computational foundations.* Pacific Grove, CA: Brooks/Cole.

SUSI, TARJA & ZIEMKE, TOM (2001). Social Cognition, Artefacts, and Stigmergy: A Comparative Analysis of Theoretical Frameworks for the Understanding of Artefact-mediated Collaborative Activity. *Cognitive Systems Research*, 2(4), 273-290.

VAN GELDER, T. (1997). Dynamics and Cognition. In HAUGELAND, J., Ed. *Mind Design II*. Cambridge, Mass.: MIT Press.

WEISS G. (1999). *Multiagent Systems: A modern approach to Distributed Artificial intelligence*, Cambridge, Mass.: MIT Press.

WOOLDRIDGE M.J., JENNINGS, N. R. (1994). Agent Theories, Architectures and Languages: A survey. In M.J. WOOLDRIDGE AND N.R. JENNINGS, Ed. *Intelligent Agents: ECAI-94 Workshop on Agent theories, Architectures and Languages*, pp. 1-39, Berlin: Springer-Verlag.

ZAHAVI, A., & ZAHAVI, A. (1997). *The Handicap Principle: A missing piece of Darwin's puzzle*. Oxford: Oxford University Press