UNIVERSITY OF SOUTHAMPTON

A Trust and Reputation Model for Agent-Based Virtual Organisations

by

Jigar Patel

A thesis submitted in partial fulfillment for the degree of Doctor of Philosophy

in the
Faculty of Engineering and Applied Science
School of Electronics and Computer Science

January 2007

UNIVERSITY OF SOUTHAMPTON

ABSTRACT

FACULTY OF ENGINEERING AND APPLIED SCIENCE
SCHOOL OF ELECTRONICS AND COMPUTER SCIENCE

Doctor of Philosophy

by Jigar Patel

The aim of this research is to develop a model of trust that will endeavour to assure good interactions amongst autonomous software agents in complex, networked environments. In this context, we identify the following as key characteristics. Firstly, such environments are open, meaning that agents are free to enter and exit the system at their will, so an agent cannot be aware of all of its interaction partners. Furthermore, there is a possibility that these interaction partners may be malicious or colluding agents. Secondly, the openness and dynamism of these environments means agents will need to interact with other agents, with which they have had no past experience. Even in this context, an agent must be able to accurately assess the trustworthiness of another. Thirdly, the distributed and heterogeneous nature of these systems influences any model or application developed for such environments. Specifically, this often requires models and applications to be decentralised. Lastly, many of the interactions that occur between agents in such systems are in the context of a virtual organisation (VO). Here VOs are viewed as collections of agents belonging to different organisations, in which each agent has a specific problem solving capability which when combined provides a particular service to meet the requirements of an end user. Now, VOs are social structures, and the presence of certain inter-agent relationships may influence the behaviour of certain members. For this reason it is important to consider not only personal experiences with an individual to determine its behaviour, but to also examine the social relationships that it has with other agents.

Against this background, we have developed TRAVOS (A Trust and Reputation Model for Agent-Based Virtual Organisations) which focuses, in particular, on providing a measure of trust for an agent to place in an interaction partner. This measure of trust is calculated by considering the past experiences between the agent and its interaction partner. In instances when there is no personal experience, the model substitutes past experience with reputation information gathered from other agents in the society or from special reputation broker agents. Reputation is gathered in a way that filters out biased or false opinions. In addition to this, the model is constrained by issues of scalability and decentralisation. Furthermore, by extending TRAVOS we developed a set of mechanisms (TRAVOS-R) related to learning and exploiting the social relationships present in VO-rich environments. More specifically, TRAVOS-R presents a novel approach to learning the type of relationship present between two agents, and uses this knowledge to adjust the opinions obtained from one agent about the other.

The TRAVOS models have been tested empirically and have significantly outperformed other similar models. Moreover, to further evaluate the applicability of our approach a realistic system evaluation was also carried out, which involved applying our models in an industrial application of agent-based VOs.

In undertaking this research, we have shown that trust is a key component of networked systems and that a computational trust model can be used by agents in large, dynamic, uncertain and open environments to account for the uncertainty inherent in their social decision-making processes. More specifically, we have shown that by using personal experience, opinions from others, and knowledge of social relationships, an agent is able to arrive at a more accurate trust value, and, as a consequence, that it can interact in a more effective manner.

Contents

List of Figures

List of Tables

Nomenclature

CHAPTER 3

$\mathcal{A}$	A set of agents
$a_1, a_2, ..., a_n$	Individual agents
$O^t_{a_x,a_y}$	Outcome of interaction between agent a_x and agent a_y, as perceived by a_x at time t.
$O^{t_0:t}_{a_x,a_y}$	A set of outcomes of interactions between agent a_x and agent a_y, as perceived by a_x from time t_0 to time t
$m^t_{a_x,a_y}$	Total number of *successful* interactions between agent a_x and agent a_y, as perceived by a_x at time t
$n^t_{a_x,a_y}$	Total number of *unsuccessful* interactions between agent a_x and agent a_y, as perceived by a_x at time t
$\mathcal{R}^t_{a_x,a_y}$	A tuple $(m^t_{a_x,a_y}, n^t_{a_x,a_y})$ representing the history of interactions between agent a_x and agent a_y, as perceived by a_x, at time t
B_{a_x,a_y}	The behaviour of agent a_x towards agent a_y
τ_{a_x,a_y}	The level of trust agent a_x has in agent a_y
$\tau^d_{a_x,a_y}$	The level of *direct* trust agent a_x has in agent a_y
$\tau^r_{a_x,a_y}$	The *reputation* level of agent a_y as calculated by a_x
$\tau^c_{a_x,a_y}$	The level of *combined* trust agent a_x has in agent a_y
γ_{a_x,a_y}	The level of confidence agent a_x has in τ_{a_x,a_y}
α, β	Parameters for the *Beta* probability density function
$\hat{m}^t_{a_x,a_y}$	Total number of *successful* interactions between agent a_x and agent a_y, as *reported* by agent a_x at time t
$\hat{n}^t_{a_x,a_y}$	Total number of *unsuccessful* interactions between agent a_x and agent a_y, as *reported* by agent a_x at time t
$\hat{\mathcal{R}}^t_{a_x,a_y}$	A tuple $(\hat{m}^t_{a_x,a_y}, \hat{n}^t_{a_x,a_y})$ representing agent a_x's opinion of agent a_y, at time t
$\rho^t_{a_x,a_y}$	The *probability of accuracy* of opinions provided by agent a_y, as perceived by agent a_x, at time t
$\mathcal{H}$	A set containing a *history of opinion provision* from a particular opinion provider
$\boldsymbol{h}$	A subset of $\mathcal{H}$ that contains *similar* opinions

CHAPTER 4

$\mathcal{S}$	A set of service provider agents
$s_1, s_2, ..., s_n$	Individual service provider agents
$\mathcal{C}$	A set of service consumer agents
$c_1, c_2, ..., c_n$	Individual consumer agents
$\mathcal{O}$	A set of opinion provider agents
$o_1, o_2, ..., o_n$	Individual opinion provider agents
o^a	An opinion provider agent that provides *accurate* opinions
o^n	An opinion provider agent that provides *noisy* opinions
o^l	An opinion provider agent that provides *false* opinions

CHAPTER 5

$\mathcal{L}_{type}^{a_x,a_y}$	A *permanent* relationship of a *type* between a_x and a_y
$\hat{\mathcal{L}}_{type}^{a_x,a_y}$	A *transient* relationship of a *type* between a_x and a_y
com	Label for a competitive type of relationship
cop	Label for a cooperative type of relationship
dep	Label for a dependence type of relationship
$\mathcal{P}$	Set containing all observable results $(p_1, p_2, .., p_n)$ of an interaction.
$\mathbf{R}^{a_x,a_y}$	The permanent relationship vector, which represents the beliefs an agent has about the permanent relationship between a_x and a_y
δ^{a_x,a_y}	The confidence an agent has in its beliefs about the permanent relationship between a_x and a_y
$E_{\hat{\mathcal{R}}^t_{a_x,a_y}}$	The level of trust in a_y, as portrayed in the opinion provided by a_x
$\bar{E}_{\hat{\mathcal{R}}^t_{a_x,a_y}}$	The *adjusted* level of trust portrayed by an opinion

CHAPTER 6

n_{ei}	The number of experience interactions between a service provider and the rest of the service providers in the population
n_{oi}	The number of opinion experience interactions betweens agents in the service provider population (that provide opinions) and the consumer agent
n_{si}	The number of signal experience interactions between a consumer agent and the service provider population

Acknowledgements

Having been through the *Ph.D* experience, I can say that contrary to popular belief (and academic requirement) no thesis is ever the product of one person's efforts. This one was no different – *my apologies if I have inadvertently omitted anyone to whom acknowledgement is due*.

First and foremost, I would like to express my heartfelt gratitude to my supervisors Professor Nick Jennings and Professor Michael Luck for their guidance, constant encouragement, and insightful advice (on the research and, on occasions, life generally). Without their Herculean editorial efforts this thesis would be a labyrinth of words for the reader. It is only through their constant review and revision that my written word has become strong and my arguments clear. I give credit to them for the flashes of literary genius you *may* encounter, and accept full responsibility for any omissions, mistakes and errors you *will* encounter.

I would also like to acknowledge the people without whose help and advice this thesis would not have been possible. I start with a hearty thanks to W.T. Luke Teacy – a valued friend, colleague and co-author, who began this journey with me, and has often carried me through the more difficult times, always presenting himself as a perfect and patient board which I can bounce my rough ideas against. Others include: Stephen Marsh for his views on trust and its darker sides; Gopal Ramchurn, Trung Huynh, and Jordi Sabater for their insights into the domain of computational trust in agent-based systems; Professor Victor Lesser for the invaluable discussions that provided encouragement for the research; Professor Carles Sierra, Dr. Nathan Griffiths, and Tomas Klos, all of whom provided me with valuable feedback on my work; and finally, Rajdeep Dash and Paritosh Padhy for reviewing my chapters when I was too afraid to submit them to my supervisors.

I gratefully acknowledge the financial support provided by the CONOISE-G project, which is funded by the DTI and the Welsh E-Science Centre. Additionally, the project has provided me with the context for my research, and a project team consisting of Alun Preece, Stuart Chalmers, Nir Oren, Timothy Norman, Peter Gray, Gareth Shercliff, Patrick Stockreisser, Jianhua Shao, Alex Gray and Simon Thompson – All of whom have patiently listened to my presentations, and given me advice.

During my time as a Ph.D student, I have had the pleasure of working within the IAM Group at the Univeristy of Southampton, a terrific place where the doors are always open. The joy outside of research can be attributed to many individuals, including: Steven Kumarappan, Lakhdip Nagra, Ruple Vaid, Raviraj Shah, Nimet Patel, Rupesh Patel and Shivam Desai - friends who were, and still are, always ready to provide a much needed distraction. In particular, I would like to show gratitude to Pawandeep Panesar, who gave me the strength to tackle the most arduous tasks with a smile, a strength I hope I carry throughout my life.

In closing, I would like to thank my family for their support through the three years. My father showed a keen interest in my work; my mother, who despite my telling her, cooked copious amounts of food to keep me going; and a brother who has grown to be a very close friend.

To my grandparents, for their love, wisdom and encouragement ...

"I have no fear that the result of our experiment will be that men may be trusted to govern themselves without a master."

– Thomas Jefferson,
Third president of the United States (1743 - 1826).

Chapter 1

Introduction

Modern computing has evolved from single computer systems to large scale networked and distributed systems capable of providing on-demand access to a range of computing facilities (for example the Web, the Grid and peer-to-peer networks). The change has been fueled by the need of the users to find and access information distributed across different systems, applications and even different geographical locations. This wave of new technologies has brought with it a number of new challenges associated with the ability to model, design and build such complex distributed systems. To tackle these challenges, it has been argued that agent-based computing provides a promising point of departure (Jennings, 1999). In this context, a software *agent* can be viewed as an autonomous entity capable of flexible problem-solving, and, generally speaking, agent-based solutions involve several such agents (i.e. a multi-agent system (Wooldridge, 2001)) interacting to provide a solution for a given problem. Agents are well suited to a variety of applications (see Jennings (1999) and Luck et al. (2005) for a review) although they are particularly appropriate for applications that involve open, complex and distributed computation, and communication across networked computers (Jennings, 2001).

Until relatively recently, most agent systems have been closed, in that the number of agents and their interactions are predefined by the system designer. This type of system is self-contained and insulated from the external world. Moreover, within the system, the agents are aware of all the other agents, resulting in comparatively little uncertainty in their decision-making processes.

However the ability of autonomous agents to act in a rational, flexible way in uncertain environments has started to result in the application of agent-based computing to the domain of complex and open computer systems. Here, open systems are viewed as those that can be influenced by events outside the defined system boundary, and in these systems it is difficult to monitor the agents and their interactions. Typically, agents can enter and leave the system at will, so that at any given time a single agent within the system will not know all the other agents that are present. Furthermore, agents in an open system are not guided by a universal protocol, nor do they have predefined and universal capabilities. So, in addition, agents are unable to determine the capabilities and behaviour of others, resulting in a system where some (malicious) agents may take advantage of others by behaving in a deceitful way. However, even with this uncertainty in the environment, agents must be able to make decisions and successfully interact with other agents. In such cases, examples of successful interactions include optimal partner selection for collaboration, task delegation to individuals who will perform the activity efficiently and effectively, and fulfilling an agreement between two or more partners. Achieving such success is a challenging task, however against this background, we seek to develop an infrastructure that assures good interactions between the agents within the system, by addressing the uncertainty present in their decision-making.

The remainder of this chapter provides motivation behind using the concept of trust to address the problem described above. Having provided the motivation, in Section 1.2 we describe the role of trust in computer systems, and in Section 1.3 present our research aims. The chapter concludes by providing a summary of the key research contributions and an overview of the whole thesis.

1.1 Managing Interactions in Human Societies

Agent-based systems share a number of common problems with human societies, such as problems with communication, collaboration, negotiation and assurance of good interaction between individuals (human or agent). Now, humans exist in a society, and likewise agents exist in their own virtual society; it is within this virtual community that there is a need to assure good interactions. Therefore, when seeking to engender successful interactions in agent systems, it is natural to look to their human societal counterpart for inspiration. More specifically we examine the role that trust, security and legal contracts play in assuring good interactions.

First let us consider the notion of trust and reputation. Often in human societies we make decisions with incomplete information. For example, we may decide to buy a product from a website; the decision to buy this product will typically include an assessment of the likelihood (uncertainty) of receiving the item ordered, receiving the item in the specified delivery time, or receiving the item at all. We can determine this uncertainty by obtaining and evaluating information that can help make good (or bad) predictions about the aspects of the decision that are uncertain. In our example we can consult the past history of purchasing products from the website or we can ask friends who may have ordered from the same website. In the former case, if each time the company has delivered the item in a good condition within the specified time then we may predict that this is the most likely outcome in the future. In the latter case, if the majority of the friends reply saying that each time they dealt with the website they were pleased with the service, then we may predict that this is the most likely outcome for us in the future. As a final option, we may look to examine the social connections the website has with other companies, or the connections between us and employees or managers of the website. For example, if the website is a sister company of one with which we have had good experience in the past, then we may be inclined to expect similar experiences from the website. In addition, we may expect such (good) experiences if the website owner happens to be a family member or a close friend.

More generally, it can be said that in the human decision-making process we assess uncertainty in the decision by using the concept of *trust* (Williams et al., 1988). Here, trust is a degree of belief in the actions of other people affecting one's own state and choice of action, and it forms an integral component of our reasoning and rationale (Dasgupta, 1990). In this context, evaluating the trustworthiness of another allows us to cope with the uncertainty in the decision-making process. The concept of trust is illustrated in the example discussed previously. When we place an order through the website, we *trust* that the company will deliver the product to us in the condition and at the time specified when placing the order. If the company succeeds in fulfilling the contract, then we would consider increasing our assessment of the *trustworthiness* of the company; likewise if the company breaks the contract then we would decrease our assessment of the *trustworthiness* of the company. In this way, the trustworthiness of an entity is dependent on its behaviour. If it behaves well, or as expected, it earns our trust. On the other hand if it does not, then it risks losing our trust.

However, in an open community it is likely that we will interact with many entities with which we may not have shared an interaction history. We can modify the description of the example to reflect this; suppose that we will be placing an order on a website belonging to a company from whom we have not ordered previously. In this case, we cannot use the outcomes of the previous orders to judge the trustworthiness of the company, because there are no such past orders, but we still have to answer the question of how much trust to place in this new company. Here the answer comes from a variety of sources, such as the opinions of others about the company, or the rating ascribed to the company by a governing body of the industry. Typically, if we do not have enough information to form an opinion about an entity, we consult our peers for their opinion. The common opinion of others regarding an entity is known as the *reputation* of the entity, which may be used in the absence of trust formed from personal opinions.

As we have discussed, trust and reputation are affected by the past actions of individuals. However, there is another important factor in determining the level of trust to place in another. As humans we implicitly evaluate the trustworthiness of others, often by examining the *social structure* that surrounds the individual. For example, we may examine the social status (a *CEO* of a company), the role (a *doctor* at a hospital) and relationships (a *friend* of the family) of individuals in determining their trustworthiness.

Now, while trust allows us to account for uncertainty, it is not the only method employed in human societies. Thus, we now turn to the issue of security and legal contracts since these are alternative methods that can help us to reduce uncertainty in decision-making.

Firstly, the concept of security provides an alternative to trust. By securing something we restrict access to only those that have permission. For example, consider a private golf club that requires membership to allow access to its facilities. In an open society this creates an authority that has the power to issue permission to those that seek it. Often, this single point of entry provides the issuer with the opportunity to assess the intentions, capabilities and behaviour of the seeker. With the golf club, the owner of the club issues membership cards to those that seem fit and capable of paying the appropriate fees. Employing security mechanisms assures that those that have permission are likely to behave in a certain way, reducing the uncertainty about their behaviour. For example the club owner is unlikely to give membership to younger individuals that may cause a disturbance on the golf course. However, the problem with security is that, essentially, it relies on a centralised authority to specify and enforce a certain protocol.

Secondly, the uncertainty in decision-making can be reduced by introducing legal contracts that guide interactions between individuals. For example an employee may have a legal contract with an employer that may specify how and when the employee is paid and what the employee's working hours are. A contract aims to specify the expected behaviour of the parties that are bounded by it. In doing so, a contract creates an environment in which individuals have to conform to a certain specification, and where those that do not conform are punished. For example, the employment contract described above may contain clauses that specify compensation for the employee if the employer breaks any part of the contract. Legal contracts reduce uncertainty by *restricting* the behaviour to that which is deemed legal (allowed).

In summary, we have discussed how security and legal contracts can help individuals, in human societies, address the uncertainty in making their decision. However, this relief in uncertainty is at the expense of either creating a closed system through security or by relying on conformance to legal contracts. Trust offers a method of addressing the uncertainty without such restrictions. Specifically, we have described how trust forms an integral part of decision making in human communities; it acts as the social glue that binds individuals within the society, by allowing us to make decisions depending on the actions of other individuals in light of uncertainty. Given this, we believe that applying the social concepts of trust and reputation in open computer systems can similarly increase the effectiveness of decision-making under uncertainty and help assure good interactions between individual components of the open system. For this reason, we choose the concept of trust as our tool to address some of the problems in assuring good interactions in computer systems.

1.2 Trust in Computer Systems

Computational systems are moving towards large-scale, open, dynamic and distributed architectures, harbouring numerous self-interested agents, often belonging to different organisations. The *Grid* (Foster and Kesselman, 2004) is perhaps the most prominent example of such an environment and, for this reason, is the one we focus on primarily in this research. Others include peer-2-peer computing (Oram and Oram, 2001), pervasive computing (Schmeck et al., 2002), E-Business (Kalakota and Robison, 1999) and the Semantic Web (Berners-Lee et al., 2001) and, moreover, we believe that many of the insights we develop will equally well transfer to these domains. Generally speaking, the Grid is a term used to describe a large distributed heterogeneous infrastructure enabling complex computing facilities for advanced sciences and engineering. As described by Foster et al. (2004), autonomous agents are key to realizing the vision of the grid, as they provide the intelligent and autonomous processes that make use of the computing facilities. Typically, such agents are owned by real life stakeholders, who profit from their actions in the virtual world. With the presence of many agents in the environment, it is likely that agents from different stakeholders will find themselves working together to achieve a goal. However, even though the agents are working together, each has its own set of goals, and this can lead to self-interested behaviour. The concept of *self-interest* is introduced by the translation of social ties that exist between agents' stakeholders in the real world to agents in the virtual world. Thus, we often find that agents in Grid systems are able to form and maintain social ties and a social structure within their virtual community that mirrors the real world. Here, the self-interest introduces the possibility of agents interacting in a way to maximise their own gain (potentially at the cost of another), to provide the optimum profits for their stakeholders. In a Grid context, it is therefore essential to ensure good interaction between agents so that no single agent can take advantage of other agents in the system.

In more detail, many of the interactions between agents in the Grid are conducted in terms of virtual organisations (VOs) (Foster et al., 2004). Specifically, VOs are collections of agents (representing individuals or organisations), each of which has a range of problem-solving capabilities and resources at their disposal. A VO is formed when there is a need to solve a problem or provide a resource that no single agent in the Grid can address. Such VOs often have variable lifespans, the duration of the time that they operate as coalitions, so that they are highly dynamic social structures. Typically agents that form VOs share certain types of relationships. For example an agent in a VO may *depend* on another to provide a particular part of a solution to a problem or before forming a VO two agents may *compete* to win the contract to supply a particular resource. In addition to these properties, they vary in scale, scope, purpose and structure. Moreover, in the Grid, the problems of assuring effective and good interactions between the individual agents are further complicated due to the size of the system and the large number of agents and interactions between them. Nevertheless, solutions to these problems are integral to the wide-scale acceptance of the Grid and agent-based virtual organisations.

Against this background, we argue that the concept of trust is key to assuring good interactions between agents in this domain. Specifically, it can be utilised to account for uncertainty about the willingness and capability of other entities to perform actions as agreed, and not defecting when it proves to be more profitable.

1.3 Research Aims

Trust provides a mechanism, influenced by an individual's personal and social experience, to assure good interactions in an open and dynamic system. Until now, researchers have developed several models of trust in multi-agent systems (see Section 2.4 for a review); however, the context of VOs and Grid computing provides an additional set of issues over and above those of traditional multi-agent systems. We can therefore state the general aim of this thesis as follows:

> *To develop a model of trust, which takes into account environmental and other available information, and aids decision-making in open and dynamic environments, particularly in relation to the formation and management of agent-based virtual organisations.*

In particular, this general aim can be broken down as follows:

Aim: 1
We aim to establish a model that determines a trust level for an individual by examining the past behaviour of that individual. More specifically, we aim to consider the outcomes of past interactions with an agent to calculate a trust level.

Aim: 2
In open systems, and especially in the Grid, it is likely that agents will often be required to interact with others they have not yet had any experience with. For this reason we aim to enhance the above model by considering, in trust determination, the opinions of others that may have interacted with the individual.

Aim: 3
By introducing opinions into trust determination, we increase the likelihood of introducing bias and errors, because opinions given by others are shaped by their subjective experiences. For this reason, we aim to ensure that the model of trust is capable of not being misled, and minimises the error introduced by opinions from others, by judging the *manner* in which individuals provide opinions.

Aim: 4
Since trust is not only influenced by the personal experience and opinions of others, but also by social structures within a society, we aim to consider the role of social structures in the trust model. More specifically, we see to exploit these structures to help minimise the errors introduced by the opinions of others.

Aim: 5
To further enhance the decision-making process, we aim to incorporate into the model a confidence level that represents how much confidence an individual should place in the determined trust value. We believe that such a metric will allow the agent to reason when further evidence is needed in determining a trust level.

Aim: 6
The context of Grid computing in general, and VOs in particular, introduces certain

problems that the model must address. One such problem is the nondeterministic nature of the distributed system; at any point in time it is difficult to determine which network nodes are accessible and which are not. To this end, it is essential that the trust model can cope with the absence of network nodes it relies on, and is therefore distributed and robust (able to cope with network failure) due to the nature of its application domain. Furthermore, the large number of agents within the Grid, and the ability of this number to grow dynamically, requires the model to be scalable.

1.4 Research Contributions

The research reported in this thesis outlines how trust is conceptualised, so that a computational model of trust can be designed for open and dynamic computer systems such as the Grid. In addition to developing and deploying a computational trust model for open, dynamic and large-scale agent-based systems, this work is the first to directly incorporate social information (the knowledge of social relationships that are present between agents in an environment) into aspects of trust calculation. From the contemporary trust models that do consider social information, none detail the manner in which the such information is obtained and translated to a form that can be used in calculating a level of trust. Through our work we further advance the claim that social information is an important facet in determining trustworthiness.

In more detail, through this work we advance the state of the art in the following ways:

- We develop the first probabilistic model of trust for agent-based systems that supports dynamic VO formation and management by agents, giving agents the ability to account for uncertainty in their decision-making processes by assessing the trustworthiness of others. This work has previously been reported by Patel et al. (2005a) and Teacy et al. (2006, 2005).
- We develop a novel component that is used to adjust erroneous opinions provided by other agents, so that an agent is not misled. Many other models attempt to filter and adjust opinions, but through empirical evaluation we show that our approach outperforms other approaches. The results of this empirical evaluation have previously been reported by Teacy et al. (2006, 2005).
- We describe how to implement a computational trust model, for the first time, in a realistic industrial multi-agent system used for providing tailored multimedia service packages through the Grid. Details of this have previously been reported by Patel et al. (2006, 2005b,c), Shao et al. (2004) and Nguyen et al. (2006).
- We develop a mechanism with which an agent is able to obtain the social information required in the trust calculations. More specifically, we describe a simple Bayesian learning

process by which an agent is able to learn the social relationships present in a multi-agent system.

- We specify a novel set of relationship-based heuristics that allow an agent to adjust, in the trust calculation, opinions provided by others, so that the agent is not misled by any bias that maybe introduced by the presence of certain social relationships.

- We demonstrate, through empirical evaluation, the value of using social information in a trust model to enhance its performance. Our evaluation is the first to show that there are certain environments in which using social relationships is advantageous and others in which it is not.

1.5 Thesis Structure

This thesis describes the techniques and tools developed in meeting the aims described previously. We start the presentation of the work with a review of the relevant literature in the field. The purpose of this review is twofold. Firstly, it serves the purpose of providing the reader with background to the problem and its domain. Secondly, through the review of the literature we elicit a list of requirements for a computational trust model that we aim to develop. More specifically, Chapter 2 begins with a general discussion of agents and multi-agent systems that describes how this software paradigm is suited to open complex systems such as the Grid. Building upon this, through the analysis of the Grid computing architecture, and its primary mode of use (VOs), we enumerate a number of trust related issues. We then examine the state of the art in computational trust models, from a variety of domains, to discover how some of the previously enumerated trust issues are addressed. Finally, the chapter concludes with a summary, which presents our working definition of trust and a list of requirements for our trust model.

In Chapter 3 we present TRAVOS — our novel trust and reputation model for agent-based virtual organisations. The chapter describes the core components of the model: (i) a mechanism that allows an agent to convert personal experience with an interaction partner to a level of trust that can be placed in that partner, (ii) a mechanism that allows this level of trust to be calculated in the absence of personal experience, through utilising the opinions of other agents in the system, and (iii) the mechanism used to adjust opinions that may otherwise mislead an agent.

Initially, we validate our approach, in Chapter 4, by showing how the different components of TRAVOS perform by evaluating them in a bespoke simulation environment. We describe how, using the simulation, we compare our approach to the most closely related model from the literature and show how our mechanisms are superior. We provide further validity for our approach by presenting a system evaluation of TRAVOS. More specifically, we present a scenario for trust in an agent-based virtual organisation, and using this scenario we show a walk through to show how TRAVOS can be used in such a system. Finally, we conclude the chapter with a description of the TRAVOS architecture as the trust subsystem in a realistic agent-based VO system.

In Chapter 5 we extend the TRAVOS model by incorporating mechanisms that make use of social information to make the trust calculations richer. We describe a method of allowing agents to learn the inter-agent social relationships that are present in multi-agent systems, and, furthermore, we outline a set of relationship-based heuristics that can be used to make the mechanism of adjusting the opinions of others more effective. We conclude the chapter, aided by a modified scenario, with a demonstration of the application of the heuristics, showing that they can prevent an agent being misled by biased opinions.

In Chapter 6 we show that the use of social information in a trust model enables it to more accurately estimate the behaviour of agents. Again, this is done through empirical evaluation in a modified simulation environment. Through discussion of the results obtained, we conclude when it is wise to adopt an approach that makes use of social information, and when it is not. Furthermore, by use of a scenario, we show exactly how our approach can be used by an agent, in a VO environment, so that it is not misled by others.

In closing, Chapter 7 provides a summary of the research, highlighting the key achievements and drawing final conclusions. Additionally, we outline a number of avenues for further research.

Chapter 2

Literature Review

The aim of this research is to develop a model of trust that helps assure good interactions in the context of agent-based virtual organisations (VOs) in the Grid. In this chapter therefore, we review the literature concerning agents and multi-agent systems, Grid computing and VOs, the broad subject of trust, and the state of the art models of trust in the agent and Grid communities. To this end, the purpose of this chapter is to present such a review.

Initially, we examine the field of agents and multi-agent systems (Section 2.1), with the aim of establishing the characteristics of agent-based systems, and seeing how these characteristics are suited to large scale open and distributed systems (Section 2.2). We then present an analysis of Grid computing and VOs that develops a clearer picture of the application domain, and brings to light problems that may be encountered in developing a solution (Section 2.3).

The field of research concerned with trust is diverse. Trust is a concept that humans use implicitly in day-to-day activities (as per Section 1.1). However even though it is used extensively, the definition of trust, both in human and virtual societies, still remains elusive. Before embarking on developing a model of trust, therefore, it is necessary to understand the various views on trust, and hence Section 2.4 provides a review of trust related literature.

We conclude the chapter by highlighting the open issues not addressed by the current state of the art computational trust models, and provide a summary of detailed requirements for this research.

2.1 Agent Basics

In the software domain, the term "agent" is quite often misinterpreted as it has numerous definitions. Smith et al. (1994) define agents to be "*computer programs that simulate a human relationship by doing something that another person could do for you*". Russell and Norvig (2003) take a more simple view of an agent; they define it to be anything that perceives its environment, and then acts accordingly upon that environment through effectors. Currently there is no general consensus across the agent community on the definition of an agent, but the description provided by Wooldridge and Jennings (1995) is probably the most widely accepted. Here, an agent is a software-based computer system that has the following properties:

- **Autonomy** – An agent should have the ability to function without intervention and have control over its own actions and internal state.
- **Socialability** – An agent should have the ability to interact with other agents and humans using a defined communication language.
- **Reactivity** – An agent should have the ability to perceive its environment and act in response to the changes in that environment.
- **Proactivity** – An agent should have goal-based behaviour that enables it to be proactive and not just react to external stimuli. The agent's goals should drive its actions.

The confusion over the concept of an agent is not only caused by the numerous definitions, but is also added to by the presence of many agent architectures that offer particular methodologies for building software agents. Here we briefly discuss agent architectures to help develop a more complete understanding of agents, which is required to consider the issues discussed in Section 2.3 and 2.4. It is important to review the different architectures because the trust model that will be developed in this research will be embedded within an agent architecture (see Section 4.2).

Architectures exhibit certain characteristics that help specify how the main problem is broken into smaller problems, that a single (or multiple) agents can solve. Typically they provide a description of the overall behaviour of an agent by describing the internal components and the interactions between these components. In addition, they provide the algorithms that bring about actions the agent performs and its future state, from the agent's percepts and current internal state. It must be noted that many agent architectures exist in the literature, perhaps the three most widely used and accepted classes being the *Deliberative* (see Wooldridge (2001), Weiss (1999) and Rao and Georgeff), *Reactive* (see Brooks (1991)) and *Hybrid* architectures, but it is outside the scope of this work to examine each in detail. Instead, here, we briefly consider the requirements a computational trust model may have on the architecture of the agent it is to be embedded in.

Typically (as can be seen through the discussion presented in Section 2.4 most computational trust models contain both reactive aspects (for example an agent can be configured not to interact with others with a trust value lower than a threshold) and deliberative aspects (for example an agent can deliberate over a trust value for a particular agent and decide that it needs more evidence to calculate a more accurate trust value). To this end, when we arrive at implementing our model in an agent architecture we choose to do so in an architecture based on the hybrid approach.

2.2 Multi-agent Systems

In a distributed system, it is common to have more than one agent (forming a multi-agent system). Luck and d'Inverno (2004) define a multi-agent system as one "*in which several distinct components, each of which is an independent problem-solving agent, come together to form some coherent whole*". In a multi-agent system there is no goal that drives the whole system; instead each agent entity has its own goal. This means that the agents have to cooperate and coordinate within the multi-agent system and therefore form a coherent system that serves the end-user. This coordination and cooperation is a necessity to "*avoid duplication of effort, unwittingly hindering other agents in achieving goals and to exploit other agents' capabilities*". For example, we can consider a football team as a multi-agent system, where each player has their own special ability and plays a part in trying to make the team win. The players are likely to cooperate and coordinate in such a way that no one player ends up running with the ball at all times, and not every player flocks to the ball. In such scenarios, and as we have done so far, multi-agent systems are shown as teams of agents working together in an effort to achieve a common goal. While this is true of traditional multi-agent systems, these systems are now becoming open with agents owned by different stakeholders. In such systems, agents may still work together to achieve a common goal, but they are motivated by their own goals and utility, leading to *selfish* behaviour (as described in the following sections).

2.2.1 Multi-agent Systems in Open and Complex Environments

In open multi-agent systems, agents can enter and leave when they want, in turn creating a dynamic environment. In these systems, agents typically represent different competing organisations and, for this reason, we assume that these agents are self-interested. Here, the notion of self-interest means that an agent will act in a manner which maximises its own utility given its own goals. The reasons for the applicability of agents to open and complex are given below (adapted from (Jennings and Wooldridge, 1998)):

Open systems — The structure of an open system is capable of changing rapidly. Typically, the entities within an open system are heterogeneous, having been developed by different people at different times. The entities present at any given time in the open system are not known in advance, undergo continual change, and are considered dynamic. Even with the presence of these characteristics, an open system must operate without much interference from users or system designers. A multi-agent system offers the ability to model these entities as autonomous agents, capable of negotiating and communicating on their own without user intervention.

Complex systems — Modularity and abstraction are techniques that help reduce the complexity of a problem, and therefore make the task of solving the problem easier by breaking it down into smaller problems. Agent-based systems are inherently modular and can offer this characteristic to complex systems, with each smaller problem being solved by a single agent, and the overall solution being produced by the interaction of these problem solving agents. This model of problem-solving enables the system designers to abstract the complex software as a virtual community of interacting problem solvers.

2.2.2 Multi-agent Systems in Grid Based Virtual Organisations

The Grid is an open and complex distributed computing environment and, having justified the use of agents in an open and complex domain, it can be said that agent-based systems exhibit characteristics that are well suited to the domain of the Grid (Foster et al., 2004). More specifically, a Grid-based virtual organisation (VO) system can be modelled as a multi-agent system in which:

1. agents are owned by different organisations;
2. agents have different problem-solving capabilities; and
3. agents have the choice of interacting and cooperating with each other.

However, using an agent-based approach has certain drawbacks. Its inherently distributed nature, and the presence of many individual problem-solving agents, means there is no overall system control. A further disadvantage, due to its scale, is that each agent only has a localised viewpoint of the entire system. This means that computational power has to be used to allow an agent to interact with all other agents in order to build up a more complete view of the system (in the majority of cases this will not be required because each agent is responsible for part of the system and does not have the need to develop a system wide view). In addition to these natural limitations of agent systems, the application of agents to large-scale open distributed environments introduces further challenges. Ramchurn et al. (2004a) identify a number of these new challenges:

1. Agents are likely to represent different stakeholders and each is likely to have its own aims and objectives. This will be the case in a VO, as each entity may belong to different organisations.
2. Agents are capable of leaving and re-entering the system under a new identity to avoid penalty for past actions.
3. Agents in an open system can exhibit different behaviours, and will themselves be faced with a choice of interaction partners showing different characteristics.

We believe the above challenges can be met by using a computational model of trust, as discussed in Section 2.5. However, before we move to a discussion about how some of these challenges may be addressed by use of a trust model, we examine the nature of the Grid, and the specific characteristics of VOs, with respect to issues concerned with trust and behaviour of agents.

2.3 Grid Computing and Virtual Organisations

The state of the art Grid has evolved over the last decade from numerous different systems (for example (FARNER, 1995) and I-WAY (Foster et al., 1996)). The earliest efforts were named "*metacomputing*" and involved linking various supercomputing sites. These early metacomputing projects were created in order to give high performance applications access to large computational resources. The state of the art Grid is defined by Foster and Kesselman (2004) as a computing system that solves the problem of coordinated resource-sharing and problem-solving in dynamic, multi-institutional VOs. The current Grid infrastructure requires an extensible and robust architecture, in which much consideration has to be given to interoperability, protocols, services and APIs to aid the formation and operation of dynamic virtual organisations.

Grid computing has developed in response to address the specific needs of a computing infrastructure that supports VO-like structures. In the Grid, electronic agents owned by different organisations can work in collaboration to achieve a collective goal. In fact, VOs are a key to realising the grand vision of the Grid and provide the principal style of use for Grid technology. In more detail, Foster et al. (2001) describe a VO, in the context of a Grid, as a set of individuals and/or institutions defined by a set of sharing rules (which define access to various resources by consumers and suppliers within the system). The VO can be seen as having a definite lifespan, consisting of a few key stages shown in Figure 2.1, and described below:

1. **Formation** — When a need arises, a VO is formed. A group of agents form a collective that is capable of addressing a special need that cannot be addressed by each of the agents on their own. This stage can be described as partner selection, because each of the agents is surveying prospective partners with which they will form a VO.

2. **Functioning** — During the functioning phase of the life cycle, the agents collaborate with the other VO members under a set agreement (or contract).

3. **Restructuring** — Under certain conditions it may be necessary to restructure a VO. For example, restructuring may occur if the initial requirements for the VO change or if a particular member of the VO provides a service below an acceptable level and is asked to leave the VO. The result could be a requirement for a new member to join, or redundancy of an existing member.

4. **Disbanding** — This is the final stage of a VO life cycle. This occurs upon completion of an agreement, after which the agents are not under any contractual obligation to work as a collaborating collective. The disbanding may also be necessary if it is computationally less expensive than major restructuring.

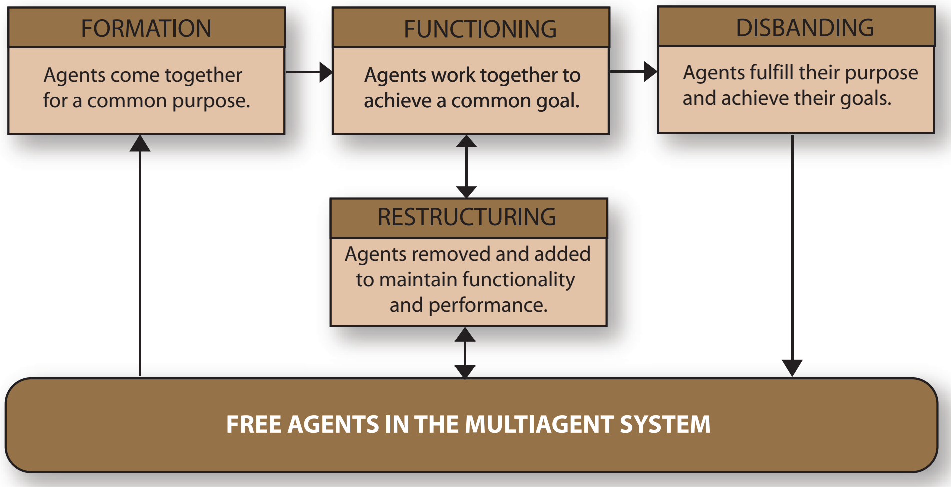

FIGURE 2.1: The main stages of a virtual organisation's lifespan.

The introduction of VO structures across a distributed domain has certain requirements and some common concerns. These are outlined below (adapted from Foster et al. (2001)):

1. VOs require highly flexible sharing relationships. The rules and policies that define the sharing of resources among the constituents of a VO have to allow for the complex sharing relationships that are defined within the VO.

2. VOs require sharing of resources across a spectrum of systems, ranging from peer-to-peer to client-server.

3. A VO may require a sophisticated and precise level of control over how shared resources are actually utilised.

4. There are also few restrictions on how many and what type of resources may be shared. Therefore, the mechanisms using these resources have to be flexible.

5. Finally, the VO may have to support a diverse set of usage modes, ranging from performance-sensitive to cost-sensitive. This gives rise to issues of accounting, quality of service, scheduling and co-allocation.

The stages in the VO life cycle and the exploitation of VO-like structures require a new type of technology, which is capable of meeting the requirements and common concerns outlined above. Grid computing offers the technology to meet such requirements; we now present an analysis of the Grid architecture that describes the key components, and elicits the issues that impact on the trustworthiness of individual entities found within VOs that make use of the Grid.

2.3.1 The Grid Architecture

The Grid architecture offers a collection of fundamental components, and interactions between these components, that collectively meet the requirements of VOs (Foster et al., 2001). In more detail, Foster et al. adopt an hourglass model to specify the layers of the Grid architecture, as shown in Figure 2.2). This architecture classifies the components into a set layers, which help in identifying the general requirements for the components, resulting in an open architecture that allows the creation of solutions that meet VO requirements.

The narrow neck of the hourglass is a set of abstractions and protocols, which are important for two reasons. Firstly, much of the high-level behaviour found at the top of the hourglass can be mapped onto this narrow neck. Secondly, the core abstractions and protocols can themselves be mapped onto many different underlying technologies that enable the overall operation of the Grid. Specifically, the resource and connectivity protocols are designed so that they can be implemented over a range of diverse resources found in the fabric layer. A range of services and behaviours, which are found in the collective layer, can be constructed from the resource and connectivity protocols. Our work is situated in the application layer, which utilises the capabilities offered by the collective layer. More specifically, the trust model will be embedded into an agent-based application within the application layer. We do not work at the lower layers, but a description of all layers is included for completeness of the context.

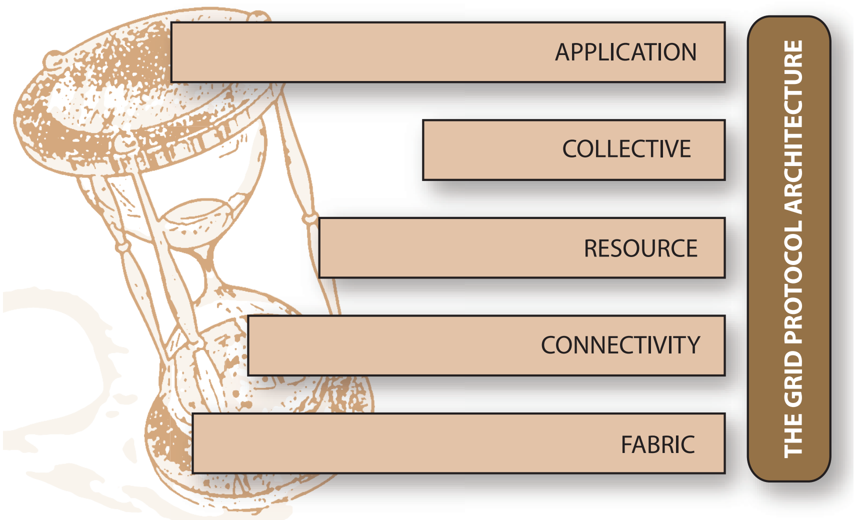

FIGURE 2.2: The Grid architecture (adapted from (Foster and Kesselman, 2004)).

The *fabric layer* consist of the geographically distributed resources, for which the Grid system provides access to protocols in the resource and connectivity layer. Examples of resources include storage systems, network resources, sensors and actuators. The purpose of this layer is to house the components that implement the local resource-specific operations that a resource is capable of performing. The decision of how much of the functionality of a resource is to be implemented in the fabric component is a trade-off between enabling more complex sharing relationships in VOs and the complexity of deploying the Grid architecture. In order to keep the deployment of the Grid architecture simple, and enable the core operations that users may request from a resource, the fabric components are required to implement at least two main mechanisms. Firstly, they are required to implement enquiry mechanisms that enable the resource to communicate its structure, state and capabilities. Secondly, they are required to implement resource management mechanisms to allow control over quality of service.

The *connectivity layer* consists of core communication and authentication protocols for Grid-specific transactions. This layer is responsible for enabling easy and secure communication within the Grid. The communication protocols provide methods of data exchange between fabric layer resources. These protocols are primarily drawn from the internet, transport and the application layers of the internet layered protocol architecture (Baker, 1995). The authentication protocols provide secure mechanisms for verifying the identity of users and resources. These protocols are required to have certain characteristics in order to support VO environments; these are summarised below (from (Butler et al., 2000)):

1. **Single sign on** — Users should be able to log on only once and gain access to a variety of resources they are authorised to access.
2. **Delegation** — A user must be able to execute a program that is capable of accessing the resources the user is authorised on. Additionally, the program should be able to delegate subsets of these rights to other programs based on certain conditions.
3. **Integration with various local security solutions** — Grid-based security solutions must be able to interoperate with various local security solutions, thus they must provide a mapping onto the existing local security infrastructure.
4. **User-based trust relationships** — The authentication solution should allow the user to access a variety of resources without requiring the security administrators of the resource providers to interact with each other.

The *resource layer* is a collection of protocols that builds upon the protocols of the connectivity layer. This collection provides the abstraction above the fabric layer functions, and mechanisms for secure negotiation, initiation, control, accounting and payment of operations of resources found in the fabric layer. Each of the protocols can be classified into two main classes:

1. **Information protocols** – These are used to obtain information about a resource (for example current loads and configuration).

2. **Management protocols** – These allow an entity to negotiate access to a shared resource by specifying certain resource requirements, such as minimum quality of service expected. Many of these also support status monitoring and controlling of the operations carried out on the individual resource.

The *collective layer* is responsible for the coordination of multiple resources. Global protocols, which capture interactions across collective resources, are employed to achieve this. Principally, they implement the wide variety of sharing behaviours to enable the VO life cycle, for example directory services, scheduling services, monitoring services and data replication services.

The *application layer* is the top layer of the Grid architecture. It contains the user-specific applications that are developed and implemented by accessing any of the services defined at the lower layers. The computational model of trust that this research aims to develop will be applicable in this layer of the Grid architecture. It will facilitate and assure interactions between software agents that are part of a particular application, implemented in this layer.

2.3.2 Current Issues in Grid Computing

The description of the Grid architecture shows how the design of the Grid computing infrastructure is capable of supporting the functions required by the VO life cycle. However, the ability to cope with the stages in the VO life cycle introduces some issues which need to be addressed by applications found in the application layer. The main issues are as follows (adapted from DeRoure et al. (2003)):

1. **Heterogeneity** — The Grid can encapsulate numerous resources that are heterogeneous and that span several administrative domains. This heterogeneity is an inherent property of large distributed systems and a successful Grid application masks the heterogeneity of the underlying Grid environment to create seamless integration, and provide a user-oriented interface for interaction.

2. **Scalability** — Often networks grow rapidly in size, and any application that is situated above networked entities has to be scalable. Scalability is not an issue related to the number of physical entities in the network alone, but also to the geographic distribution of these entities and their organisational affiliation.

3. **Adaptability** — In a system that is composed of many resources, the probability of failure of one or more resources is high. Resource managers and Grid applications must be able to detect and manage the pool of resources to ensure effective performance of the system.

4. **Security** — A large open networked environment has many security related issues. Many of these are concerned with ascertaining the true identity of the entities that exist within it. Security mechanisms have to establish that entities are who they advertise they are, and to ensure authentication and confidentiality of certain private and sensitive interactions.

Currently the most widely used Grid infrastructure is the Globus Toolkit (Foster and Kesselman, 1997), which provides a number of mechanisms to aid in addressing the issues detailed above. The Globus Toolkit provides a range of protocols, components and standards that are found throughout the different layers of the Grid architecture. One of the drawbacks of the toolkit is that it lacks mechanisms to address the problems relating to security (in particular at the application layer). The issue of security is tackled by the *Grid Security Infrastructure* (GSI) (Foster et al., 1998), which provides basic security properties and is used as the standard in the Grid community. However, this too fails in addressing the complete set of issues that fall under the security category. More specifically, Li et al. (2003) highlight certain drawbacks of GSI, identifying problems of uncontrolled delegation, leaky infrastructure, and insecure services.

Security is a significant problem and there is a substantial part of the Grid community working on solutions to problems relating to it. A large part of these solutions are based on *trusted* certificate authentication, and the word "*trusted*" is used interchangeably with the word "*secure*". This is especially the case in the connectivity layer (described in 2.3.1). Mostly, this type of trust is built upon authentication of identity certificates, and if this identification process provides a positive result then the party in question can be deemed trustworthy. This notion of trust is little more than a façade over the well established field of security. However, more recently, due to the nature of the many heterogeneous entities, the use of security has evolved. The evolution of the old security related issues has resulted in the need to have a much *softer* approach to security within these large scale virtual communities (Rasmusson and Jansson, 1996). The *hard* approach provided by security fails to meet the flexibility required to be applicable in these situations (an example of this is applications that use multi-agent systems to facilitate grid-based virtual organisations, the properties of which are discussed in 2.2.2). This does not mean that security is redundant and that a replacement must be found, however, indeed it is essential that the lower level security protocols and mechanisms remain in place, but it is also necessary to have a softer (more flexible) form of security between the interacting entities in these virtual societies. We believe that a computational trust model can be used to provide this necessary flexibility, and, in the following section, we describe in detail the trust related issues in the Grid environment.

2.3.3 Trust Issues in Grid Computing

Having discussed the major issues in Grid computing, here we explore their implications on trust. Many of these implications can be credited to large scale distributed systems, and whilst they are discussed with respect to the Grid, it must be noted that they are applicable to any similar distributed domain.

Due to heterogeneity and adaptability issues, in the Grid environment it is likely that one may find an entity working in a non-deterministic physical network, where there is a constant threat of broken links outside its influence. Now, despite an entity's best efforts, it may come across as being untrustworthy if interactions with it fail (for example if it fails to provide a service) due to the intermittent network connection. Some may argue that in such cases the trustworthiness of the entity should not be affected because the reason for the shortfall in expectation is due to the environment and not the entity's intentions. On the other hand, we believe that while it is important to distinguish between the shortcomings of the environment and the behaviour of the agent, from the perspective of the interaction partner the end result is the same – a failed interaction.

Much like other applications developed to operate on the Grid architecture, a trust model has to be scalable, so that it remains practical and useful regardless of the number of entities involved. The current vision of the Grid is that of a large, geographically distributed system that will grow from the combination of many smaller systems, so that the number of entities interacting in our target environment may therefore vary by several orders of magnitude.

Due to the dynamic and non-deterministic nature of the Grid, the model should be robust in the face of the failure of system components. If provision and use of trust is to successfully work in a large distributed environment, we must take for granted that elements of the system may fail on a regular basis, and must take steps to minimise the effect of such failures on the performance of the system as a whole. To this end the system should not be centralised, and a distributed approach should be taken during the design of the model.

A very big area within the Grid where security and trust are needed, concerns elements leaving the system and entering under a false identity. This problem is one that has to be solved, and currently the solutions are very limited. The GLOBUS toolkit limits this behaviour, but by no means does it prevent it. Currently there is only one model that provides a mechanism that removes the incentive for agents to exhibit this behaviour (see Section 2.4.1.1), but this still does not stop an agent leaving and entering the system under a new identity. We believe that the solution to this problem lies at a lower security level, rather than at the level that trust can be used in the system. A model of trust for use by agents will not be able to provide a definite method of identification of fake identities, unless the lower level security mechanisms can be used to do so. The problem is thus outside the scope of the current study and will not form a requirement for our trust model.

Given the discussion above, we can now enumerate some requirements for a trust model for the Grid domain:

> **General Requirement 1**
> *Scalable model* — The model should be scalable, and its performance should not be affected by the number of entities added to it.

> **General Requirement 2**
> *Decentralised model* — The model should be robust, and continue functioning even if there are problems with part of the network on which it has to operate.

> **General Requirement 3**
> *Distinguish between entity and environment* — Due to the non-deterministic nature of the network on which it is to operate, the model should distinguish between the role of an entity and the role of the environment in the perceived behaviour of the entity.

Having considered trust issues in the Grid in general, we now examine trust related issues present in different stages (formation, functioning, restructuring and disbanding) of the VO lifecyle.

2.3.4 Trust Issues in VO Formation

In a large open system it may be assumed that a large number of agents may not have interacted before. In these systems there may also be entities that are self-interested or malicious. For these reasons it is important to select the right (reliable) interaction and VO partners. A trust indicator, suggesting which parties are more trustworthy than others, would prove to be a useful factor in the partner selection phase (Griffiths and Luck, 2003). The notion of trust may also prove useful in negotiation to form contracts that bind the VO together in this initial VO formation phase.

Initially, an agent may choose as its partner, an agent with whom it has had many previous encounters. It is through past behaviour that it is able to gauge the capabilities, behaviour, and trustworthiness of its potential partner. However, during the formation phase of a VO, all the agents are in an open system, where agents may come and go as they wish, and it is likely that, due to the sheer numbers, the *majority* of agents will not have interacted with each other. This poses an interesting problem regarding trust, because if we assume that trust is evaluated based on past performance, then how do we evaluate someone that we have not encountered or interacted with previously?

When this scenario occurs in human societies we often look towards the *reputation* of the potential interaction partner (see discussion in Section 1.1). In an open system, while it is unlikely that we may not have interacted with the potential partner, it is likely that there are others in the system that have. Therefore, the potential interaction partner has a reputation, which is a result of its behaviour with others in the system.

> **General Requirement 4**
> *Calculate direct trust* — The model should allow an agent to calculate a level of trust for a potential interaction partner (or more generally another agent) based on the past experiences of the agent with the potential partner.

> **General Requirement 5**
> *Calculate reputation* — In cases of no prior experience, the model should allow an agent to calculate a level of trust for a potential interaction partner based on the opinions of others who *have* had prior experiences with the potential partner.

However, the concept of reputation in an open agent system raises a set of problems that need to be addressed, as follows:

1. We need to determine the *kind* of agent to obtain reputation information from (e.g. trusted system agent or a peer). In human societies, for example, if we are interviewing a candidate then we may choose to ask the candidate's friends, previous employers or teachers to provide us with their opinions, or we may examine the candidate's personal records at the hospital or the police station.

2. Having identified the kind of agent, we must identify the specific *individual* agents from all of that kind. Extending our example, having decided that its wise to ask a previous employer for an opinion, we are left with the decision of which particular employer to ask.

3. Having identified an agent, we need to determine how to actually access that individual agent to obtain an opinion. Again, with reference to our example, having identified a particular employer, we are left with a choice of how to approach, and request their opinion.

4. We need to consider the need for an incentive for the other agent to provide reputation information about another. In our example, we may approach an employer, but the employer may ask for a payment in return for their opinion. This is an important point, since in human societies there are many businesses whose sole purpose is to act as brokers for reputation information.[1]

5. Finally, in an open system, due to the variability and non-determinism in agents' behaviour, it is important to consider the reliability of opinions provided by others.

The issues presented above offer us more requirements that must be satisfied by a good trust model:

> **General Requirement 6**
> *Finding reputation* — The model should offer mechanisms to allow the identification of potential sources of opinions, and effective protocols of how to obtain those opinions.

> **General Requirement 7**
> *Incentives for providing opinions* — To allow it to meet Requirement 5 the model should provide a reason for agents to offer their opinions about another.

[1]Examples of businesses that profit from selling or publishing opinions (and other related revenue streams, such as advertising alongside those opinions) include *Which?* magazine (`http://www.which.co.uk`) for opinions on various items for consumers, and *Parkers – Car Price Guide* (`http://www.parkers.co.uk/`) for prospective car buyers.

General Requirement 8
Adjust unreliable opinions — The model should provide a means of assessing the reliability of other's opinions, so that an agent is not misled by false opinions.

2.3.5 Trust Issues in VO Functioning

During the functioning phase of the life cycle, the agents work collaboratively under a predefined agreement. They work towards a common goal, such as providing a composite service (that is an amalgamation of their individual services) to an end user.

When a VO is formed, the agents are bound by a legal contract, which typically requires them to conform to a particular behaviour or requires them to meet a set of definite expectations. This is reason enough to assume that under such restrictions it is not necessary to consider trust – everyone is likely to achieve what is expected. Additionally, calculating trust may be computationally expensive for an agent, and therefore it may decide that due to the presence of a contract binding the actions of others, there is no need to evaluate their trustworthiness. In this case, all agents may adopt a trusting disposition towards their fellow VO members.

However, if complete trust is assumed and trust values are not used in the VO, the question of whether an agent still needs to update the information about others, which it uses to calculate trust levels, needs to be addressed. It is necessary to evaluate the performance of other agents, with respect to trust, even in a closed system such as an individual VO. This is because bad partners may be selected in the formation phase, due to lack of information available when calculating the trustworthiness of agents in an open system (prior to VO formation). The continual assessment of the VO partners with respect to trust is essential so that an agent may make a more accurate partner selection decision next time a VO is formed.

Whilst a VO is functioning, its internal organisation may be analysed using social relations. For example, Figure 2.3 shows a VO that may be classified as having a hierarchical structure (with one agent at the top of the social structure) and one with a peer-based structure (where all agents are socially equal). Social relationships are not a prerequisite for agents to form VOs with each other, however certain relations may emerge between agents that regularly interact together. These relationships may lead to agents behaving in a particular manner towards each other, for example an agent may behave biased towards another with which it regularly forms a VO. Therefore, observing and learning the social relations that emerge from regular interaction is crucial in assessing the expected behaviour of an agent (both with respect to its behaviour inside and outside a VO). Social structure plays an important part in trust and trusting, we can see this from human societies. For example, an organisation is unlikely to take advice from a competing entity without considering the fact that the information it receives may be misleading.

During the functioning of the VO there may be occasions when the VO fails to meet a requirement that was agreed upon in the contact. From a trust perspective, it is important to isolate the individual or group of individuals within the VO that were the cause of this failure, due to the

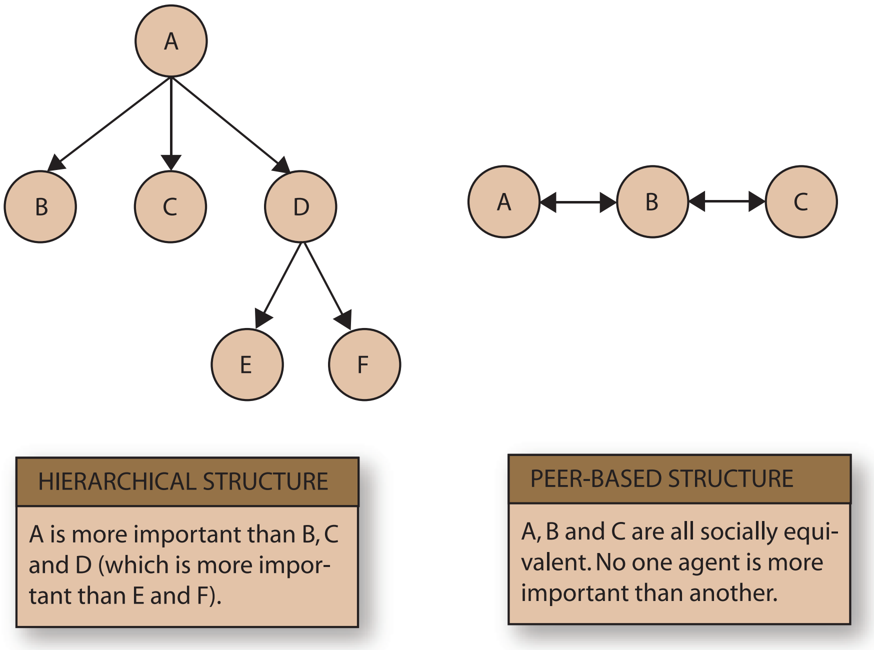

FIGURE 2.3: Examples of social structures in virtual organisations.

fact that agents in the system are likely to lose trust in the parties upon whom the blame falls. The identification of who to blame is outside the scope of this study[2], however it is important to consider the way in which trust changes when blame is assigned (to an individual or group) and when it is unattributed. For example, a consumer of the service offered by a VO may reduce their trust in all the members of a VO if they cannot assign a blame for a fault in the service to a single or group of agents. Likewise, they may lose trust in only those that are assigned the blame for the fault.

From the analysis of the trust issues in the VO formation stage of the VO life cycle, we obtain the following requirements:

General Requirement 9
Maintaining interaction history — Regardless of an agent operating in a VO or outside a VO (in the mix of free agents in the multi-agent system) it is necessary for the trust model to store the experiences of the agent for future trust calculations.

General Requirement 10
Use social information — Since social structures are present in VO environments, and are important in calculating trust, it is important for the model to include social factors and information in the calculation of a trust value.

General Requirement 11
Dynamic trust value — The model should provide a means of changing a trust

[2]For further details about research concerned with blame assignment and fault finding, see Grossi et al. (2004).

value (that an individual agent calculates for another) based on the experience of an individual agent. More specifically it should dynamically change trust values based on the individual's observations over time. Every observation should have a defined impact on trust.

2.3.6 Trust Issues in VO Restructuring

A VO may require restructuring due to a number of different reasons. For example, in the contract a service may become redundant, or the contract may be modified to reflect the changes in the user requirements which may then necessitate the addition of a new service.

In restructuring the VO there are many issues regarding trust that have to be addressed. Most of these are concerned with group-level decision-making, by which we mean that the VO as a whole must assess how much trust can be placed in an individual (that is to be eliminated from or added to the VO). There is a definite need for a mechanism that collates the distributed trust information from members of a VO about a single non-member agent, so that it may be used in making a group decision that affects the structure of the VO.

Similarly, it is necessary to be able to represent and evaluate the trustworthiness of groups of agents. Up until now trust has been discussed with respect to a single individual, but VOs introduce the need to represent trust and reputation of groups of individuals. When a VO is restructuring, the VO members may find that there is another smaller VO that they wish to incorporate into their own VO. In this case, the members must evaluate the smaller VO as a whole, since they will have to place their trust in the VO. The need for representing trust of a collective is further complicated by the fact that a VO is dynamic and may have a short lifespan. Therefore, in the context of VOs it is necessary to represent the trust of groups of individuals that form a VO, independent of the VO lifetime.

The restructuring stage in a VO's life cycle provides us with two further requirements for a trust model:

General Requirement 12
Trust level consensus — The model should provide mechanisms that allow a group of agents to come to a consensus about the trust they are all willing to put in a particular individual.

General Requirement 13
VO-level trust — In VO environments it may be necessary to evaluate the trustworthiness of a VO and in such cases the model should provide mechanism to produce a VO-level trust values from the trust values of the individual members.

2.3.7 Trust Issues in VO Disbanding

VOs disband because they reach the end of the contract that binds them and the need that they fulfill no longer exists, or if the VO has to undergo such a radical restructuring that it may be more efficient to disband and reform again. In either case, during its lifetime, the VO as a whole acquires much information that can be used to assess an agent's trustworthiness. This information is distributed across the members of the VO, and will be subjective to the owner of the information, introducing a further trust related issue regarding the fate of this information. Upon disbanding, the information might:

1. stay internal to the agent that acquired it;
2. get reported to a central repository, making it available to all agents;
3. get reported to all the members of a VO before disbanding.

In addition, the context in which the trust information was obtained may change so that it may have to be used differently when the VO has disbanded. For example an agent A may have recorded that agent B always delivers in time whilst in the VO, but this might not be the case outside the VO. This is because outside a VO there might not be a contract specifying that if B fails to deliver then it is penalised heavily.

Finally, from the last stage in the VO life cycle, we obtain the following requirements, which are concerned with the manner in which trust information (used in calculating trust values) is stored and accessed:

> **General Requirement 14**
> *Effective exchange of opinions* — The model should provide a means of recording trust information (evidence used in the trust calculation) in a way that can be shared quickly and effectively.

> **General Requirement 15**
> *Distributed trust information* — Given Requirement 2 (decentralised model), the model should not rely on the trust information being in a central place.

> **General Requirement 16**
> *Context dependent* — The model should store and use trust information in a way that allows agents to factor in the context in which the information was obtained, in the trust calculation.

2.4 Computational Models of Trust

Having reviewed multi-agent systems and agent-based virtual organisations, we now turn to the notion of trust. First, we discuss some generic definitions of trust, and then we present a review of computational trust models in light of the general requirements developed in the previous section.

Many sociologists have carried out research on the idea of trust within human society (for example (Misztal, 1996), (McKnight and Chervany, 1996) and (Williams et al., 1988)), resulting in a number of definitions for trust. A brief review of the literature concerned with the philosophy of trust and the application of trust-related concepts in computer science reveals the need to define trust before attempting to create a trust model. To this end, Dasgupta (1990) defines trust as a "*sense of correct expectations about the actions of other people that have a bearing on one's own choice of action when that action must be chosen before one can monitor the actions of those others*". This definition allows us to extract certain attributes of trust, which would help in building a computational model of trust. Specifically we can see that trust is both an expectation and a value that is estimated and used for judgement. Dasgupta also states that trust has no "*obvious units*", but it is measurable in its context. Identification of this attribute is essential, as it has an impact on a computational model of trust. Computationally it is better that we have explicit values (for example integer values) in place of vague measurements of trust, values that can be compared and calculated. Dasgupta's definition is adequate to the study of trust in society; however, for use in a computer system we need a more formal definition of trust, because mathematical concepts are easier to transfer to a computational model.

To this end, Gambetta (1988) offers a more probabilistic definition of trust and summarises it as "*a particular level of the subjective probability with which an agent assesses that another agent will perform a particular action, both before he can monitor such action and in a context in which it affects his own action*". He defines trust as a probability that has a threshold value, which can be "*located on a probabilistic distribution of more general expectations, which can take a number of values suspended between complete distrust and complete trust, and which is centred on a midpoint of certainty*". This definition clearly views trust as being a probability distribution representing complete trust and distrust at the two extremes of the distribution, also explicitly embodying the certainty of the trust that it represents. This is the definition of trust that we adopt in this work.

Having reviewed the general definition of trust from a sociological perspective, we now examine state of the art of computational models of trust. The purpose of our review is to examine techniques that have been used to model the concept of trust and to examine whether any of the current models meet the requirements stated in the previous section. However, before we examine trust models, we consider the different ways in which the models can be classified. A simple way in which to do this might be to group them based on where they can be applied, but this approach is not valid as almost all the models seem to be aimed at different domains. Instead, Ramchurn et al. (2004a) propose a classification made up of two classes: *individual-level* trust and *system-level* trust. The former contains models that allow agents to have beliefs about the behaviour and trustworthiness of others in the system, whereas the latter groups together models which enforce trustworthy behaviour through certain protocols and mechanisms. Classifying models in such a way allows one to see, as described in Ramchurn et al. (2004a), the parts of the trust puzzle that are solved and the gaps that exist between higher-level trust models (individual-level) and the lower level (system-level) trust protocols. As described in Section 2.3.1, we are concerned with developing a model of trust that is to be applied at the application layer of the Grid architecture, and one that is not aimed at *enforcing* trustworthy behaviour. For this reason, and according to this classification, we only consider the models which fall under the individual-level classification.[3]

Furthermore, the majority of trust models can be divided into two distinct categories based on the overall architecture of the model: *centralised* and *distributed*, reflecting the nature in which information used to calculate trust is stored. In the centralised approach trust information is stored in a central repository, whereas in the distributed approach trust information is distributed amongst the many entities in the system. The rest of this section is divided into two main parts, each discussing a specific architecture along with models that fit that architecture. In each case the description of a model provides a brief overview, the definition and composition of trust used, and an evaluation of the model.

2.4.1 Centralised Models of Trust

In centralised systems there is a particular entity that is responsible for the activities of gathering trust information from the community, performing calculations on this information (for example, calculating a reputation value from all opinions), and making the results of its calculations public to anyone in the community. In this context, trust information largely refers to opinions of individuals about the behavior of others. Often this architecture is used by online communities such as eBay[4] and Amazon.com[5]. Figure 2.4 shows a centralised system, where a central authority known as a *reputation centre* which is updated by individuals when they provide their opinion of another (as a rating), after each interaction. After each update, the centre calculates new reputation values based on a function of the opinions stored. The reputation values are then provided to all who query the centre for reputation information.

In more detail, Jøsang et al. (2006) describe two key components of centralised trust systems:

1. **Centralised communication protocols** — These allow communication between individuals in the community and the central information repository. More specifically, they describe methods to provide opinions to and obtain reputation values from the central authority.

[3] Here, we are not stating that all system-level trust approaches are not valid, and where possible we will draw inspiration from such models.

[4] `http://www.ebay.com`

[5] `http://www.amazon.com`

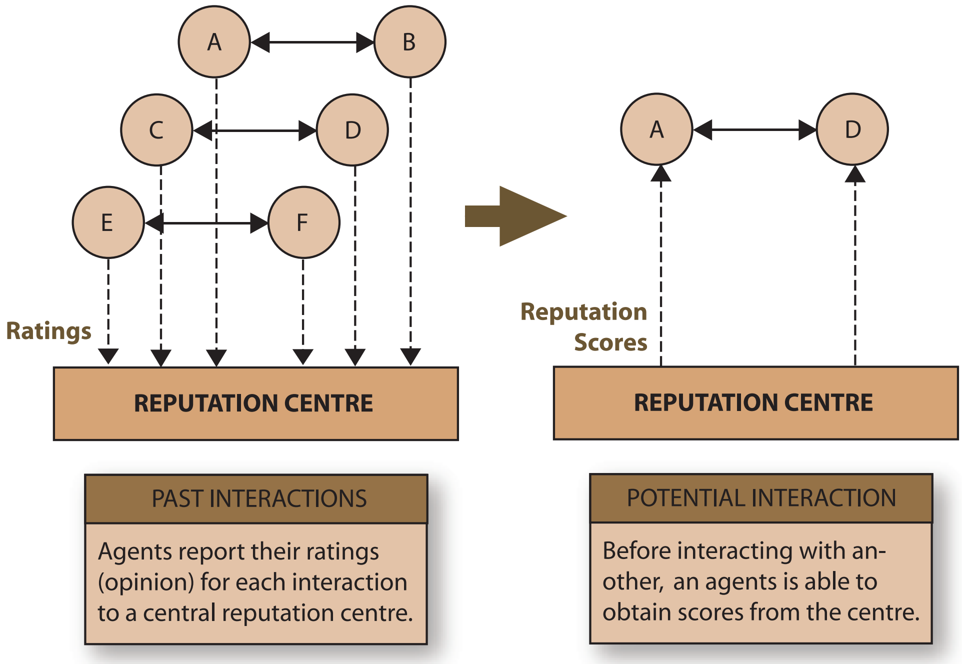

FIGURE 2.4: A centralised trust system (adapted from Jøsang et al. (2006)).

2. **Reputation computation engine** — This is used by the centre to calculate a single reputation value for an individual from all the opinions provided by others in the community for that particular individual.

This architecture's key advantage is that the protocols used are very simple, and centralisation of information means that individuals do not have to seek information that is distributed across many members in the community. Often this provides a simple solution to General Requirement 6 (finding reputation), but it does introduce two main disadvantages. First, the approach does not meet General Requirement 1 (scalable model), as there is a limit to the scalability of such an architecture. Thus, if the numbers of individuals and their interactions within the community increase, the reputation centre may struggle to serve requests made upon it. Second, when a truster requests a reputation value from the centre, it does not know the actual opinions (and its source) that are used to form the single value that it receives. As for General Requirement 10 (use social information), social structures (such as the social connection between the opinion provider and the truster) are ignored by central systems, and not used as a factor in assessing the reliability of evidence, in the reputation calculations. The evaluation of evidence in centralised systems is largely based on an endogenous approach (for example, (Whitby et al., 2005)), where the opinions supplied about a particular trustee are compared with all opinions provided about that trustee, regardless of the identity of the opinion provider.

We now review the most prominent computational trust models that use the centralised approach.

2.4.1.1 SPORAS

Moukas et al. (1999) propose a reputation mechanism for online communities, with a number of distinct properties. Here, a newcomer to the online community starts with a *minimum reputation value* which is updated as a result of their activity in the system. An important property of an individual's reputation is that the reputation cannot fall to a level below that of a newcomer. This means that an individual never has an incentive to leave the system and re-enter under a new identity. In a system that allows reputation levels of an individual to fall below the level of a newcomer, there is an obvious incentive to leave and re-enter the system. This type of behaviour allows an agent to falsely increase its reputation.

The reputation level of an individual is updated after each transaction by obtaining feedback ratings from the other parties involved in the transaction. This feedback represents the trustworthiness that the other parties place in the individual after the latest transaction. In SPORAS, each individual can only rate another individual once. Thus, when an individual has rated another more than once, the latest rating is used. In SPORAS, the amount an individual's reputation level is increased or decreased is not only dependent on feedback, but also on the current reputation level, so that individuals with high reputation values experience a smaller rating change after each update. This property limits the increase in an individual's reputation level to a high level quickly. This mechanism provides a solution to General Requirement 11 (dynamic trust value).

Finally, the ratings used to calculate reputation are discounted over time, so that recent ratings have more weight. This is a simple and effective mechanism to solve the problems related to trust and reputation, exhibited by the dynamic behaviour of individuals. By increasing the weight of more recent ratings the value of reputation obtained will be a more accurate representation of an individual's recent behaviour.

In the context of this work, however, the SPORAS approach has two main limitations. Firstly, with respect to General Requirement 4 (calculate direct trust), it does not account for a personalised view of reputation. By this we mean that SPORAS does not have a mechanism by which an agent can obtain reputation from the agents that it deems more trustworthy. This *social knowledge* (knowing which agents are more trusted to provide accurate reputation information) is not taken into consideration when calculating trust from aggregated reputation values. Secondly, SPORAS is a centralised system. This approach meets General Requirements 14 (effective exchange of opinions), but fails to satisfy General Requirements 1 (scalable model) and 2 (decentralised model). More specifically, the agents do not have an individual database of their own ratings, since ratings are stored centrally and accessed when needed. This is not an appropriate approach in a dynamic environment, because the network node that houses the central data may be inaccessible from time to time. In such cases if an agent requires ratings from the database, it will not have an alternative source of data for those ratings and the agent will be unable to calculate an effective level of reputation.

2.4.1.2 HISTOS

Moukas et al. (1999) propose HISTOS as a more complex reputation mechanism for an online community. Thus, while SPORAS provides a global reputation value for any individual in the system, HISTOS provides a more personalised reputation. Here a personalised reputation value for an individual is based on the principle that an agent trusts its friends more than strangers. This is a rudimentary solution to meet General Requirement 8 (adjust unreliable opinions), which essentially works by decreasing the risk of recieving bad evidence by selecting good sources (from friends) of reputation information.

In more detail, in HISTOS, the pairwise ratings are represented as a directed graph in which the nodes represent the users, the weighted edges represent the latest reputation value, and the direction of the edge points to the rated user. Using this graph, an individual A_0 can calculate a more personalised reputation level for A_L, if a path exists from A_0 to A_L. In order to establish whether this path exists, the individual queries the system and the system performs a breadth first search to find all paths connecting A_0 to A_L. If the search fails, the reputation level from the SPORAS mechanism is used. If the search is successful, then the rating from the node before A_L is taken and this step is performed recursively at each node found traversing back along the path to A_0.

This mechanism allows SPORAS to find and obtain reputation information, General Requirement 6 (finding reputation), particularly in a highly connected graph (such as in a community), but its use in a very large system is limited, especially where the interactions of individuals may not be so tightly coupled. In a large-scale open system, it would be almost impossible to draw a single global graph where nodes represent agents and the edges represent interactions between them. While many agents will have a local view of the entire system and they may be able to construct a social graph by using this local information, in this context, an unmodified HISTOS algorithm will fail to deliver the desired results.

Another important limitation to the HISTOS approach is that the graph data is held centrally, and the absence of the authority that provides this information may have a catastrophic result. If the graph data was distributed or replicated across a network, then there would be additional problems in maintaining the information so that it was consistent and accurate across all the network nodes that housed it.

2.4.1.3 The Beta Reputation System

In contrast to the other centralised systems discussed, Ismail and Jøsang (2002) propose the Beta Reputation System (BRS) as a Bayesian system, in which ratings are given to individuals based on the quality of their performance as perceived by the rater.

In the BRS an individual gives a rating r in the range [0,100] to another, which translates into a positive rating (r) and a negative rating ($100 - r$). Ratings are stored in a central ratings database and are used as a basis for calculating reputation values. Each time a new rating is reported to the central store, an update rule is used to modify the existing reputation score of the individual the rating applies to. More specifically, the *a posteriori* reputation score for an individual is calculated by taking the *a priori* value and combining it with the new rating, and effective solution for General Requirement 5 (calculate reputation). The reputation score is represented as a pair, of total positive ratings ($\hat{r}$) and total negative ratings ($\hat{s}$), which forms beta probability distribution function (PDF) parameters. A beta PDF is a theoretically sound way in which to calculate the posterior probability of a binary random variable (Karian and Dudewicz, 2000), and the beta PDF expresses the uncertainty in future behaviour of an individual. In the BRS, the reputation is defined to be the expected value of the beta PDF given $\alpha = \hat{r} + 1$ and $\beta = \hat{s} + 1$. While it is true that BRS offers a grounded mechanism to combine ratings to form a single reputation value, it fails to meet General Requirement 4 (calculate direct trust) as it does not allow individuals to locally store their own personal experiences (ratings). In BRS the individuals submit their own ratings to a central repository where an overall reputation value (which is an amalgamation of the individual ratings) is stored. This mechanism allows an individual to query the central store about a particular individual's reputation, but does not allow a query that will list all the ratings that are used in calculating this reputation value.

The ratings that are stored are degraded over time, so that greater emphasis is placed on recent ratings. The BRS satisfies General Requirement 8 (adjust unreliable opinions) by employing a primitive form of filtering ratings to give a greater weighting to reliable ratings. This is achieved by discounting ratings by a weight determined by the overall reputation of the rating provider. It assumes that in a system the entities with a high reputation will always give accurate and good ratings to others. This assumption may be supported in closed systems where there are few incentives for individuals to provide false information. However, in an open system it is important to consider other factors as well as the overall reputation of an individual, in determining whether or not the ratings provided by that individual should carry more weight than others.

In later work, Whitby et al. (2005), build upon this filtering mechanism, and propose a statistical method of filtering unfair (or inaccurate) ratings by comparing all ratings to each other. In this method, the set of ratings about a certain entity is used to determine a mainstream opinion, and any ratings from the set that do not fall within predefined boundaries of this opinion are ignored. This approach has obvious benefits in societies where the majority of individuals provide accurate and fair ratings. However, this approach is very limited and in fact works as a reverse filter (letting the unfair ratings pass) in an environment where the majority of individuals are being deceptive.

2.4.2 Distributed Models of Trust

In contrast to their centralised counterparts, distributed systems have no central authority. The central reputation centre is replaced by several smaller distributed ones, or in the extreme case, each individual records only its own interaction history. In these systems, information is therefore gathered from one or more distributed stores, or directly from several individuals, as shown in Figure 2.5. The biggest problem with the distributed approach is that each time a truster wishes to assess a trustee, the truster must find and gather data from several sources (reputation stores or individuals in the community) and combine it. Therefore a good distributed model needs to address General Requirements 6 (find reputation), 7 (incentives for providing opinions) and 14 (effective exchange of opinions).

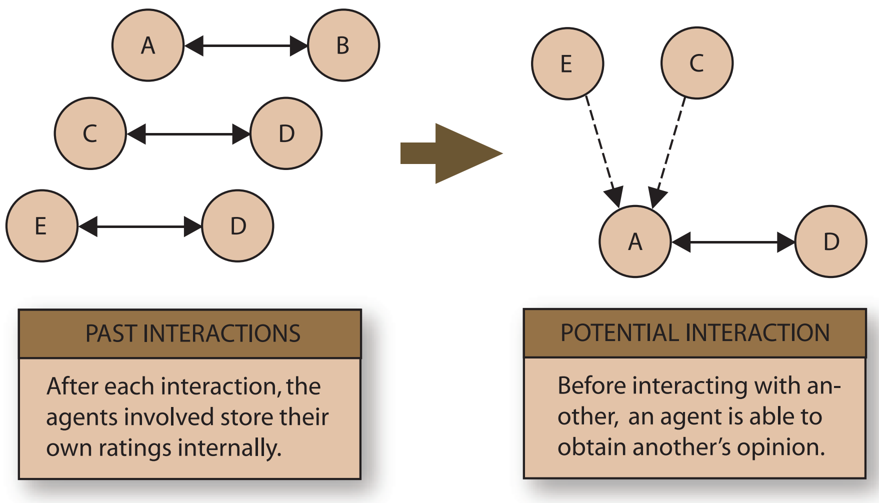

FIGURE 2.5: A decentralised trust system (adapted from Jøsang et al. (2006)).

Jøsang et al. (2006) describe two key components of distributed trust systems:

1. **Distributed communication protocols** — These allow communication between individuals in the community for the purpose of exchanging trust information and opinions. These protocols differ from those required in the centralised approach by the fact that they have to support functionality related to locating the right individual to communicate with. In the centralised approach there is no need to find the location of the central repository, which is fixed and known to all individuals in the system.

2. **Reputation computation method** — This is used by each individual to obtain a single reputation value from a number of opinions it has gathered from others in the community.

The distributed approach is well suited to large open systems and meets General Requirements 1 (scalable model) and 2 (decentralised model). Furthermore, in such large systems it may be computationally expensive to gather opinions from all partners an individual has interacted with. In such environments, the distributed approach allows an individual to obtain a subset of this reputation information from its 'neighborhood', by directly asking a subset of individuals in the community for their opinions. For example, an agent may ask only the members of the VO that it belongs to for their opinions about a certain individual.

In the following sections we describe some prominent distributed computational trust models.

2.4.2.1 Marsh's Trust Model

Marsh (1994) was one of the first to consider the concept of trust from a computational perspective and his model is the first well known computational model of trust. The model has a set of agents A, a variety of situations (α,β,γ,...), and a set of Boolean predicates $K_x(y)^t$ which indicate if agent x knows agent y at time t. Marsh separated trust into three categories:

1. **Basic Trust** – This is the level of trust which represents the general trust disposition of agent $x \in A$ at time t.
2. **General Trust** – Given agents $x, y \in A$, the general trust $Tx(y)^t$ represents the amount of trust that x has in y at time t.
3. **Situational Trust** – Given agents $x, y \in A$, and a situation α, the situational trust $Tx(y, \alpha)^t$ represents the amount of trust that x has in y in situation α at time t.

Trust is represented as a real number between -1 and +1. The model represents the utility gained by an agent x in situation α at time t by $Ux(\alpha)^t$. This utility has a value from -1 to +1. The importance of situation α at time t for agent x is represented by a value between 0 and 1.

In this model, trust can be used by agent x to decide whether or not to cooperate with agent y in situation α. In order to use trust in this context, certain assumptions have to hold: (i) an agent x has a choice whether to cooperate, (ii) there is another agent y to cooperate with, (iii) x and y have met before ($K_x(y)^t$ is true), and (iv) x has knowledge of the situation. If all the conditions hold, then cooperation can occur when the situation trust of x in y exceeds a set threshold. Situation trust is estimated by multiplying the gain in utility in situation α by the importance of situation α and the general trust of x in y. Although the concept of situational trust provides a good solution for General Requirement 16 (context dependent), Marsh's approach has limitations. In particular, two of the components in the calculation can be negative, at a given time, possibly resulting in a positive trust value, making the application of the trust value ambiguous. In addition, the components may be fractions, resulting in a small product, making the end result incomparable.

The estimation of the situational trust is only complete when all possible values of trust of x in y have been considered. The previous values of trust are considered using three approaches: a maximum (optimistic), minimum (pessimistic) and pragmatic (the mean) estimate of the situation trust. Marsh proposes these three different approaches to allow flexibility in the trust estimation process. Due to memory restrictions, the model allows the imposition of a limit on the number of experiences to be taken into consideration in the trust calculation, so very old experiences cannot heavily influence the value of trust obtained. This is a desirable characteristic because if an agent is capable of changing its behaviour dynamically, then a set of recent experiences will best represent the behaviour of the agent.

An important property of Marsh's model is that it incorporates and promotes reciprocation, to a certain degree. Thus, cooperation between agents helps increase their trust in each other, whereas defection reduces the trust. This is a valid abstraction from human society. In our society if we trust someone to complete a specific task, then upon successful completion of the task we may feel obliged to return the favour. Therefore, in the future if they trust us to complete a task, we will be more likely to complete it to their expectation.

Marsh's model meets General Requirement 4 (calculate direct trust), examining many issues relating to direct interaction (experience) based trust; however it is limited with respect to propagation and amalgamation of the trust knowledge within the system, failing to meet General Requirement 5 (calculate reputation). Marsh does not provide mechanisms by which trust information about a particular agent can be gathered from a group of agents. Thus it can be said that his model does not support the spread and collection of reputation information. As stated in Section 2.3.4, we believe reputation information is essential in order to make an accurate trust assessment if one does not have personal experience of interacting with other agents.

2.4.2.2 A Cognitive Trust Model

Castelfranchi and Falcone (1998); Falcone and Castelfranchi (2001) built a model of trust based on a cognitive perspective. They criticise the probabilistic models of trust (such as the Beta Reputation System, described in 2.4.1.3) by claiming that this view only examines the predictability dimension of trust, and that it ignores the "competence" dimension which takes into account the mental attitudes and beliefs of the parties that are interacting. In their model, an agent x trusts an agent y about an action α that results in a world state g. Thus here trust is seen as a mental state or attitude that results in the action of delegation of part of agent $x's$ plan to agent y. The work divides the concept of trust into two distinct components: (i) an internal characteristic of the trustee (internal trust), and (ii) evaluation about the probability and consistence of obstacles, opportunities and other external factors (external trust).

In more detail, internal trust requires the truster to have a conceptual model of the trustee's mind. This requires modelling and reasoning about a complex structure of beliefs and goals. Each belief in this structure must be evaluated to yield the degree of trust, and an estimation of risk to place in the trustee. The beliefs in the mental state of agent x, which are important to determine the amount of trust to place in agent y by agent x, are described below:

1. Competence Belief: a positive evaluation of the trustee, where x believes that y is useful to achieve its goals and that y is capable of carrying out the action that x will delegate to it. The work argues that the notion of trust is made redundant if y is incapable of carrying out the task.

2. Disposition Belief: x should believe that y will actually do what x needs. This is a belief related to the willingness of agent y.

3. Dependence Belief: For x to trust y, x must believe that x needs to delegate the task.
4. Fulfillment Belief: x believes that by the action of trusting and delegating the world state it desires will be achieved.
5. Willingness Belief: x believes that y has decided and intends to do what it has said it will. This requires x to model the mental attitudes and goals of agent y.
6. Persistence Belief: x believes that y is stable enough in its intentions to carry out the task fully. A lower level of trust should be placed in y if x believes that y is unstable.
7. Self-Confidence Belief: x believes that y knows that it is capable of carrying out the given task. This makes y self-confident. This is a necessity as it is useless to trust someone that does not trust themself.

Castelfranchi and Falcone argue that principled trust requires Belief-Desire-Intention (BDI) like agents (Rao and Georgeff). The beliefs that they discuss are all valid in the evaluation of the amount of trust to place in another agent, but most of these beliefs are very difficult to implement. This, in turn, limits the applicability of this model in a real-life application. In particular, the model that they present is the very much akin to that used in human society, but due to its deep roots in cognitive theory, it is very difficult to translate and ground in a computational domain.

2.4.2.3 REGRET

Sabater and Sierra (2001) propose REGRET as a reputation based model for gregarious societies. These societies contain agents that tend to form groups with others of the same kind and enjoy the company of others. Reputation in this gregarious society is defined as the "*opinion or view of one about something*" and it is a concept that is built up over time by directly interacting with the entity or by obtaining information about that entity from others in the society. An important property of reputation is identified in this work. Reputation is described as a multi-facet concept, where a single entity is described as having several different reputation values, each for a different context; providing a good solution to General Requirements 16 (context dependent). For example, a retailer may have a good reputation for selling high quality products, but a low reputation for quality of customer service. The different types of reputation and the manner in which they are combined are defined as the *ontological dimension* of reputation. In addition to this dimension, two other dimensions of reputation are defined. Firstly, experience gained by direct interactions with an entity in the society forms the *individual dimension* (General Requirement 4 — calculate direct trust). Secondly, the experience gained by interacting with the group (within the society) to which the individual belongs forms the *social dimension* (General Requirement 5 — calculate reputation). Subjectivity in the reputation formed by an agent regarding another is a result of the fact that each agent has its own ontological structure, which identifies the importance (weighting) of all the different types of reputation.

A dialogue between agents in REGRET is represented by an *outcome*, which is the initial contract outlining the terms and conditions for a transaction between two parties and the actual result of the transaction. These outcomes contain two types of variables. Firstly, they contain common variables, which represent the attributes of the transaction that are known to both parties and to which both parties have agreed. Secondly, an outcome may also contain expected variables, which implicitly represent parts of the transaction that are assumed to be completed by one of the parties. The expected variables are related to the subjectivity of the agent, and as a result a single transaction will form a different outcome for each of the agents involved. The subjective evaluation made by an agent on a certain aspect of the outcome is called the *impression*, which is a tuple *(a, b, o,*φ*, t, W)* containing identifiers for the agents involved in the transaction (a, b), the outcome o relating to the transaction, a variable φ from the outcome that is being judged, the time the impression was recorded (t), and a rating $W \in$[-1,1] associated with the attribute being evaluated from the evaluating agent's point of view. The rating represents the subjective evaluation of the agent evaluating with respect to a specific variable in the outcome. REGRET meets General Requirement 14 (effective exchange of opinions), as all the impressions are stored in an *impressions database*, and this data structure is used to evaluate the reputation of others.

The *subjective reputation* of an agent with respect to a given outcome attribute is obtained by obtaining a subset, which matches a given pattern, from the set of impressions stored in the impressions database. The actual value is calculated by a weighted mean of the impression's rating factors in the subset found as a result of the query. Here the more recent impressions are given more relevance on the end reputation value. Furthermore, the reliability of the reputation value calculated is obtained from considering the combination of the number of impressions used in the calculations and the variance of the rating values in the impressions used (General Requirement 8 — adjust unreliable opinions). However, this approach fails to address the issue of strategic lying due to the assumption that there is an altruistic society. Furthermore, the model is highly susceptible to noise as a result of the manner in which the impressions are weighted and summed.

In the case of this work, REGRET can be seen to successfully deal with many of the issues of trust and reputation in virtual communities, and its strengths lie in the compositional definition of reputation that it uses as a basis for the model.

2.4.2.4 An Evidential Model of Distributed Reputation Management

Yu and Singh (2002) present a reputation-based model that utilises the Dempster-Shafter theory (Shafer, 1976) of evidence as the underlying computational framework. The main focus of their work is to address three main issues.

Firstly, they address the problem of how an agent can rate another based on direct interactions by capturing the ratings of the past interactions and recoding them in the given agent's history.

Secondly, they tackle the issue of how an agent is to find the witnesses that will supply it with reputation information (General Requirement 6 — finding reputation information). The proposed solution to this involves a process of *referrals*, which help agents locate witnesses in the society. An agent can use a referral to point another agent to other sources of information that it is aware of, especially in the case where the agent providing the referral has no information of use. They then go onto show that an agent is capable of building up a model of the social network from its neighbours and the gathered referrals (Yu and Singh, 2003). This social network is then used to obtain the concept of a *group* (much like the work discussed in Section 2.4.2.3), which is identified as a group of nodes that are close in the social network. The identification of a group allows an agent to consult other agents in its own group for reputation information or referrals, believing that agents in its group (nearer in the social network) are more reliable than those that are outside the group (further away in the social network).

Finally, Yu and Singh present the *TrustNet* representation, which allows agents to systematically incorporate the testimonies of the witnesses (General Requirement 5 — calculate reputation).

In the context of this work, the greatest strength of Yu and Singh's model is its ability to cope with the absence of trust information, which is a drawback of other work in the field (see Section 2.4.2.3). The use of the Dempster-Shafter theory allows for the combination of beliefs that state that an agent is trustworthy, untrustworthy or unknown. The special case where there is no information, and the nature of an agent is unknown, is considered as a state of uncertainty where each belief is equally likely.

2.4.2.5 FIRE

The REGRET model (Section 2.4.2.3) introduced the notion of reputation as a compositional value. It identified three dimensions of reputation: ontological, social and individual. The greatest strength of this approach is that in the absence of personal experience an agent can obtain information from others (witnesses) in the society. In the absence of any witnesses in the society, an agent can calculate a level of trust using the role-based relationships that exist between agents. However this approach reaches its limitation when the assumption of the availability of role information is removed. This limitation is addressed by Huynh et al. in the FIRE model (Huynh et al. (2006) and Huynh (2006)) that incorporates interaction trust, role-based trust and witness reputation. In addition to these different types of trust (formed from different sources of information) this model proposes *certified reputation* as another source of trust information and a solution to the above limitation.

In more detail, certified reputation is formed by using the ratings that an agent provides by itself. For example, suppose agent A is trying to evaluate the trustworthiness of agent B. In the absence of any information that A can use in its trust calculation, it asks B to provide ratings from its previous experiences (much like asking for a reference letter when applying for a job). In response, B provides A with a set of certified ratings, which it has gathered from asking others to evaluate its performance at the end of an interaction. This means that A can ask B to provide it with ratings of B's past activities without having to search a large social network or consult other agents that have interacted with B previously. The ratings that B provides are certified by the agents that gave that particular evaluation of B's performance. The model assumes that a security mechanism is present that prevents agents from tampering with these certified ratings.

In the context of our work, FIRE addresses some of the limitations with searching social networks for agents that can provide reputation information about a certain agent and with identifying which agents have interacted with the agent for whom a level of trust is being calculated. However the certified reputation has to be treated with some doubt, since agents provide the raw information to others for the calculation of their own trustworthiness. There is, therefore, a large incentive for an agent to provide false information to enhance its trustworthiness, and in open dynamic systems an agent has to be able to cope with this false information.

2.4.2.6 CREDIT

Ramchurn (2004) presents a model, called CREDIT, of trust that differs from REGRET and FIRE in the manner in which it arrives at a trust level for a particular agent. Specifically, CREDIT equips an agent with the ability to assess the trustworthiness of an agent using two types of evidence (similar to those found in REGRET and FIRE): using direct interactions and using reputation. However, CREDIT differs in the mechanism used to translate the evidence to a trust value, and how this trust value is subsequently used. In particular, the CREDIT model uses fuzzy sets to model trust levels that are used by agents to assess their partners with respect to agreed contracts.

Furthermore, this model incorporates the fact that agents exist in electronic institutions, and therefore, the agents' interactions are governed by the norms and conventions of that institution (Ramchurn et al., 2004b). This results in a key feature, which no other trust models provide: distinguishing between the performance of the agent and the environment in which the agent is situated (General Requirement 3 — distinguish between entity and environment). The majority of models take the stance that from the viewpoint of the agent that is calculating a trust value for an interaction partner, it does not matter if the interactions with that partner fail due to the partner's behaviour or due to the environment from which it is operating. However, the CREDIT model offers a good solution to this problem. Here, the agent is able to distinguish the source of the failure by examining the norms and rules that define the environment, and the norms and rules that guide the behaviour of agents. For example, if an agent is working from a faulty network, then CREDIT is able to distinguish between the agent's performance and the faulty network by examining the norms that define that environment, and recognising that in this particular environment it is the norm to have a faulty connection.

2.4.2.7 Incentives for Agents

Some of the models reviewed above have mechanism with which an agent is able to obtain opinions from others (opinion providers) and calculate a reputation value for an individual agent. However, all the models that have this functionality fail to specify *why* opinion providers would supply opinions to allow a an agent to calculate a reputation value.

Against this background, Jurca and Faltings (2003) propose an *incentive compatible* mechanism, saying that without a side payment scheme (or incentive scheme) an agent will be indifferent between providing and not providing an opinion (this satisfies General Requirement 7 — incentives for providing opinions). More specifically, they claim that without such a scheme an agent is indifferent between providing true and false opinions. Their mechanism employs special *R-Agents*, which are broker agents that buy and sell reputation information from others at a given price (it is this buying and selling price that offers agents an incentive to exchange reputation information).

2.5 Summary

In summary, we have seen that issues of deception and guarantees in interaction, and of risk and confidence, are significant when interactions take place with new partners. These issues are further compounded in large scale open environments such as the Grid. In their description of the Grid architecture (Section 2.3.1), Foster et al. describe various VO based scenarios, each of which is summarised as a coming together of "*mutually distrustful participants with varying degrees of prior relationship*" in order to perform a given task. This description of a VO confirms the need for a model of trust that assures good interactions between the entities and their mutually distrustful partners with a VO.

In light of the literature reviewe, we present this summary in two parts. First, we summarise the various definitions of trust encountered, into the one that we use in this research. Second, we provide a summary of how existing state of the art models address some of the requirements given in Section 2.3, highlighting those that we aim to address.

2.5.1 A View of Trust

Having reviewed the literature, we view trust as *modular* concept. Our view is shown in Figure 2.6, which consists of distinct parts that come together to determine a single trust value for a single agent. In more detail, we define, at a high-level, two sources that provide evidence (trust information) that is used in trust calculations: *personal* and *social* evidence.

Personal evidence comprises all information experienced *first hand* by the individual that is performing the trust calculation, and therefore, the main information source is the individual itself. This type of evidence leads to forming the *direct trust* component of an individual's trust value, as shown in Figure 2.6.

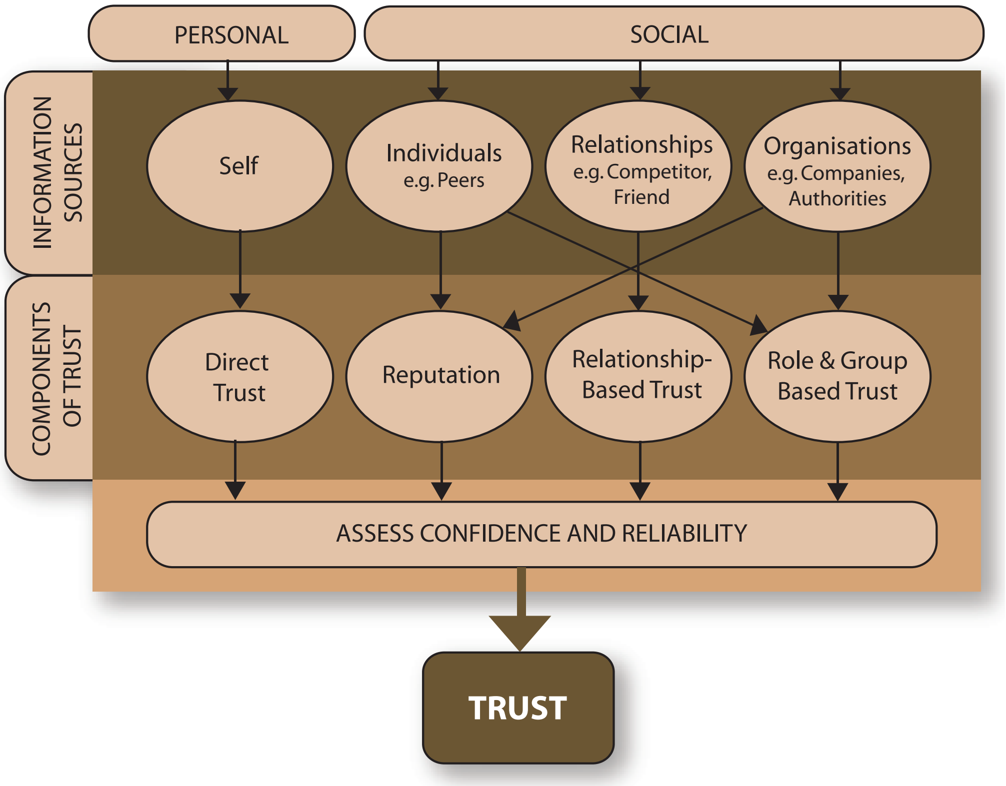

FIGURE 2.6: The modular view of trust.

Social evidence describes all the information obtained from the society, the sources of which can be categorised as follows:

1. **Individuals** – In a society there are many entities from whom one can obtain trust information about a particular individual. Mainly this information is in the form of opinions, which contribute to the *reputation* component of an individual's trust value.

2. **Relationships** – The relationships that individuals share in a society have an impact on the trust values of certain individuals. We define this impact as a component (the *relationship-based trust* component) of the overall trust value for an individual.

3. **Organisations** – There are many organisations about which trust-based generalisations can be made. Furthermore, in a society there are organisations with defined roles and expectations of individuals in those roles. Knowing the roles, expectations and organisation in a society may lead an entity to adjust, in a particular way, an individual's trust value. We encapsulate the impact of organisations, on the final trust value, in a component called *role and group-based trust*.

Finally, as we believe that trust is an amalgamation and a function of relevant evidence, it is important to assess the confidence (in the case of personal evidence) and reliability (in the case of social evidence) of that evidence.

2.5.2 Model Requirements

In Section 2.3 we presented a review of the domain in which we aim to develop a trust model, and through the analysis we developed a number of requirements. More specifically, these requirements were formed from the analysis of the different stages of the VO life cycle and the Grid architecture in general. Here, we present a short summary of how some of these requirement have been addressed by existing models. For those that have not been addressed we state whether this research is concerned with tackling them or justify the reasons for not examining them in more detail.

General Requirement 1 (scalable model) — The distributed approach used by systems described in Section 2.4.2 is scalable. To this end, we do not explore other methods of achieving scalability, and simply adopt a distributed approach when designing the trust model.

General Requirement 2 (decentralised model) — Decentralised models (as discussed in Section 2.4.2) store the information used in trust calculations in a distributed manner. In most cases, each agent using the model keeps its own store of evidence for trust calculations. This is the approach we will take to ensure that our model is robust in the face of network problems.

General Requirement 3 (distinguish between entity and environment) — CREDIT (see Section 2.4.2.6) is the only model we have encountered that addresses the requirement of distinguishing an agent's behaviour from the performance of the environment it is situated in, and its proposed solution is satisfactory. To solve this solution a trust model has to operate within a virtual society which supports norms and electronic institutions (Esteva et al., 2001). To this end we do not explore further, this aspect of a trust model.

General Requirement 4 (calculating direct trust) — REGRET (discussed in Section 2.4.2.3) and FIRE (as discussed in Section 2.4.2.5) provide methods of using previous experiences with an agent, to calculate a trust level in that agent. In our model, we take inspiration from their approach, but unlike their underlying mechanism we aim to evaluate trust in a probabilistic manner (much like the BRS, as discussed in Section 2.4.1.3). The reason for basing the calculation of trust on a probabilistic foundation is that probability theory is well developed and will make a stronger mathematical model of trust than an arbitrary approach.

General Requirement 5 (calculating reputation) — Again, we take inspiration from the REGRET and FIRE models, but aim to provide this functionality with a probabilistic mechanism.

General Requirement 6 (finding reputation) — With regards to finding others from whom to obtain opinions about a particular agent, Yu and Singh (2002) and Sabater and Sierra (2001) offer methods of obtaining reputation information through use of social networks. Furthermore, FIRE provides an effective solution by using certified reputation to locate trust information. For these reasons, we do not focus on this aspect of a trust model.

General Requirement 7 (incentives for providing opinions) — Jurca and Faltings (2003) (as discussed in Section 2.4.2.7) describe an effective solution to incentivise agents to provide their opinion upon request. We recognise that this is critical to systems that rely on reputation information, and, as such, our model should provide such incentives. However, the solutions provided are sufficient to be embedded into our model with little work, and therefore the thesis will not concentrate on developing yet another solution.

General Requirement 8 (adjust unreliable opinions) — There are certain limitations to approaches in the literature (Section 2.4.2.3 and 2.4.1.3) that aim to minimise the affect of misleading (or inaccurate) opinions on the trust value that an individual calculates. We wish to explore this area further and, therefore, the research aims to provide a more effective solution to this problem than has hitherto been developed.

General Requirement 9 (maintaining interaction history) — Every trust model reviewed has a unique way of recording interaction history, because each makes use of this information in different ways. Our model will not adopt any of the other approaches, but we will aim to record and maintain an interaction history in a manner that allows it to be easily exchanged (for opinions) and searched (for trust calculations).

General Requirement 10 (use social information) — Contemporary trust models (see Sections 2.4.2.3, 2.4.2.4 and 2.4.2.5) state that social structures are important in trust calculation. However all the approaches in the literature are limited by the fact that they assume social information is available, and they do not specify the exact impact it has on a trust value. We believe this is a significant gap in the research and choose to concentrate on addressing this limitation of contemporary trust models.

General Requirement 11 (dynamic trust value) — Both REGRET and FIRE have mechanisms that allow an agent to change its trust level in another when more evidence is observed. Again, we will use their approaches as inspiration, but we will develop a probabilistic mechanism that provides the same functionality.

General Requirements 12 (trust level concensus) and 13 (VO-level trust) — Currently, no trust models provide a solution that addresses these requirements. While the requirement for VO members to reach consensus about an individual's trust value and to obtain a single trust value for an entire VO are important, we see them as secondary requirements. By this we mean that once a basic VO trust model (like the one we aim to develop) is developed, then these higher level requirements can be addressed. Thus, we do not address these issues in this research.

General Requirement 14 (effective exchange of opinions) — As per General Requirement 9, we will aim to store and maintain trust information in a way that is easily exchanged as opinions.

General Requirement 15 (distributed trust information) — As per General Requirements 1 and 2, we choose to adopt the distributed approach (described in Section 2.4.2) for developing our trust model.

General Requirement 16 (context dependent) — Many models categorise trust evidence (like in REGRET and FIRE) allowing an agent to use only the trust evidence that is applicable to the context in which it finds itself. To this end, we do not concentrate on this requirement, but we consider its implications when designing the model, ensuring that evidence is stored in a way that allows it to be categorised by the context in which it was observed.

In the following chapters we present our work. More specifically, in Chapter 3 we present a basic computational trust model that meets some of our aims, and satisfies General Requirements 1, 2, 4, 5, 8, 9, 11, 14, 15 and 16. Furthermore, in Chapter 5 we extend our model so that it satisfies General Requirement 10, by incorporating social information in trust calculations. Finally, in Chapter 7 we suggest ways in which our approach can be extended to satisfy General Requirements 12 and 13.

Chapter 3

TRAVOS: A Trust and Reputation Model for Agent-Based Virtual Organisations

In Chapter 1 we argue that the concept of trust is key to enabling entities in complex and dynamic computer systems to account for the uncertainty in their decision-making processes, and so we aim to develop a computational model of trust that an individual (truster) can use to arrive at a trust level for another (trustee). In doing so, the truster will be able to account for the uncertainty regarding the actions of the trustee. More specifically, we aim at creating this model to determine a trust level in a trustee by considering the behaviour of the trustee in previous episodes (Aim 1), and opinions given to the truster by other agents about the trustee (Aim 2). Furthermore, we know that using opinions adds another source of uncertainty in the trust calculation and so we aim to create mechanisms to lower this uncertainty by adjusting the opinions prior to using them in the trust calculation (Aim 3).

To this end, in this chapter we present a *basic* computational trust model for agent-based virtual organisations (VOs). In later chapters (Chapter 5) we expand this basic model with respect to our initial aims, and incorporate the use of social information in determining a trust level (Aim 4). In more detail, in this chapter we describe TRAVOS (**T**rust and **R**eputation model for **A**gent-based **V**irtual **O**rganisation**S**), a model of trust that takes a probabilistic view of trust, and contains mechanisms to obtain a trust value based on past interaction and from reputation information obtained from peer agents.

TRAVOS is a model of trust and reputation that can be used to support informed decision-making to assure good interactions in a Grid environment that supports virtual organisations. We argue that the use of trust in decision-making can assure good interactions by enabling an agent to reason whether or not it should interact with a potential partner. For example, if a group of agents are to form a VO, then it is important for them to choose the most appropriate partners. Here, the choice not only factors the capabilities of the partners, but also the partner's trustworthiness. Forming a VO with trustworthy partners is better than forming a VO with untrustworthy partners. TRAVOS equips an agent (the truster) with three methods for assessing the trustworthiness of another agent (the trustee).

- First, the truster can make the assessment based on the direct interactions it has had with the trustee.
- Second, the truster can assess the trustworthiness of the trustee based on the opinions provided by others in the system
- Third, the truster can assess the trustworthiness of another based on a combination of the direct interactions with and the reputation of the trustee.

The remainder of this chapter provides a detailed description at how TRAVOS enables an agent to achieve the above. We begin by describing how we model one agent's behaviour towards another, and then we introduce the basic notation that is used throughout the chapter. Having established the notation to describe the mechanisms in the model, we move onto describing how a truster is able to use its past experiences with a trustee to determine a trust level for the trustee (Section 3.3). Knowing that in large systems it is likely that an agent may encounter another for the first time, in Section 3.4, we present a mechanism that allows a truster to gather opinions from other agents to determine the trustee's reputation. In particular, within this section, we describe how an agent is able to handle incorrect or misleading opinions provided by others (Section 3.4.2). Following this, in Section 3.5, we present a mechanism that allows an agent to combine direct trust and reputation, enabling it to make use of both types of evidence (personal and social). Finally, to allow an agent to efficiently combine the two types of evidence, we describe a mechanism that allows an agent to determine the confidence it has in the evidence (Section 3.6).

3.1 Agent Behaviour

Gambetta (1988) defines trust as being a measure that represents the probability of an agent carrying out a particular action. Trust is, therefore, an indication of the reliability of an agent. For this reason, it is important to be able to represent and model the behaviour of an agent before developing mechanism that allow an agent to determine a level of trust.

Often agents are configured to behave in a particular way by their designers, and it is likely that agents (with different behaviours) from a number of different designers will be present in an open multi-agent system. Typically, these behaviours are complex, and dictate the exact manner in which the agent operates in the virtual world. For simplicity, in our work, we reduce this complex behaviours to a simple behaviour, which we can use in our model. More specifically, TRAVOS considers the behaviour of an agent as a probability that it will participate in a successful interaction (trustworthy behaviour) and a probability that it will perform an unsuccessful interaction (untrustworthy behaviour). This abstraction of agent behaviour means that in our model the outcome of an interaction is a binary value (successful or not).

A successful interaction is one where all the mutual (explicitly represented in a contract) and hidden (not stated in the contract) expectations of the interacting parties are satisfied. For example, if a truster and a trustee agree that the trustee will provide a movie service every day for a month at a minimum frame rate of X, this agreement is recorded in a contract held by the truster and the trustee. This interaction is deemed successful if and only if the service was provided as specified in the contract. If the trustee fails to deliver on any day or below the minimum frame rate, then the outcome is unsuccessful, and has a negative impact on the trustworthiness of the trustee. Again, for simplicity, the agreement to interact between two individuals is summarised as a contract which, in our model, is defined as a set of service attribute and value pairs, held between a consumer agent (truster) and a supplier agent (trustee). This set of tuples represents the agreement between the truster and the trustee of the level of service that is to be provided.

Having described how we model the behaviour of an agent, we now present the notation which we use to describe the mechanisms in TRAVOS. In addition outlining the notation we use, the next section also presents basic definitions that are used throughout the chapter.

3.2 Basic Notation

We model the environment in which TRAVOS is applied as a multi-agent system consisting of n agents, and we denote the set of all agents as $\mathcal{A} = \{a_1, a_2, ..., a_n\}$. Over time, distinct pairs of agents $\{a_x, a_y\} \subseteq \mathcal{A}$ may interact with one another, governed by contracts that specify the obligations of each agent towards its interaction partner. An interaction between a_1 and a_2 is considered successful by a_1 if a_2 fulfils its obligations. From the perspective of a_1, the outcome of an interaction between a_1 and a_2 is summarised by a binary variable, O_{a_1,a_2}, where $O_{a_1,a_2} = 1$ indicates a successful (and $O_{a_1,a_2} = 0$ indicates an unsuccessful) interaction[1] for a_1 with a_2 (Equation 3.1). Furthermore, we denote an outcome observed at time t as $O^t_{a_1,a_2}$, and the set of all outcomes observed from time t_0 to time t as $O^{t_0:t}_{a_1,a_2}$:

$$O_{a_1,a_2} = \begin{cases} 1 & \text{if contract fulfilled by } a_2 \\ 0 & \text{otherwise} \end{cases} \tag{3.1}$$

At any point of time t, the history of interactions between agents a_1 and a_2 is recorded as a tuple, $\mathcal{R}^t_{a_1,a_2} = (m^t_{a_1,a_2}, n^t_{a_1,a_2})$ where the value of $m^t_{a_1,a_2}$ is the number of successful interactions of a_1 with a_2 up to time t, while $n^t_{a_1,a_2}$ is the number of unsuccessful interactions of a_1 with a_2 up to time t.

[1]The outcome of an interaction from the perspective of one agent is not necessarily the same as from the perspective of its interaction partner. Thus, it is possible that $O_{a_1,a_2} \neq O_{a_2,a_1}$.

The tendency of an agent a_2 to fulfil or default on its obligations to an agent a_1, is governed by its behaviour. We model the behaviour of a_2 towards a_1, denoted B_{a_1,a_2}, as the *intrinsic* probability with which $O_{a_1,a_2} = 1$. In other words, B_{a_1,a_2} is the *expected value* of O_{a_1,a_2} given complete information about a_2's decision processes and all environmental factors that affect its capabilities (Equation 3.2):

$$B_{a_1,a_2} = E[O_{a_1,a_2}], \quad \text{where } B_{a_1,a_2} \in [0,1] \tag{3.2}$$

In TRAVOS, each agent maintains a *level of trust* in each of the other agents in the system. The level of trust of an agent a_1 in an agent a_2, denoted as τ_{a_1,a_2}, represents a_1's assessment of the likelihood of a_2 fulfilling its obligations. Specifically, the level of trust calculated using only an agent's own interactions with another is known as *direct trust* and is denoted by $\tau^d_{a_1,a_2}$. On the other hand, the level of trust calculated using only opinions provided by others is known as *reputation* and is denoted by $\tau^r_{a_1,a_2}$. The trust calculated from combining personal experience with opinions provided from others is known as *combined trust* and is denoted by $\tau^c_{a_1,a_2}$. The *confidence* of a_1 in its assessment of a_2 is denoted as γ_{a_1,a_2}. Confidence is a metric that represents the accuracy of the trust value calculated by an agent given the number of observations (the evidence) it uses in the trust value calculation. Intuitively, more evidence would result in more confidence.

Having presented the notation that we use to describe the different components of the model, in the following section, we present TRAVOS in detail. We begin by describing the mechanism that allows a truster to determine the direct trust in a trustee. Following this, we show how a truster can obtain and use opinions from others to determine a trustee's reputation, and how this can be combined with direct trust to form a combined trust value.

3.3 Modelling Direct Trust

The first basic requirement of a computational trust model is that it should provide a metric for comparing the relative trustworthiness of different agents. From our definition of trust (Section 2.5.1), we consider an agent to be trustworthy if it has a high probability of performing a particular action which, in our context, is to fulfil its obligations during an interaction. This probability is unavoidably subjective, because it can only be assessed from the individual viewpoint of the truster, based on the truster's personal experiences.

In light of this, we have adopted a probabilistic approach to modelling direct trust, based on the individual experiences of any agent in the role of a truster. If a truster, agent a_1, has complete information about a trustee, agent a_2, then, according to a_1, the probability that a_2 fulfils its obligations is expressed by B_{a_1,a_2}. In general, however, complete information cannot be assumed; the best we can do is to use the expected value of B_{a_1,a_2} given the experience of a_1, which we consider to be the set of all interaction outcomes it has observed. Thus, we define the level of direct trust $\tau^d_{a_1,a_2}$ at time t, as the expected value of B_{a_1,a_2} given the set of outcomes $O^{1:t}_{a_1,a_2}$ (see Equation 3.3):

$$\tau^d_{a_1,a_2} = E[B_{a_1,a_2}|O^{1:t}_{a_1,a_2}] \tag{3.3}$$

In the following section we describe a statistical method that can be used in calculating the expected value of continous random variables, such as B_{a_1,a_2}. Simply, the statistical method involves keeping a count of all successful and unsuccessful interactions with an individual, which allows us to calculate a trust value (the proportion of successful interactions from the total number of interactions) for that individual.

3.3.1 The Beta Distribution

The expected value of a continuous random variable is dependent on the *probability density function* (PDF) used to model the probability that the variable will have a certain value. Thus, we must choose such a function that is suitable to our domain.

In *Bayesian* analysis, the *beta* family of PDFs is commonly used as a prior distribution for random variables that take on continuous values in the interval $[0, 1]$. For example beta PDFs can be used to model the distribution of a random variable representing the unknown probability of a binary event (DeGroot and Schervish, 2002) – B is an example of such a variable. For this reason, we use beta PDFs in our model. (Beta PDFs have also previously been applied to trust for similar reasons, see Section 2.4.1.3).

The general formula for beta distributions is given in Equation 3.4. It has two parameters, α and β, which define the shape of the density function when plotted. Examples plotted for B with various parameter settings are shown in Figure 3.1; here, the horizontal axis represents the possible values of B, while the vertical axis gives the *relative* likelihood that each of these values is the true value for B. The most likely value of B is the curve maximum. The width of the curve represents the amount of uncertainty over the true value of B. If α and β both have values close to 1, a wide density plot results, thus representing a high level of uncertainty about B. In the extreme case of $\alpha = \beta = 1$, the distribution is uniform, with all values of B considered equally likely; this is shown in Figure 3.2.

$$f(b|\alpha, \beta) = \frac{b^{\alpha-1}(1-b)^{\beta-1}}{\int U^{\alpha-1}(1-U)^{\beta-1}dU}, \quad \text{where } \alpha, \beta > 0 \tag{3.4}$$

3.3.2 Calculating Direct Trust

Against this background, we now show how to calculate the value of the direct trust, $\tau^d_{a_1,a_2}$, based on the interaction outcomes observed by a_1. First, we must find values for α and β

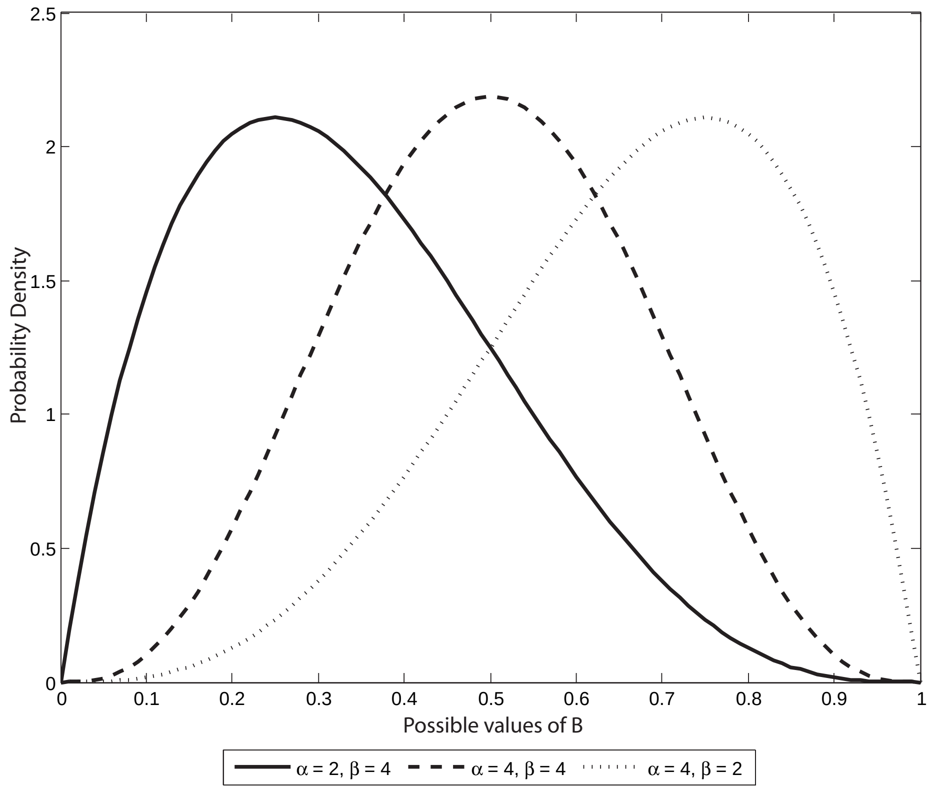

FIGURE 3.1: Three example beta plots with different parameter settings.

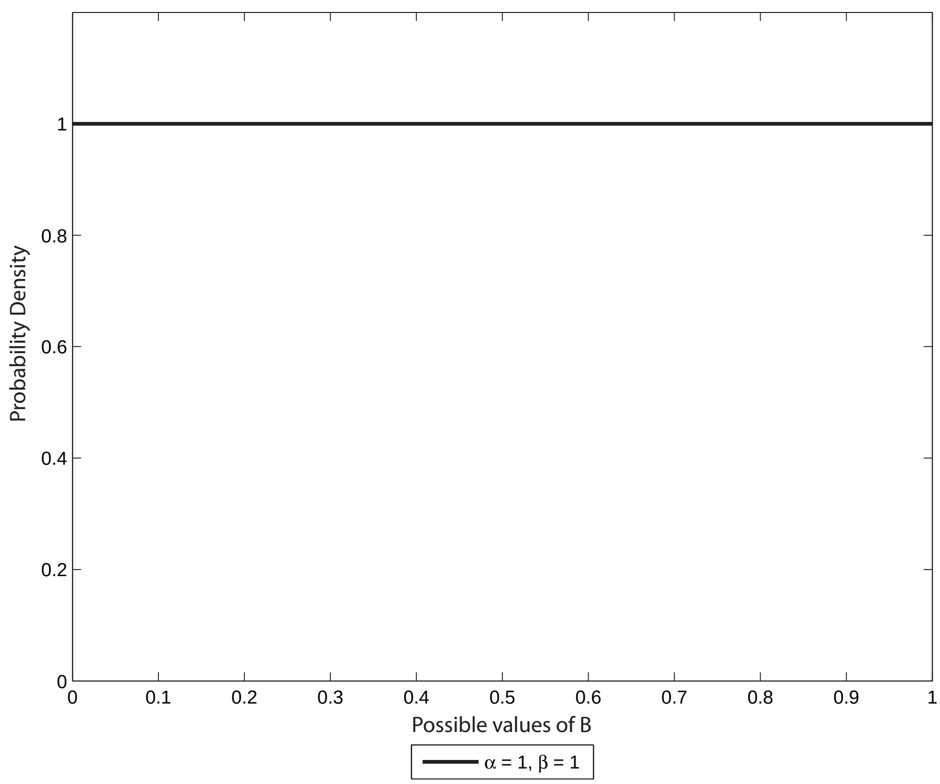

FIGURE 3.2: A special case of a beta curve resulting in a uniform distribution.

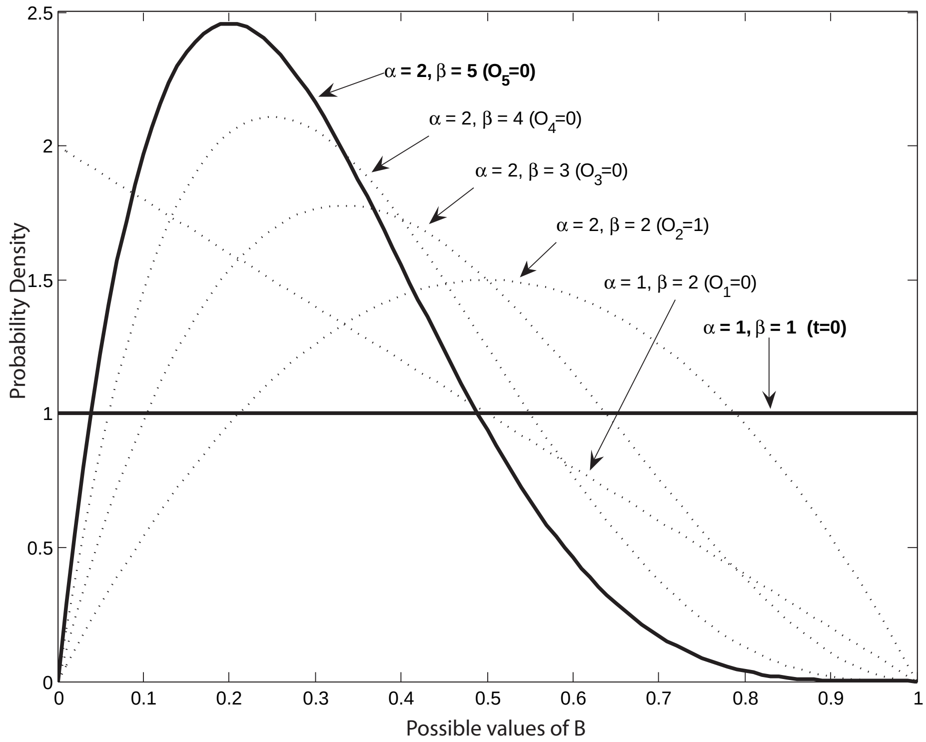

FIGURE 3.3: Changes to the beta distribution (representing the expected behaviour of a trustee) as the truster observes five outcomes from time t_0 to t_5.

that represent the beliefs of a_1 about a_2. Assuming that, prior to observing any interaction outcomes with a_2, a_1 believes that all possible values for B_{a_1,a_2} are equally likely, then a_1's initial settings for α and β are $\alpha = \beta = 1$. Based on standard techniques (Karian and Dudewicz, 2000), the parameter settings in light of observations[2] are achieved by adding the number of successful outcomes to the initial setting of α, and the number of unsuccessful outcomes to β. In our notation, this is given in Equation 3.5. Then the final value for $\tau^d_{a_1,a_2}$ is calculated by applying the standard equation for the expected value of a beta distribution (Equation 3.6) to these parameter settings. Figure 3.3 shows how $\tau^d_{a_1,a_2}$ and the distribution changes as the agent observes the outcomes of interactions (gains experience) and modifies the beta distribution accordingly.

$$\hat{\alpha} = m^{1:t}_{a_1,a_2} + 1 \quad \text{and} \quad \hat{\beta} = n^{1:t}_{a_1,a_2} + 1 \quad \text{where } t \text{ is the time of assessment} \tag{3.5}$$

$$\tau^d_{a_1,a_2} = \frac{\hat{\alpha}}{\hat{\alpha} + \hat{\beta}} \tag{3.6}$$

Having described the method of calculating the level of direct trust a truster a_1 is willing to place in a trustee a_2, we can now create a *Direct Trust agent* (DTA), which has the ability to

[2]We are using the hat notation ($\hat{\alpha}$ and $\hat{\beta}$) to indicate that the values for the beta distribution α and β parameters are estimates based on observed evidence.

assess trustees purely on the direct trust it has in them. Therefore, the overall trust a_1 has in a_2 is equal to the direct trust it has in a_2, (τ_{a_1,a_2} = $\tau^d_{a_1,a_2}$). The DTA has the advantage of knowing that the information used to calculate the direct trust value is accurate and true, as it was the one that made the observations. In this case there is no uncertainty introduced by using information provided by others.[3] The limitation of the DTA is that in the absence of personal experience, the agent is unable to calculate a value of trust. In response to this limitation, in the next section we present a mechanism that allows a truster to calculate a trust level for a trustee using opinions provided by others in the community.

3.4 Modelling Reputation

When assessing a trustee, the most reliable evidence for predicting the behaviour is personal experience of the opponent's past behaviour. Unfortunately, it will often be the case that the assessing agent will have limited or no experience of a potential interaction partner.

Reputation is therefore cited as a useful means of gathering evidence. It involves asking for the opinion of other parties who have interacted with the trustee in the past. We have already shown that an agent a_3 is capable of calculating levels of direct trust in a trustee a_2 based on its interaction history with that entity. The interaction history of a_3 with a_2 at time t can be regarded as a tuple, $\mathcal{R}^t_{a_3,a_2} = (m^t_{a_3,a_2}, n^t_{a_3,a_2})$, defined in Section 3.2. Similarly, we define a_3's opinion about a_2 as a tuple, $\hat{\mathcal{R}}^t_{a_3,a_2} = (\hat{m}^t_{a_3,a_2}, \hat{n}^t_{a_3,a_2})$, which represents the history of interaction between a_3 and a_2 as *reported* by a_3 at time t to an agent a_1 (who requested a_3's opinion). It is important to note that in general $\mathcal{R}^t_{a_3,a_2} \neq \hat{\mathcal{R}}^t_{a_3,a_2}$ because the opinion provider may have an incentive to lie and may exaggerate the true interaction history. In the special case where the opinion provider is being completely honest, $\mathcal{R}^t_{a_3,a_2} = \hat{\mathcal{R}}^t_{a_3,a_2}$.

3.4.1 Combining Opinions

The truster, a_1, must calculate a single reputation value, $\tau^r_{a_1,a_2}$ for a trustee a_2 by combining all the opinions provided by others. An elegant and efficient solution to this problem is to enumerate all the successful and unsuccessful interactions from the reports that it receives (see Equation 3.7). The resulting values, denoted N_{a_1,a_2} and M_{a_1,a_2} can then be used to calculate shape parameters (see Equation 3.8) for a beta distribution. The reputation value $\tau^r_{a_1,a_2}$ is calculated by using these parameter values in Equation 3.9. Figure 3.4 shows the beta plots for three example opinions provided by three separate agents, and the beta plot from the combination of these opinions. The figure clearly shows that when three separate opinion distributions are considered the resulting combined distribution has less variance, meaning that the agent can be

[3]The uncertainty in other information arises from the fact that others may have an incentive to provide false information to influence the value of trust calculated for a certain trustee.

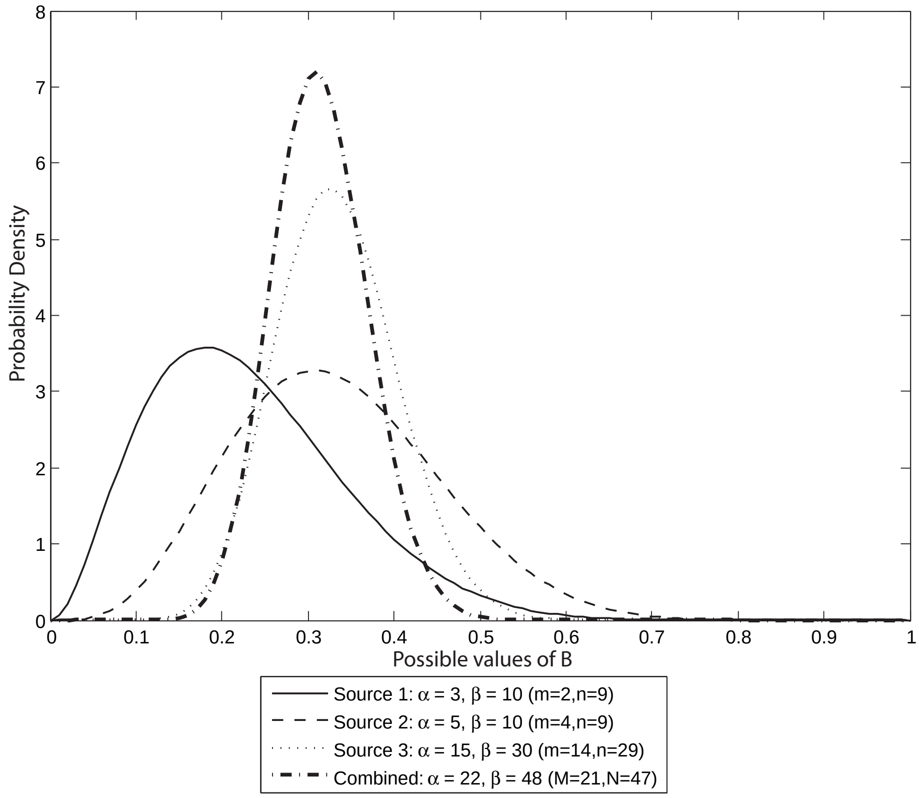

FIGURE 3.4: Three separate opinions and the reputation that is calculated once the three opinions are combined.

more confident in the trust value obtained from the combined distribution.

$$N_{a_1,a_2} = \sum_{k=0}^{p} \hat{n}_{a_k,a_2}, \quad M_{a_1,a_2} = \sum_{k=0}^{p} \hat{m}_{a_k,a_2}, \quad \text{where } p = \text{number of reports} \tag{3.7}$$

$$\hat{\alpha} = M_{a_1,a_2} + 1 \quad \text{and} \quad \hat{\beta} = N_{a_1,a_2} + 1 \tag{3.8}$$

$$\tau^r_{a_1,a_2} = \frac{\hat{\alpha}}{\hat{\alpha} + \hat{\beta}} \tag{3.9}$$

Having shown how the reputation of a trustee can be calculated, we can now create a *Reputation Trust agent* (RTA), which has the ability to assess trustees purely on their reputation. Thus, the overall trust that a_1 has in a_2 is equal to the reputation of a_2 ($\tau_{a_1,a_2} = \tau^r_{a_1,a_2}$).

There are two desirable features of this approach. Firstly, this addresses the problem of not having any information to perform a calculation of trust, which is exhibited by the DTA. For the DTA, having no interaction history means that the agent is unable to compute a level of trust, but for the RTA, the agent's own interaction history is not relevant since it always seeks to calculate a level of trust based on opinions provided by others. Here we are exploiting the fact that in a large system it is probable that even though the truster might not have interacted with a trustee, there will be others in the system that have done so. Secondly, provided Assumptions 1 and 2 below hold, the resulting trust value is the same as it would be if all the observations had been observed directly by the truster itself.

Assumption 1
Common Behaviour: The behaviour of the trustee must be independent of the identity of the truster it is interacting with. Specifically, the following should be true:
$\forall a_x \quad \forall a_y, \quad B_{a_x,a_1} = B_{a_y,a_1}$.

Assumption 2
Truth Telling: The reputation provider must report its observations accurately and truthfully. In other words, it must be true that:
$\forall a_x \quad \forall a_y, \quad \mathcal{R}^t_{a_y,a_x} = \hat{\mathcal{R}}^t_{a_y,a_x}$.

Unfortunately, we cannot expect these assumptions to hold in a broad range of situations. For instance, a trustee may value interactions with one agent over another, and might therefore commit more resources to the valued agent to increase its success rate, thus introducing a bias in its perceived behaviour. Similarly, an opinion provider (who is requested to provide an opinion about the trustee) may have an incentive to misrepresent its true view of the trustee. Such an incentive could have a positive or negative effect on a trustee's reputation; if a strong co-operative relationship exists between trustee and opinion provider, the opinion provider may choose to overestimate its likelihood of success, whereas a competitive relationship may lead the rater to underestimate the trustee. Due to these possibilities, we consider the methods of dealing with inaccurate reputation sources an important requirement for a computational trust model. In the rest of this section, we introduce our solution to this requirement, building upon the basic model introduced thus far.

3.4.2 Handling Inaccurate Opinions

The method that calculates a reputation value from a number of opinions provided by others (see Section 3.4.1), is simple and elegant, but it is highly susceptible to influence from inaccurate opinions.

For example a truster agent a_1 asks the opinion of agents a_2, a_3 and a_4 about a trustee agent a_5. In the first part of this example we will assume agents a_2, a_3 and a_4 are accurate opinion providers. In this context, accurate means that their opinions are based on an unexaggerated and accurate representation of their past interaction with a_5. Suppose that the following opinions are provided, $(7, 2)$, $(4, 2)$ and $(6, 1)$, from a_2, a_3 and a_4 respectively. By using the methods described in Section 3.4 and 3.6, a_1 can calculate a reputation value and its confidence in that value for a_5. In this instance $\tau^r_{a_1,a_5} = 0.75$ and the associated confidence $\gamma_{a_1,a_5} = 0.75$, which is shown graphically in Figure 3.5 (plot a).

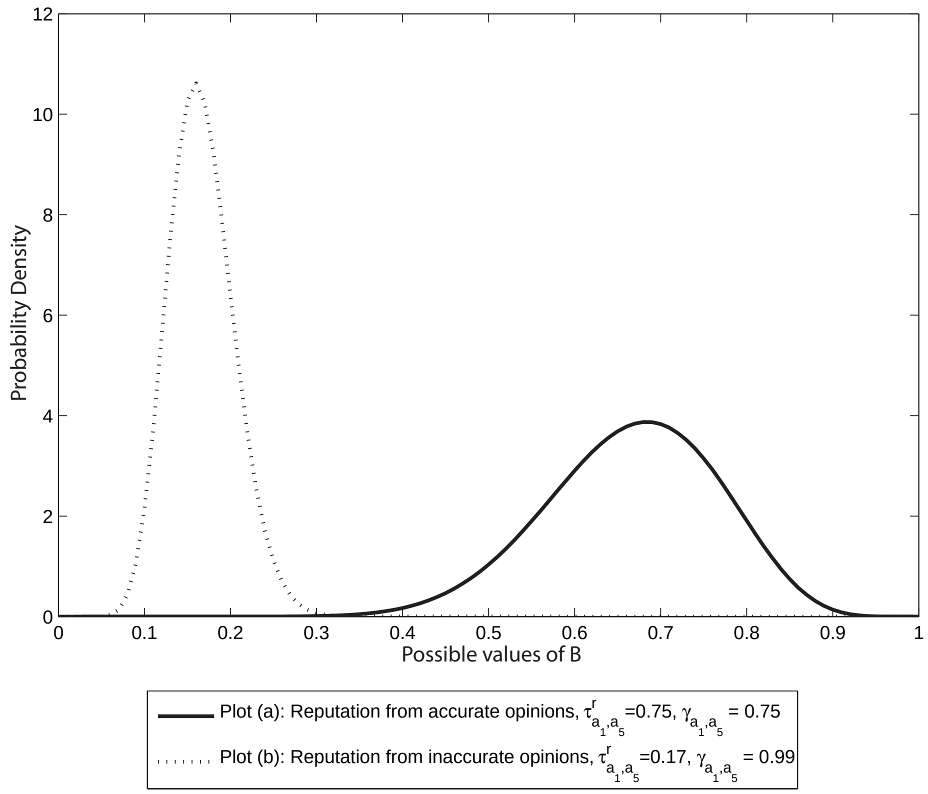

FIGURE 3.5: How inaccurate opinion can affect the reputation of a trustee.

In the second part of this example we remove the assumption that agents a_2, a_3 and a_4 have to provide accurate opinions (Assumption 2). Suppose that a_4 is in direct competition with a_5, and a high reputation value for a_5 (as calculated by a_1) has a direct negative impact on its utility. This gives a_4 an incentive to misrepresent the behaviour of a_5 when asked by a_1 for an opinion about a_5. In this case let us assume that opinions, $(7, 2)$, $(4, 2)$ and $(4, 75)$, are provided by a_2, a_3 and a_4 respectively. Once again using these, a_1 can calculate $\tau^r_{a_1,a_5} = 0.17$ and the associated confidence $\gamma_{a_1,a_5} = 0.99$, which is shown in Figure 3.5 (plot b).

This example highlights how a single agent can affect the trustee's reputation, and mislead the truster, by providing inaccurate opinions. In the first part when all agents provide accurate opinions, a_1 sees a_5 as moderately trustworthy ($\tau^r_{a_1,a_5} = 0.75$) and is moderately confident ($\gamma_{a_1,a_5} = 0.75$) in its calculation. In the second part, due to the inaccurate opinion provided by a_4, a_1 sees a_5 as untrustworthy ($\tau^r_{a_1,a_5} = 0.17$) and is very confident ($\gamma_{a_1,a_5} = 0.99$) in this view.

In this particular example, the opinion provided by a_4 is inaccurate because the agent lies about the real number of successful and unsuccessful interactions it has had with a_5 (which can be seen as a malicious act, specifically employed to mislead the truster). Inaccurate opinions are not only a result of malicious actions, but may also arise because the opinion provider has incomplete information. In both cases it is important for the truster agent to be able to assess the *probability of accuracy* of an opinion given by a third party.

The general solution to coping with inaccurate opinions is to adjust or ignore unreliable opinions prior to the combination of these into a single reputation value. This method of combining opinions results in a reputation value that is based more on the actual behaviour of the trustee than what others would have the truster believe about the behaviour of the trustee.

3.4.2.1 Our Approach

In the literature, Jøsang et al. (2006) discuss two basic approaches for assessing the reliability of opinions; these are *endogenous* and *exogenous* methods. Briefly, endogenous methods establish the accuracy of opinions by using a variety of statistical methods on a group of opinions obtained at a specific time. On the other hand, exogenous methods rely on observing and predicting the behaviour of an individual opinion provider that provides the opinion, and then uses this evidence to determine the reliability of the opinion obtained.

Currently, endogenous techniques, such as those reported by Whitby et al. (2005), Dellarocas (2000) and Chen and Singh (2001), assume that opinion providers giving inaccurate opinions will be in the minority of the complete set of providers giving opinions for a single request. These techniques classify opinions as inaccurate if they statistically deviate from the mainstream opinion. This form of assessment has two main limitations, which we describe below.

Firstly, the statistical methods employed are limited in environments where the number of inaccurate opinion providers are less than or equal to the number of accurate providers. In these environments it is difficult to determine the mainstream opinion, and even if one can be determined it is likely to be formed by the inaccurate opinions.

Secondly, the endogenous approach does not factor in the outcomes from the assessment, made by the truster, on previous occasions. After numerous episodes of gathering opinions and assessing their accuracy, a truster will have identified a number of inaccurate opinion providers in each episode, some of which may be consistently providing inaccurate opinions. The endogenous approach ignores this historic information, and the ability of the opinion provider to provide an accurate and reliable opinion in previous episodes is not factored into the assessment made on the latest opinion provided by the same provider.

Our approach can be classified as exogenous (approaches described in Section 2.4.1.2 and 2.4.2.4 can also be classified as exogenous), because it does not assess the accuracy of an opinion by comparing that opinion to the others obtained in the same episode. In our approach, we believe that the behaviour of the opinion provider in previous episodes of opinion provision is the best indication of whether or not future opinions will be accurate. Our approach introduces a *probability of accuracy*, a metric that represents the truster's belief of how accurate the opinion from an opinion provider is, given the past outcomes of interactions that the truster had where the same provider gave a similar opinion. We make the assumption that similar opinions from an individual opinion provider can be compared regardless of the identity of the agent the opinion applies to. For example, an opinion provider may have given a truster (on three individual requests for opinions) the opinion of "*trustworthy*" for agents a_1, a_2 and a_3 at some point in the past. We use these three opinions (regardless of the fact that they apply to three separate agents) to predict the probability of accuracy of the opinion provider's opinion the next time it gives an opinion of "*trustworthy*".

More specifically, we monitor each opinion provider, recording the history of opinion provision as a series of opinions that were given by the provider and the actual observed outcome of the interaction that the opinion was requested for. This opinion history is then used to calculate the probability of accuracy. For example, consider a truster a_1 that requests the opinion of others about a trustee a_2, and a particular opinion provider a_3 that responds with an opinion that tells a_1 that a_2 is "*very trustworthy*". To determine if this opinion is accurate, first a_1 consults its opinion history for a_3 and obtains a set of previous opinions where a_3 has given the opinion of "*very trustworthy*" and the corresponding outcomes of the interactions that these opinions were asked for. This set is then used to calculate a probability that the new opinion of "*very trustworthy*" is accurate. Finally, before a_1 uses the "*very trustworthy*" opinion from a_3 to calculate the reputation of a_2, it adjusts it using the probability of accuracy calculated in the previous step. If more than one opinion is being used to calculate the reputation, then all the adjusted opinions are aggregated using the technique described in Section 3.4.1. This results in a more accurate reputation value for agent a_2. The remainder of this section, in three parts, describes our approach in more detail. We begin by describing how an agent records the opinion history for a particular opinion provider.

3.4.2.2 Recording Opinion History

The first step in our method of handling inaccurate opinions is to record each opinion provided from each opinion provider and the outcome from the interaction that the opinion was required for. An outcome at time t, O^t, is binary and can simply be recorded as 0 (unsuccessful) or 1 (successful), as described in Section 3.2. Opinions are recorded as $\hat{\mathcal{R}}^t$, therefore, the opinion history can be seen as a series of tuples containing an opinion $\hat{\mathcal{R}}^t_{a_3,a_2}$ from an opinion provider a_3, about a trustee a_2, and an outcome of an interaction between the truster a_1 and the trustee a_2 at time t, $(a_3, a_2, \hat{\mathcal{R}}^t_{a_3,a_2}, O^t_{a_1,a_2})$.

This simple representation presents us with a problem, especially in the model's deployment — since an opinion can take on an infinite number of values. Using this method, an empty or a very small set will be obtained when using the history to create a set of tuples containing opinions that are the same as the one being assessed. This means that the agent will never have enough evidence to calculate the probability of accuracy of a given opinion confidently.

We solve this problem by approximation. All possible values of $\hat{\mathcal{R}}^t$ are split into predefined bins, according to the expected value $E_{\hat{\mathcal{R}}^t}$ (Equation 3.10) resulting from the beta distribution obtained using $\hat{\mathcal{R}}^t$. In TRAVOS we define the bins with an upper (bin_{max}) and lower (bin_{min}) limit, and $\hat{\mathcal{R}}^t$ falls into the bin where $bin_{min} \leq E_{\hat{\mathcal{R}}^t} < bin_{max}$. The default bins used are shown

Lower bound	Upper Bound
0	0.2
0.2	0.4
0.4	0.6
0.6	0.8
0.8	1

TABLE 3.1: Table describing the setup of the five bins used in categorizing opinions in TRAVOS.

in Table 3.1. Using this approach, the opinion history $\mathcal{H}$ is represented as a set of tuples of the form $(a_3, a_2, bin_{min}, bin_{max}, \hat{\mathcal{R}}^t_{a_3,a_2}, O^t_{a_1,a_2})$

$$E_{\hat{\mathcal{R}}^t} = \frac{\alpha}{\alpha+\beta} \quad \text{where} \quad \begin{aligned} \alpha &= \hat{m}^t + 1 \\ \beta &= \hat{n}^t + 1 \end{aligned} \tag{3.10}$$

3.4.2.3 Calculating The Probability Of Accuracy

The second step in our method for handling inaccurate opinions is to calculate a probability that the opinion $\hat{\mathcal{R}}^t_{a_3,a_2}$ provided by a particular agent is accurate. We denote this probability as $\rho^t_{a_1,a_3}$ — the accuracy of the opinion provided by opinion provider a_3 according to the truster a_1 at time t.

To calculate this probability we must first calculate the expected value of the opinion $\hat{\mathcal{R}}^t_{a_3,a_2}$ being assessed, using Equation 3.10, to determine the bin into which the opinion falls. More specifically, this allows us to calculate the upper and lower bounds of the bin. Then, using the opinion provider, and the upper and lower bounds of the bin the opinion belongs to, we can obtain a subset, ***h***, of $\mathcal{H}$ that contains all tuples matching the pattern $(a_3, -, bin_{min}^{\hat{\mathcal{R}}^t_{a_3,a_2}}, bin_{max}^{\hat{\mathcal{R}}^t_{a_3,a_2}}, -, -)$ containing previous opinions from a_3 that are similar to $\hat{\mathcal{R}}^t_{a_3,a_2}$.

The set ***h*** is used to determine the α and β parameters of a beta distribution (using Equation 3.11), which represents the actual behaviour of the trustee a_2 (as observed by the truster a_1) in all situations where the opinion provider a_3 has provided a similar opinion to $\hat{\mathcal{R}}^t_{a_3,a_2}$.

$$\begin{aligned} \alpha &= \text{Number of tuples in } \boldsymbol{h} \text{ (where } O_{a_x,a_y} = 1) + 1 \\ \beta &= \text{Number of tuples in } \boldsymbol{h} \text{ (where } O_{a_x,a_y} = 0) + 1 \end{aligned} \tag{3.11}$$

The probability of accuracy $\rho^t_{a_1,a_3}$ is then defined as the area under the beta curve produced using ***h***, bounded by the upper and lower limit of the bin that $\hat{\mathcal{R}}^t_{a_3,a_2}$ belongs to (see Equation 3.12).

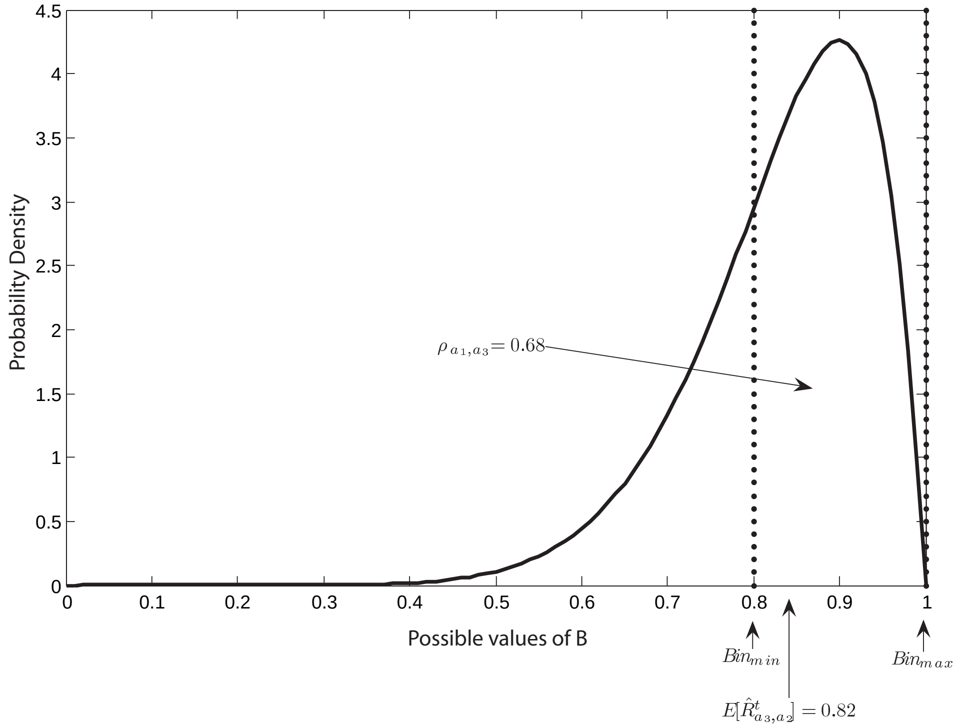

FIGURE 3.6: An accurate opinion yields a large value of ρ: The beta curve is drawn from outcomes of past interactions where the opinion provider gave a similar opinion to $\hat{\mathcal{R}}^t_{a_3,a_2}$.

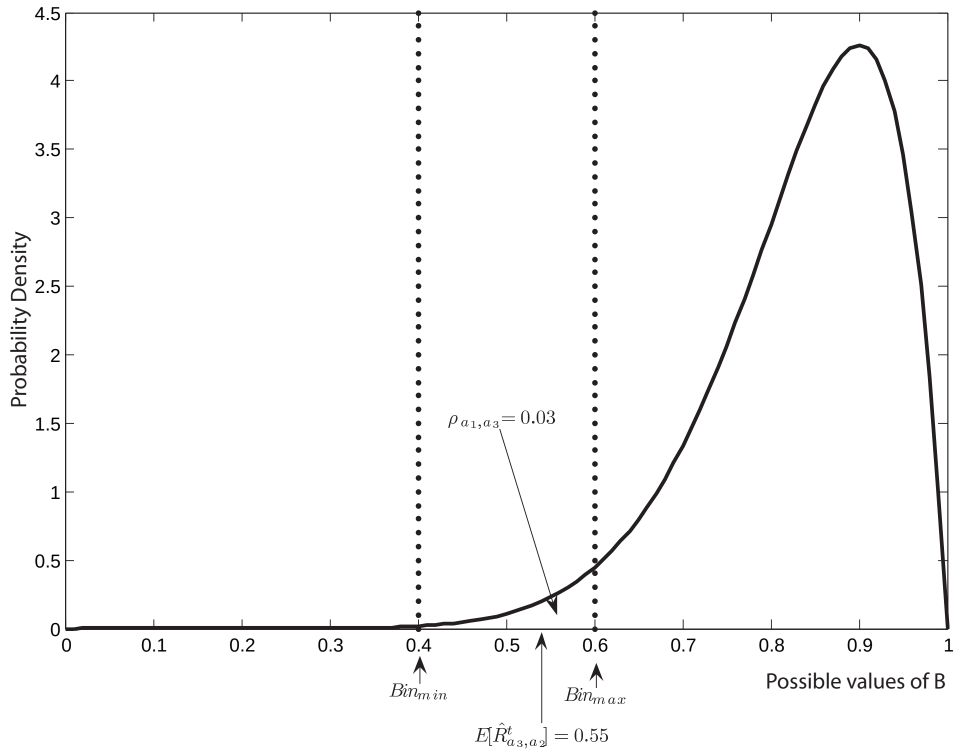

FIGURE 3.7: An inaccurate opinion yields a small value of ρ: The beta curve is drawn from outcomes of past interactions where the opinion provider gave a similar opinion to $\hat{\mathcal{R}}^t_{a_3,a_2}$.

If the opinion provider a_3 has always been telling the truth and providing accurate opinions, one would find over time that the beta curve will peak in the bin that $\hat{\mathcal{R}}^t_{a_3,a_2}$ falls in, resulting in a high value for $\rho^t_{a_1,a_3}$ (see Figure 3.6). On the other hand, if agent a_3 constantly lied and provided inaccurate opinions, then one would see a beta distribution that does not peak in the bin that $\hat{\mathcal{R}}^t_{a_3,a_2}$ falls in, resulting in a low value for $\rho^t_{a_1,a_3}$ (see Figure 3.7).

$$\rho_{a_1,a_3} = \frac{\int_{bin_{max}^{\hat{\mathcal{R}}^t_{a_3,a_2}}}^{bin_{min}^{\hat{\mathcal{R}}^t_{a_3,a_2}}} (B_{a_1,a_2})^{\alpha-1}(1-B_{a_1,a_2})^{\beta-1}dB_{a_1,a_2}}{\int_0^1 U^{\alpha-1}(1-U)^{\beta-1}dU} \tag{3.12}$$

3.4.2.4 Adjusting the Opinion

The final stage in handling inaccurate opinions is to reduce their impact on the overall reputation of a trustee. To achieve this we first consider the properties of the beta distribution, obtained from a single provider's opinion (which we call the *opinion distribution*), that determines its effect on the beta distribution representing the trustee's reputation (which we call the *combined distribution*). More specifically, we consider the expected value of the distribution and its standard deviation. By adding opinions to obtain a reputation value, we move the expected value of the combined distribution in the direction of the opinion distribution. The standard deviation of the opinion distribution contributes to the confidence of the combined distribution, but more significant is that relative to the standard deviation of the prior distribution it determines how far toward the opinion distribution the expected value will move. The relationship between the change in expected value, and the standard deviation, of the distribution after combining an opinion, is non-linear.

Consider an example with a combined distribution d_1 and two opinion distributions, d_2 and d_3, with shape parameters, expected value and standard deviation (denoted σ) as shown in Table 3.2. The results of combining d_2 and d_3 with d_1 are shown in the last two rows. These distributions are shown in Figure 3.8, to aid understanding. From the table and the figure one can see that the expected value of the distributions d_2 and d_3 are the same, but there is a difference in their standard deviation. This can be seen more clearly in Figure 3.8: plots (b) and (c) peak at the same value, but plot (b) is wider than plot (c). Although the difference between the standard deviations of d_2 and d_3 is small (0.02), the result of combining d_2 with d_1 is significantly different from combining d_3 with d_1. The new combined distribution, resulting from combining the opinion distribution d_2 with the combined distribution d_1, has an expected value approximately between the expected values of d_1 and d_2. However, in the case of combining d_3 with d_1, the relatively small parameter values for d_1 compared to d_3 means that the new combined distribution has an expected value much closer to that of d_3, and d_1 has almost no impact on the combination. This result is very significant because it shows how, if opinions are not adjusted, an inaccurate opinion provider could deliberately increase the weight the truster puts in its opinion by providing very large values of $\hat{m}$ and $\hat{n}$ in its opinion $\hat{\mathcal{R}}$, which in turn determine α and β. This is obviously

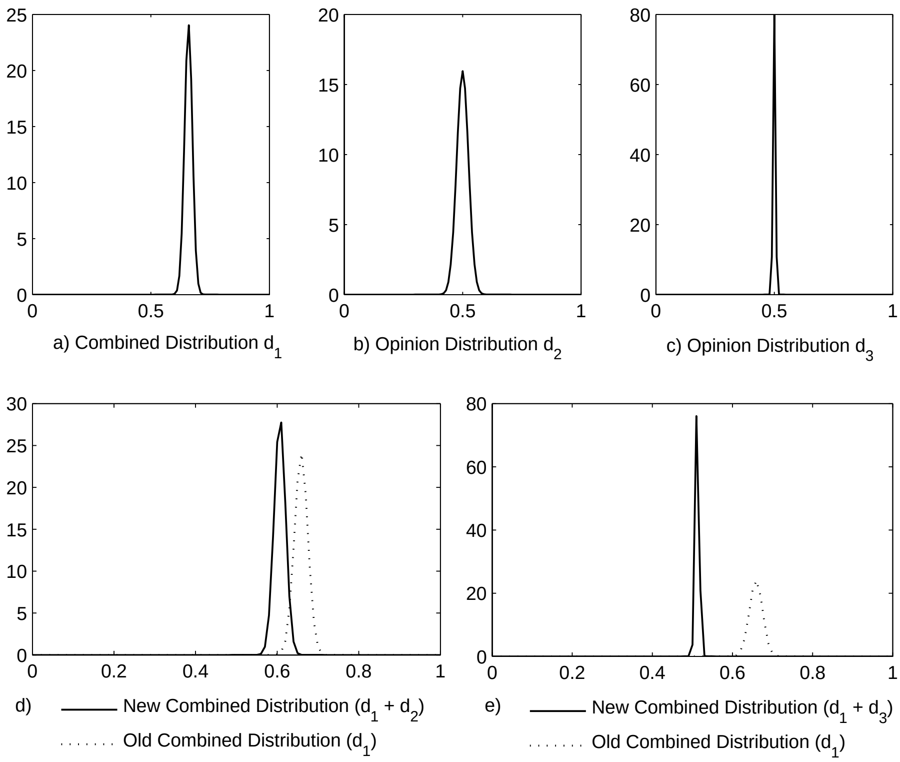

FIGURE 3.8: Beta plots showing the effect of combining two different opinions on the combined distribution.

a limitation of the method we use to combine opinions, in which parameter values are simply summed (see Section 3.4.1).

Distribution	α	β	$E[B]$	σ
d_1	540	280	0.6585	0.0165
d_2	200	200	0.5000	0.0250
d_3	5000	5000	0.5000	0.0050
$d_1 + d_2$	740	480	0.6066	0.0140
$d_1 + d_3$	5540	5280	0.5120	0.0048

TABLE 3.2: Example beta distributions and the results of combining them.

In light of this, we adopt an approach that significantly reduces $\hat{m}$ and $\hat{n}$ (in an opinion $\hat{\mathcal{R}}$), thus decreasing α and β, based on the probability of accuracy for a given opinion. This method reduces the distance between the expected value, $E_{\hat{\mathcal{R}}^t}$ and the variance, $\sigma^2_{\hat{\mathcal{R}}^t}$, of the opinion distribution, and the uniform distribution[4]. We denote the expected value of the uniform distribution as $E_{uniform}$ and its variance as $\sigma^2_{uniform}$. Referring back to our example where an opinion provider a_3 provides an opinion to a truster a_1 about a trustee a_2, equations 3.13 and 3.14 show how this reduction in distance is achieved. We use the over-bar, for example $\bar{E}$, to indicate we are referring to the adjusted distribution.

[4]In the uniform distribution $\alpha = 1$ and $\beta = 1$. In TRAVOS the prior distribution is the uniform distribution and it represents a state of no information. However, a different prior distribution may be used.

$$\bar{E}_{\hat{\mathcal{R}}^t_{a_1,a_3}} = E_{uniform} + \rho_{a_1,a_3} \cdot (E_{\hat{\mathcal{R}}^t_{a_1,a_3}} - E_{uniform}) \tag{3.13}$$
$$\bar{\sigma}^2_{\hat{\mathcal{R}}^t_{a_1,a_3}} = \sigma^2_{uniform} + \rho_{a_1,a_3} \cdot (\sigma^2_{\hat{\mathcal{R}}^t_{a_1,a_3}} - \sigma^2_{uniform}) \tag{3.14}$$

Once all opinion distributions (from all the opinion providers) have been adjusted in this way, we would like to combine them as described in Section 3.4.1. However, before an opinion can be combined we must calculate the adjusted values for $\hat{m}^t$ and $\hat{n}^t$ which form the opinion. We can use the standard beta parameter estimation equations (DeGroot and Schervish, 2002) to estimate the parameters for the adjusted distributions, using $\bar{E}_{\hat{\mathcal{R}}^t_{a_1,a_3}}$ and $\bar{\sigma}_{\hat{\mathcal{R}}^t_{a_1,a_3}}$ as shown in Equation 3.15. The adjusted values for $\hat{m}^t$ and $\hat{n}^t$ are then given by subtracting the prior parameter settings from the adjusted distribution parameters, as shown in Equation 3.16.

$$\begin{aligned}
\bar{\alpha} &= \frac{(\bar{E}_{\hat{\mathcal{R}}^t_{a_1,a_3}})^2 - (\bar{E}_{\hat{\mathcal{R}}^t_{a_1,a_3}})^3}{(\bar{\sigma}_{\hat{\mathcal{R}}^t_{a_1,a_3}})^2} - (\bar{E}_{\hat{\mathcal{R}}^t_{a_1,a_3}}) \\
\bar{\beta} &= \frac{(1 - \bar{E}_{\hat{\mathcal{R}}^t_{a_1,a_3}})^2 - (1 - \bar{E}_{\hat{\mathcal{R}}^t_{a_1,a_3}})^3}{(\bar{\sigma}_{\hat{\mathcal{R}}^t_{a_1,a_3}})^2} - (1 - \bar{E}_{\hat{\mathcal{R}}^t_{a_1,a_3}})
\end{aligned} \tag{3.15}$$

$$\bar{m}_{a_3,a_2} = \bar{\alpha} - 1 \quad , \quad \bar{n}_{a_3,a_2} = \bar{\beta} - 1 \tag{3.16}$$

3.5 Combining Direct Trust and Reputation

We have defined two agents that have different methods of using evidence in the system to calculate the level of trust in another. Both approaches have advantages and disadvantages, and we can exploit them by devising a method to calculate trust by combining the methods used by the DTA and the RTA.

The use of the beta distribution to model the behaviour of the trustee gives us another elegant solution to the problem of combining personal observations with the opinions provided by others. As shown in Section 3.2, the personal interaction history between agents a_1 and a_2 is represented by $\mathcal{R}^t_{a_1,a_2} = (m^t_{a_1,a_2}, n^t_{a_1,a_2})$. The opinions provided by others can be represented in a similar form, $\hat{\mathcal{R}}^t_{a_3,a_2} = (\hat{m}^t_{a_3,a_2}, \hat{n}^t_{a_3,a_2})$, as described in Section 3.4. Then, to combine the personal experience with the opinions, we must first enumerate all the opinions provided as shown in Equation 3.7. Having obtained the resulting opinion values N_{a_1,a_2} and M_{a_1,a_2}, we can use them with $m^t_{a_1,a_2}$ and $n^t_{a_1,a_2}$ in Equation 3.17 to give beta shape parameters. These shape parameters can then be used in Equation 3.18 to give a level of combined trust $\tau^c_{a_1,a_2}$.

$$\hat{\alpha} = M_{a_1,a_2} + \bar{m}^t_{a_1,a_2} + 1 \quad \text{and} \quad \hat{\beta} = N_{a_1,a_2} + \bar{n}^t_{a_1,a_2} + 1 \tag{3.17}$$

$$\tau^c_{a_1,a_2} = \frac{\hat{\alpha}}{\hat{\alpha} + \hat{\beta}} \tag{3.18}$$

This method of combining direct trust with reputation yields the *Combined Trust Agent* (CTA), which calculates the level of trust in others by using $\tau^c_{a_1,a_2}$ and, thus, in the CTA, $\tau_{a_1,a_2} = \tau^c_{a_1,a_2}$. The advantage of the CTA is that both evidence internal to the agent (which can assumed to be accurate) and external evidence (opinions which may be biased)[5] is used. This means that initially the agent's calculated level of trust in another agent will be based heavily on others' opinions, as it will have had little or no personal experience. However, as it gains experience, a significant proportion of the observations that are used to calculate the trust will be observations that the agent has made. The problem with the CTA is that even with the combination, there may be some reputation providers that are able to influence the trust calculation by providing heavily (positively or negatively) biased opinions. This limitation of the Combined Trust Agent can be overcome by giving the agent the ability to determine how much confidence it has in its own observations and then only seek the opinions of others if this level of confidence is not sufficient. To this end, in the following section, we introduce a metric that allows such switching capability in the agent's reasoning.

3.6 Modelling Confidence

In the previous sections, we have shown how an agent can derive a trust value that it can use to compare the trustworthiness of different agents. However, the DTA is susceptible to two problems created by the need for adequate evidence (observations) to calculate a meaningful value for trust. Firstly, an agent may not have interacted with another agent for which it is calculating a level of trust. This means that it has no personal experience and $m^t_{a_1,a_2} = n^t_{a_1,a_2} = 0$. Secondly, an agent may have had few interactions and observed outcomes with another. In both these cases, the calculated value of τ_{a_1,a_2} will be a poor estimate for the actual value of B_{a_1,a_2}. Intuitively, having observed many outcomes for an event will lead to a better estimate for the future probability for that event (assuming all other things are equal). The RTA overcomes these problems, but suffers from the fact that despite the agent gaining personal experience over time, the agent does not factor it into the calculation of trust.

The CTA combines both direct trust and reputation to form a combined level of trust, which overcomes the problems exhibited by the DTA and RTA. However, this method of combination will continue regardless of the level of experience the agent has achieved. Ideally, we would like the agent to stop using opinions from others once it has enough experience of its own. These problems create the need for an agent to be able to measure its *confidence* in the value of trust it calculates. To this end, we incorporate a confidence metric in TRAVOS, based on standard methods of calculating confidence intervals taken from statistical analysis.

[5]We have already presented a mechanism that can allow an agent to cope with misleading opinions.

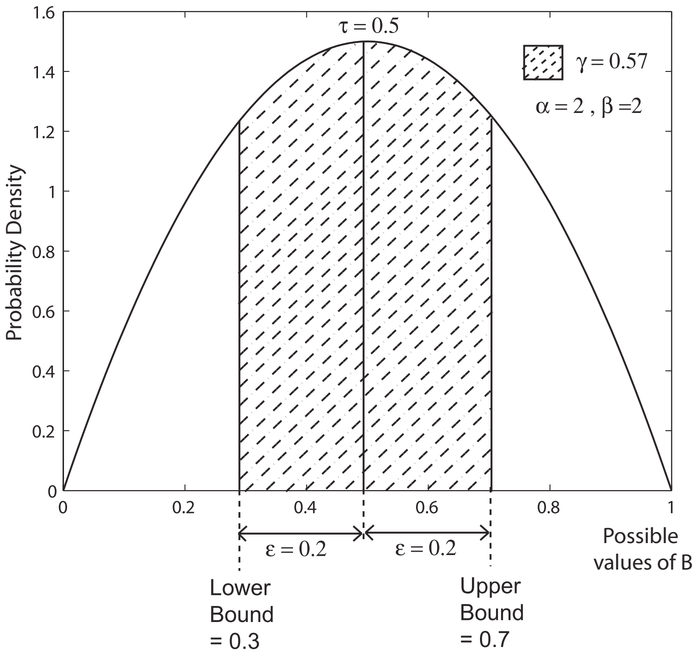

FIGURE 3.9: Confidence is the area under the beta distribution bounded by the upper and lower limits, calculated by adding and subtracting the error ϵ from the trust value τ.

Specifically, the confidence metric γ_{a_1,a_2} is a measure of the probability that the actual value of B_{a_1,a_2} lies within an acceptable level of error ϵ about τ_{a_1,a_2}. It is calculated using Equation 3.19, which is shown graphically in Figure 3.9. The acceptable level of error ϵ influences how confident an agent is given the same number of observations. For example, if the number of observations remains constant, a larger value of ϵ causes an agent to be more confident in its calculation of trust than a lower value of ϵ.

$$\gamma_{a_1,a_2} = \frac{\int_{\tau_{a_1,a_2}+\epsilon}^{\tau_{a_1,a_2}-\epsilon} (B_{a_1,a_2})^{\alpha-1}(1 - B_{a_1,a_2})^{\beta-1} dB_{a_1,a_2}}{\int_0^1 U^{\alpha-1}(1-U)^{\beta-1} dU} \tag{3.19}$$

Having described how to calculate confidence we can can modify the CTA, so that it has a decision-making process that enables it to use only direct trust or reputation, or a combination of both as a basis for calculating overall trust. So, now in TRAVOS, a CTA a_1 calculates τ_{a_1,a_2} based on its personal experiences with a_2. If this value of τ_{a_1,a_2} has a corresponding confidence γ_{a_1,a_2} below that of a predetermined *minimum confidence level*, denoted θ_γ, then a_1 will seek the opinions of other agents about a_2 to boost its confidence above θ_γ. These collective opinions form a_2's reputation and, by seeking it, a_1 effectively obtains a larger set of observations.

Having described the mechanisms of TRAVOS in detail, in the following section we present a summary of the model, identifying which of our aims are satisfied by TRAVOS.

3.7 Summary

In Chapter 1 we argue that in large complex open systems there is a need to assure good interactions between agents. More specifically, in the context of systems that support agent-based VOs there is a need to enable an agent to make sound decisions given the uncertainty inherent in such systems. A large part of this uncertainty can be attributed to an agent's inability to determine the behaviour of its interaction partner. For example, in an agent-based VO an agent may wish to delegate a task to one of the members, but it is unsure whether or not the task will completed as expected. To this end, we present TRAVOS, a computational trust model that allows an agent to determine a level of trust that can be placed in its interaction partner. In our model, trust represents the probability that an interaction will result in a good (successful) outcome. So, by using trust in its decision-making processes, an agent is able to account for some uncertainty regarding the behaviour of its interaction partner. For example, consider an agent that uses TRAVOS, if this agent calculates a high level of trust in a potential interaction partner, then it believes that an interaction with this partner is most likely to be successful.

In more detail, TRAVOS provides an agent (the truster) with a number of different ways in which it can determine the trust level of another agent (the trustee). We briefly describe the different ways below, identifying the aims (as presented in Section 1.3) that are satisfied.

Direct trust — TRAVOS provides a mechanism for a truster to evaluate the trust level of a trustee using past experiences the truster has had with the trustee (Aim 1). Here, the trust level calculated for trustee is a function of the trustee's past behaviour, and allows a truster, over time, to select the most reliable (trustworthy) interaction partners.

Reputation — TRAVOS is designed to operate in large open systems, where it is likely that agents will often interact with others with whom they have no previous interaction history. In such situations, the truster is unable to determine a meaningful trust level in a trustee using direct trust. However, in TRAVOS, we accommodate for this limitation by providing a mechanism that allows a truster to ask other agents (opinion providers) for their opinions about the trustee, and then to combine all these opinions to give a reputation level for the trustee (Aim 2). The risk of receiving misleading opinions is high in open systems, and so, we incorporate into this reputation mechanism a filter that prevents a truster being misled by such opinions (Aim 3). More specifically, the filtering mechanism allows the truster to adjust the opinions received from an opinion provider by learning the reliability of that opinion provider in providing good opinions over time.

Combination of direct trust and reputation — The mechanisms described above provide a truster with two methods of determining the trust level for a trustee. Using direct trust is advantageous because any evidence used in calculating direct trust is observed by the truster, and is therefore more reliable than the evidence used in calculating reputation (which is reported by others in the form of opinions). On the other hand, when there is no interaction history between the truster and trustee, the truster has little choice but to use reputation. To allow a truster to efficiently use both mechanism to determine a trustee's trust level, we incorporate a confidence metric in TRAVOS. The confidence metric allows a truster to determine how confident it is in the trust level it calculates for a trustee (Aim 5), given the evidence used in the calculation. By using the confidence metric, a truster is able to use direct observations to determine a trust level, and only seeks opinions from others if the associated confidence is below a minimum.

The mechanisms described above meet the requirements of the VO formation stage in a VO life cycle (as described in Section 2.3.4). However, TRAVOS falls short of meeting one of our main aims (Aim 4) and a key requirement for the VO functioning stage (as described in 2.3.5), which is to use the social information in the trust calculations. We address this shortcoming in Chapter 5, where we describe in detail how this basic model can be extended to allow agents to exploit the social information present in VOs.

Finally, in Section 2.4 we described many models, some of which already meet our aims (as discussed in Section 2.5.2). We believe that the probabilistic approach employed is more effective than other solutions that exist, and in particular our exogenous approach to assessing the reliability of opinions is better than the exogenous approach used by others. Furthermore, we believe that our model can be applied to realistic applications. To this end, in the following chapter, to support this argument, we present an empirical and system evaluation of our model. The former shows how the different components of TRAVOS perform against each other and against other similar models, and the latter complements this by showing how TRAVOS can be used by an agent to account for uncertainty in its decision-making processes within a VO.

Chapter 4

Evaluation of TRAVOS

In Chapter 3 we presented the theory for TRAVOS, a novel model of trust for agent-based virtual organisations(VOs). The model enables agents to make sound decisions under uncertainty, and allows an agent to cope with inaccurate evidence (for example misleading opinions provided by others) in the trust calculations. Having presented the theory, it is necessary for us to evaluate how different components of TRAVOS perform, and how TRAVOS performs against other such models; by so doing we demonstrate the validity of our unique approach.

In this chapter, we evaluate our approach in two ways: empirical and system evaluation. The empirical evaluation allows us to, in a simulated environment, explore how the different components (Direct Trust agent, Reputation agent and Combined Trust agent) of TRAVOS perform against each other, allowing us to understand which component is appropriate for a given context. The system evaluation serves the purpose of demonstrating how our model is implemented in a real system, and the impact it has on agents's decisions in such a system.

We begin by presenting the results of the empirical evaluation performed on TRAVOS, consisting of a number of experiments, which are run in a simulated environment. Initially, we describe the evaluation testbed and the experimental methodology in Section 4.1.1. Two main types of experiments are carried out. The first set (described in Section 4.1.2) explores TRAVOS performance, with respect to an agent's ability to assess the trustworthiness of another agent. The second set (Section 4.1.3) describes how TRAVOS performs against the most similar model found in the literature (namely, the Beta Reputation System, which is described in Section 2.4.1.3).

Following the empirical evaluation, we present the TRAVOS system evaluation in Section 4. In more detail, we begin by describing a agent-based VO scenario, which we then use to demonstrate how an agent can use TRAVOS in its decision-making processes. We conclude the system evaluation by providing details of how TRAVOS can be deployed in an industrial system.

4.1 Empirical Evaluation

4.1.1 Experiment Methodology

The evaluation testbed for TRAVOS is a computer simulation consisting of three distinct sets of agents: service provider agents $\mathcal{S} \subset \mathcal{A}$, consumer agents $\mathcal{C} \subset \mathcal{A}$, and opinion provider agents $\mathcal{O} \subset \mathcal{A}$.

Service provider agents behave in a particular way when providing a service to a consumer agent determined before each experiment. Specifically, the behaviour of a particular service provider $s_i \in \mathcal{S}$ is determined by the parameter B_{s_i}, which is the probability that the service provider will provide the service as specified (see Section 3.2). Consumer agents can request opinions about service providers from opinion provider agents. The behaviour of an opinion provider agent, $o \in \mathcal{O}$, falls into one of three mutually exclusive categories, which determine the type of opinion that it gives to the consumer agent. The categories are described below.

1. **Accurate Opinion Providers**, $o^a \in \mathcal{O}$ These opinion providers provide a true account of what they have observed in the past. This means that the opinion distribution, derived from their opinion (see Section 3.4.2.4), represents the actual history of interaction between the opinion provided and the service provider under question.
2. **Noisy Opinion Providers**, $o^n \in \mathcal{O}$ These opinion providers add gaussian noise to their opinion before giving it to the consumer agent. By using this, we model the fact that the behaviour of an opinion provider may be slightly different depending on who they are interacting with.
3. **Lying Opinion Providers**, $o^l \in \mathcal{O}$ These opinion providers aim to mislead the consumer agent by lying about their previous encounters with a particular service provider. They provide opinions which have been adjusted so that $E[\hat{\mathcal{R}}^t] = 1 - E[\mathcal{R}^t]$.

Against this background, in this empirical evaluation, each experiment consists of a series of episodes in which a consumer agent c_1 is asked to assess its trust in all the providers $\mathcal{S}$. The purpose of the experiments is to measure how accurately a consumer agent $c \in \mathcal{C}$ evaluates the level of trust for all $s \in \mathcal{S}$, denoted by $\tau_{c,s}$. Based on the trust values calculated by c_1, we calculate c_1's mean estimation error for the episode using Equation 4.1. The mean estimation error is chosen because it serves as a metric to allow us to measure the performance of a consumer agent's ability to assess the trust and, in turn, predict the behaviour, of the service provider population as a whole.

The composition of the service provider population has an impact on this metric. If the service provider population is configured in a way such that each supplier's behaviour is equal to the prior value that a consumer may calculate, then the resulting mean estimation error produced by the consumer would be very low and this value would be misleading. For this reason we keep the configuration of the service provider population constant throughout all the experiments. All experiments are run with a population of 101 service providers with the values of B chosen uniformly between 0 and 1 at intervals of 0.01. To ensure that the results of each episode are independent, the interaction history between all agents is cleared before every episode, and re-populated according to set parameters. All the results that we discuss have been tested for statistical significance using Analysis of Variance techniques and Scheffé tests (Cohen, 1995).

$$avg\_estimate\_err = \frac{1}{N}\sum_{i=1}^{n} abs(\tau_{c_1,s_i} - B_{s_i}) \tag{4.1}$$

Experiment	No. of Lying	No. of Noisy	No. of Accurate	Total
1	0	0	20	20
2	0	10	10	20
3	0	20	0	20
4	10	0	10	20
5	20	0	0	20

TABLE 4.1: Configurations for the opinion provider population.

4.1.2 TRAVOS Agents Performance

In Chapter 3 we developed a set of agents, namely the Direct Trust agent (DTA), the Reputation Trust agent (RTA) and the Combined Trust agent (CTA), where each agent has a different method of assessing trust. Given this, we carry out a set of experiments to evaluate the performance of these different agents in different environments. More specifically, we design a set of environments where we vary the number of accurate, noisy and lying opinion providers. The composition of the opinion provider population for the different experiments is shown in Table 4.1. We evaluate the DTA, RTA and CTA by making the consumer agent in each experiment be a particular one. Each profile is tested in a series of five experiments (as shown in Table 4.1), and in each experiment the number of times the consumer agent requests an opinion from an opinion provider is fixed at 10. Each experiment is carried out over a number of episodes and in each episode we increase the number of consumer agent to service provider agent interactions (up to 20); this means that the experiment starts with 0 interactions between the consumer and service provider, and finishes with 20 interactions between them.

The mean estimation error from the results obtained is shown in Figure 4.1. Adopting the approach taken by the DTA, the mean estimation error in a consumer's trust assessment for a service provider decreases as the number of interactions between the consumer and service provider increase. The CTA shows similar performance to the RTA for a low number of interactions between consumer and service providers. However, in some cases such as the experiments

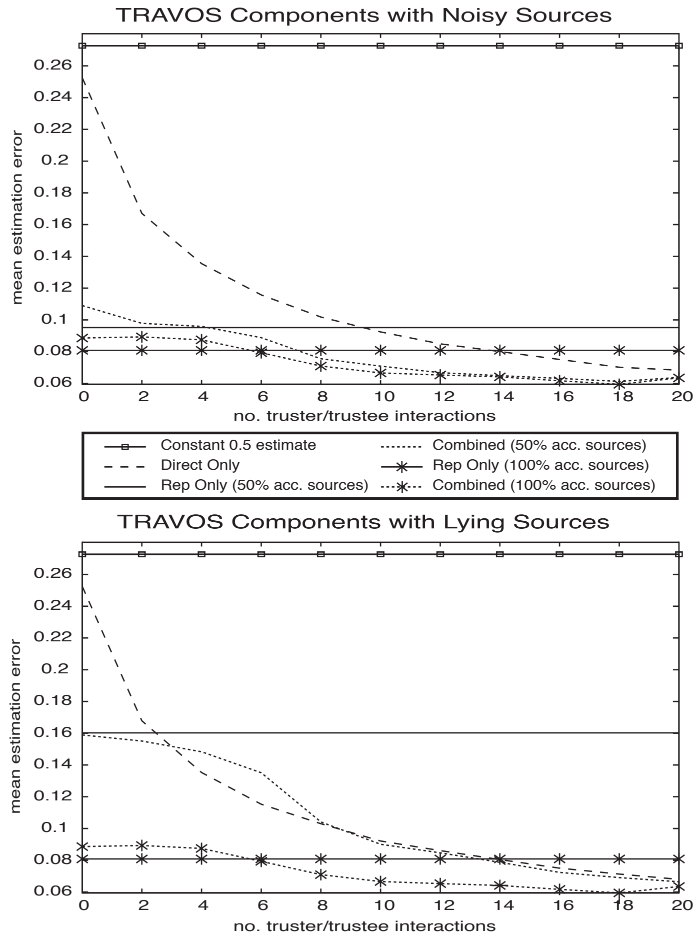

FIGURE 4.1: TRAVOS component performance.

with noisy sources (Figure 4.1a), the CTA performs marginally worse than the RTA[1], which can be attributed to the fact that TRAVOS places a bias on the direct interactions of a consumer. This means that whilst the consumer has low experience, it still places more weight on its own experiences than opinions provided by others. This low experience is the source of the greater error and the poorer performance.

An interesting result is produced when the CTA is used in an environment where 50% of the opinion provider population is made up of lying opinion providers. In this environment, the CTA is misled enough, temporarily, to produce a greater error than the DTA. This is a symptom of the relatively few interactions between consumers and opinion providers (10), which is not enough for a consumer to develop a sufficiently rich opinion provision history to discount the liars completely. The effect disappears when the number of such interactions is increased to 20 with all other conditions kept the same.

[1]This effect was not considered significant under a Scheffé test, but was considered significant by Least Significant Difference Testing. The latter technique is, in general, less conservative at concluding that a difference between groups does exist.

4.1.3 TRAVOS Against the Beta Reputation System

Of the existing computational trust models in the literature, the most similar to TRAVOS is the Beta Reputation System (BRS) (see Section 2.4.1.3 for more detail). Like TRAVOS, the BRS uses the beta density function to calculate the value of trust in a trustee. However, the models are significantly different in the manner in which they cope with inaccurate opinions produced by opinion providers for the reputation calculation of a trustee. TRAVOS uses the exogenous approach to handle inaccurate opinions, which assesses each opinion source individually, based on the perceived accuracy of past opinions. In contrast, BRS uses an endogenous approach, and assumes that the majority of reputation sources provide an accurate opinion, ignoring any opinions that deviate significantly from the average. Due to its similarities, the BRS forms a significant benchmark to which the performance of TRAVOS can be compared. To this end, we focus our evaluation of the performance of each system only on the aspects that differ in the two models. Thus, we have limited our evaluation scenarios to using consumer agents that evaluate a trustee based on the trustee's reputation alone. To show variation in performance depending on opinion provider behaviour, we choose to run experiments with a variety of different environments (consisting of different compositions of the opinion provider population, again as shown in Table 4.1). In each experiment every opinion provider interacts with every service provider a fixed number of times (10). This allows each opinion provider to build up an interaction history from which it can base its opinion when needed. Each model, TRAVOS and BRS, is evaluated in each environment by means of an experiment consisting of a series of independent episodes. The number of times a consumer asks an opinion provider for an opinion increases in each episode, up to 200.

Figure 4.2 shows the mean estimation error of TRAVOS and BRS with these different opinion provider populations. To provide a benchmark, the figure also shows the mean estimation error of a consumer $c_{0.5}$, which keeps $\tau_{c_{0.5},s} = 0.5$ for all $s \in \mathcal{S}$. Results are plotted against the number of previous interactions that have occurred between the consumer and each opinion provider, representing the number of times a consumer has asked the opinion provider for an opinion.

As can be seen, in populations containing lying agents, the mean estimation error of TRAVOS is consistently equal to or less than that of BRS. Moreover, estimation errors decrease significantly for TRAVOS as the number of consumer to opinion provider interactions increase. In contrast, BRS's performance remains constant, since it does not learn from past experience. Both models perform consistently better than $c_{0.5}$ in populations containing 50% or 0% liars. However, in populations containing only lying sources, both models are sufficiently misled to perform worse than $c_{0.5}$, but TRAVOS suffers less from this effect than BRS. Specifically, when the number of past consumer to opinion provider interactions is low, TRAVOS benefits from its initially conservative belief in opinion providers' opinions. The benefit is enhanced further as the consumer becomes more skeptical with experience.

Similar results can be seen in populations containing noisy sources. In general, the performance

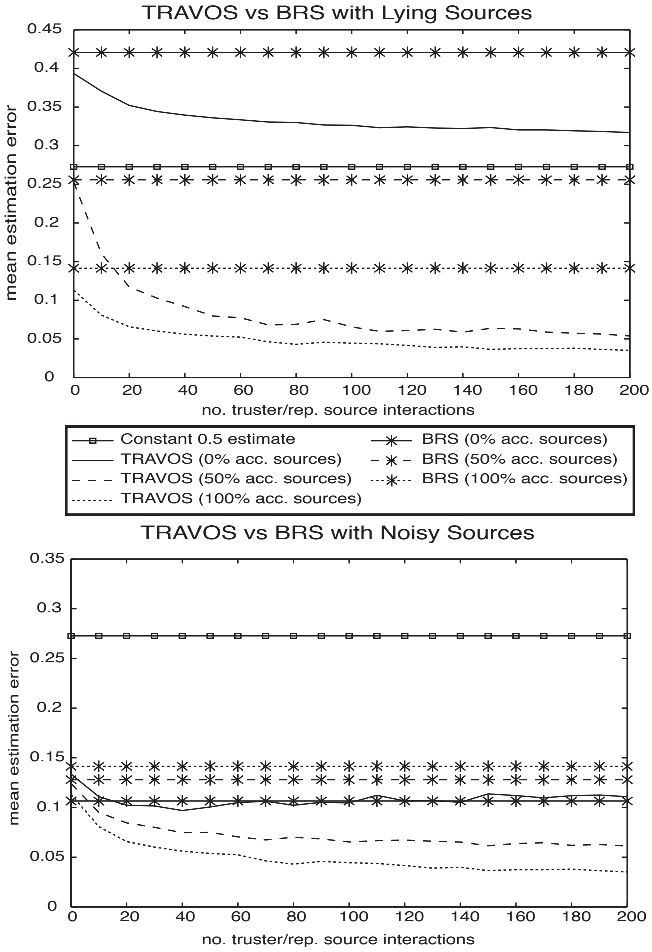

FIGURE 4.2: TRAVOS reputation system vs BRS.

of both models is better because noisy opinions are not as misleading as lying opinions on average. TRAVOS still outperforms BRS in most cases, except when the population contains only noisy sources. In this case, BRS has a small but statistically significant advantage when the number of consumer to reputation source interactions is less than 10.

4.1.4 Summary

The empirical evaluation described in this chapter considers two main aspects of TRAVOS. Firstly, the experiments evaluate the performance of the different components of TRAVOS in different environments. The key findings in this part of the investigation are that over time a consumer agent using only its own direct experiences as a means of evaluating the level of trust in another improves its estimation of another's behaviour as it gains more experience. Therefore, by using TRVAOS, an agent entering a new system is able to improve its estimates of other agents' behaviour as it gains more experience. Furthermore, in some cases using both personal experience and opinions of others was marginally worse than just using only others opinions.

Secondly, the experiments evaluate the performance of TRAVOS with a similar model (namely

BRS), showing that in the simulated environments and under most experimental conditions TRAVOS performs better. The key finding here is that even though the models perform similarly when there are few interactions between the truster and the opinion provider, as the number of interactions increase, TRAVOS performs significantly better than BRS.

Overall the evaluation of TRAVOS is positive. However, this evaluation has been carried out in a simple simulated system, and it does not assess the ability of TRAVOS to be applied to a realistic industrial application. To this end, we follow the empirical investigation by carrying out a system evaluation of TRAVOS, which is described in more detail in the following section.

4.2 System Evaluation

The general aim of the research presented in this thesis is to develop a model that allows an agent to make sound decisions in light of the inherent uncertainty in large, open systems. However, we believe that showing that the model works (better than some other approaches) in a simulated environment is not sufficient to show that we have achieved our aims, and it is necessary to carry out a system evaluation of the model, which will apply the model to a realistic application.

The system evaluation of TRAVOS takes place within the context of an agent-based virtual organisation (VO) system, designed for the Grid. In more detail, the context is provided by the CONOISE-G Project (Patel et al., 2006), which aims to address the automated formation and management of VOs in an open environment, and develop a set of techniques to ensure that VOs are agile and resilient. Whereas its predecessor, CONOISE (Norman et al., 2003) focused on traditional problems like resource management and bidding strategies, CONOISE-G focuses on contract management, trust between VO participants and policing of contracts. More specifically, the project aims to make the following contributions.

1. Develop techniques for modelling, using, and maintaining trust in virtual organisations where partial knowledge is pervasive.
2. Develop theory and mechanisms for defining and enforcing social laws in virtual organisations, including the policing of stake-holders' behaviour.
3. Develop representations for quality policies that support accreditation of service-providers, and maintenance of relationships between service-providers and consumers.

We begin this section by describing an agent-based VO scenario in the context of the CONOISE-G environment (Section 4.2.1). Following this, in Section 4.2.2, using the scenario as a framework, we describe in detail how TRAVOS is used in the decision-making process of agents participating in VO formation and restructuring. We conclude the system evaluation (in Section 4.2.3), by describing how TRAVOS is implemented in the CONOISE-G environment.

4.2.1 An Agent-Based Virtual Organisation Scenario

This section describes a scenario in the CONOISE-G environment with the aim of contextualising the use of trust in such an environment. A screenshot of the CONOISE-G system is shown in Figure 4.3, which shows the two main screens that allow a user to examine the state of the system.

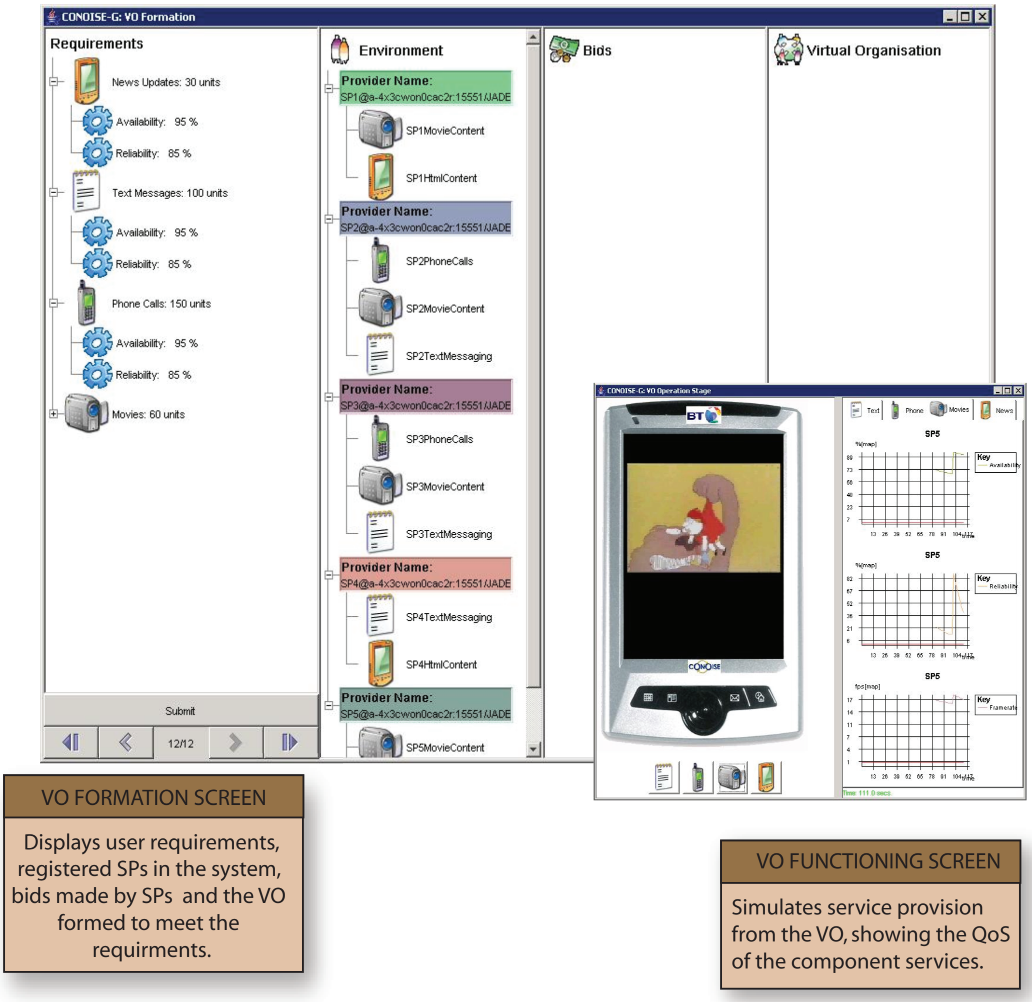

FIGURE 4.3: A Screenshot of the CONOISE-G system.

The overall scenario is as follows:

"A user wants to purchase and receive a monthly movie subscription package on his PDA/phone, and a monthly news service. He also wants a monthly package for his PDA/phone that includes a certain number of free text messages and a minimum number of free minutes per month".

The agents that are found throughout the scenario are described below, and the main interactions between them can be seen in Figure 4.4:

1. **Yellow Pages Agent (YP)** – This agent provides a *lookup* service, giving any agents the names/addresses/service descriptions of other agents upon request.

2. **User Agent (UA)** – This agent provides the initial *bootstrap* request for the Virtual Organisation Manager agent.

3. **Virtual Organisation Manager Agent (VOM)** – This agent initiates the request for a service on behalf of a user-initiated request. It represents a *front-end* that the user can query and ask for a specific required service. It then forms a VO with other agents to provide that service, and manages the resulting VO.

4. **Service Provider Agent (SP)** – This agent provides one or more services that are sought by the user. The agent may provide the service in a number of different ways when a request for a service comes to it.

   (a) It may provide the service itself.

   (b) It may provide the service by using a VO it currently belongs to.

   (c) It may delegate the service provision to other agents.

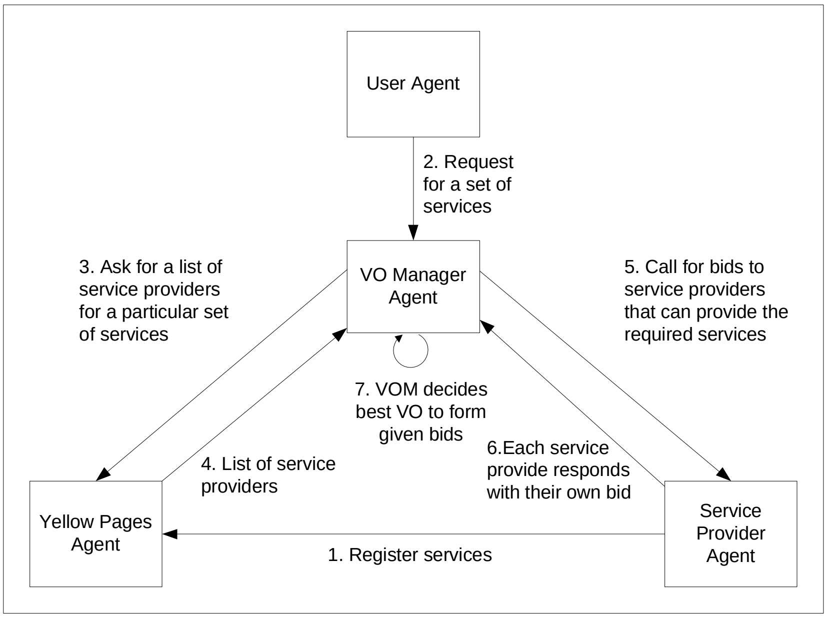

FIGURE 4.4: The main agents in the scenario, and the interactions between them.

This scenario can be broken down into four main stages — system initiation, user querying, VO formation and VO restructuring — each of which involves different agents and interactions between these agents. The end result of this scenario is a system state in which there is a virtual organisation that meets the requirements of the user.

The rest of this section describes in detail the stages in seven discrete steps needed to achieve the requirements that the user requested, with emphasis on the need for trust in the environment.

Stage 1: System Initiation — The system is initiated.

Prior to any input or request from a user, the system must be initiated. In this phase, all the necessary agents are created and undergo a registration process with the YP agent.

Stage 2: User Query — Query is entered into the system.

Once the initial booting and registering process is complete, the system is ready. We assume that it has been operating for some time so that there is some interaction history in the system. After a while, the system receives a request by a user who wants to purchase and receive a monthly movie subscription package on his PDA, and a monthly news service. He also wants a monthly package for his PDA that includes a certain number of free text messages and a minimum number of free minutes per month. The system receives this query from a human user via the UA. It is important that the user *trusts* the UA and believes that the UA will successfully begin the process, which will result in the user receiving the services he requires.

Stage 3: User Query Forwarding — User agent forwards the query to a VOM agent.

Once the UA receives the query, it translates it into a common language that all agents in the system understand. It then evaluates the possible VOM agents to which it can delegate the responsibility of forming an appropriate VO to meet the user's requirements. The UA uses its past interaction history (if it has any) in order to identify possible VOM agents and determine their *trustworthiness*. It then selects the most *trusted* VOM and forwards the user specification to the VOM agent.

Stage 4: VO Formation (a) — VOM identifies which agents can provide appropriate services.

In this stage the VOM uses the YP to identify which SP agents can provide the services required by the user. The VOM must *trust* that the YP agent is not colluding with a subset of the SP agents in order to receive some personal gain.

Stage 5: VO Formation (b) — VOM sends a call for bids to the SP agents and the SP agents respond.

Here the VOM issues a call for bids to all of the SPs that are capable of providing part of or the entire services requested by the user. In response, the SP agents decide whether to bid. The three main factors that influence this decision are:

1. Does the SP agent provide this specific service?
2. Given the current workload, is it possible to provide the service when required?
3. Does the SP *trust* the VOM enough to form a VO with it if the bid is accepted?

Stage 6: VO Formation (c) — VOM uses the pool of received bids to form a VO to serve the end user.

Upon receiving all the bids from the SP agents, the VOM evaluates each with respect to two factors:

1. What is the utility gain or loss that it will receive if a particular bid was selected?
2. How *trustworthy* is the SP agent or group of SP agents (recall that a SP agent can use the capabilities of its existing VO to form a bid) that the bid comes from?

Having evaluated each bid with the above two factors, the VOM forms a VO with the most *trusted* agents that will maximise its own utility.

Stage 7: VO Restructuring — Change in the VO structure.

After the VO begins functioning, suppose a new agent, offering the same services as one of the members of the VO, enters the system. This agent offers a better quality of service at a lower price and is *very trustworthy*, giving more utility to the VOM. At the same time the agent in the VO that provides the same service decides to stop providing the service. The VOM has a need to fill this vacant space in the VO. Unaware of the characteristics of the new agent, the VOM requests a bid from this unknown agent, and when it receives the bid it evaluates it with respect to utility gain/loss and trustworthiness. However, since the VOM has not interacted with it before, it will not be able to accurately judge the *trustworthiness* of the new SP. The VOM therefore consults other agents in the system to develop an assessment of the *reputation* of the unknown agent, and uses it to calculate its *trustworthiness*. Assuming this is adequate, the existing member of the VO is replaced with this new agent, resulting in a more *trusted* VO.

4.2.2 Applying TRAVOS in the Scenario

In this section we demonstrate exactly how the model presented in Chapter 3 is used in the scenario. There is a variety of agents in the scenario, but we concentrate on the stages that involve the User Agent, the VO Manager Agent and a number of Service Provider Agents (VO formation stages 4, 5 and 6 from the scenario). More specifically, we use the following agents:

- One User Agent, a_{ua1}.
- Four VO manager Agents, a_{vom1}, a_{vom2}, a_{vom3} and a_{vom4}.
- Six Service Provider Agents (for simplicity, we assume that each agent in the system has the ability to provide only one service):
  - a_{sp1}, a_{sp2} and a_{sp3} providing the phone call service.

– a_{sp4}, a_{sp5} and a_{sp6} providing the HTML content service.

We begin by stating that there is a need to create a VO to meet a specific requirement to provide a composite multimedia communication service to an end user. The composite service consists of the following basic services: HTML content provision and phone calls.

Firstly, a_{ua1} sends the user's requirements as a query to a_{vom1}. Agent a_{vom1} parses the query and realises that it can only meet these requirements by forming a VO with an agent that can supply a service for phone calls and an agent that can supply a service for HTML content. Agent a_{vom1} is aware of three agents that can provide a phone call service, and its interaction history with these is shown in Table 4.2. Similarly, it is aware of three agents that are capable of providing HTML content, and its past interactions with these entities are given in Table 4.3.

Agent	**Past interactions**	
	Successful	**Unsuccessful**
a_{sp1}	17	5
a_{sp2}	2	15
a_{sp3}	18	5

TABLE 4.2: Agent a_{vom1}'s interaction history with phone call service provider agents.

Agent	**Past interactions**	
	Successful	**Unsuccessful**
a_{sp4}	9	14
a_{sp5}	3	0
a_{sp6}	18	11

TABLE 4.3: Agent a_{vom1}'s interaction history with HTML content service provider agents.

Agent a_{vom1} would like to choose the most trustworthy phone call and HTML content service provider. The following describes how this is achieved using TRAVOS. Before we calculate which of the possible candidates are the most trustworthy, we must specify certain parameters that a_{vom1} requires. First, we specify the level of error that a_{vom1} is willing to accept when determining the confidence in a calculated trust value as $\epsilon = 0.2$. Second, we specify that $\theta_\gamma = 0.95$, below which a_{vom1} will seek other opinions about the trustee.

4.2.2.1 Calculating Trust and Confidence

Using the information from Tables 4.2 and 4.3, a_{vom1} can determine the number of successful interactions n, and the number of unsuccessful interactions m, for each agent it has interacted with. Feeding these into Equation 3.5, a_{vom1} can obtain shape parameters for a beta distribution function that represents the behaviour of each service provider agent. For example, the shape parameters α and β, for a_{sp1}, are calculated as follows:

Using Table 4.2: $n_{a_{vom1},a_{sp1}} = 17$, $m_{a_{vom1},a_{sp1}} = 5$.

Using Equation 3.5: $\alpha = 17 + 1 = 18$ *and* $\beta = 5 + 1 = 6$.

The shape parameters for each agent are then used in Equation 3.6 to calculate a direct trust value for each agent that a_{vom1} is assessing. For example, the trust value $\tau^d_{a_{vom1},a_{sp1}}$ for a_{sp1} is calculated as follows:

Using Equation 3.6: $\tau^d_{a_{vom1},a_{sp1}} = \frac{\alpha}{\alpha+\beta} = \frac{18}{18+6} = 0.75$.

For a_{vom1} to be able to use the trust values it obtains for each agent, it must also determine the confidence it has in the calculated trust value. This is achieved by using Equation 3.19 and ϵ (which is 0.2). For example, the confidence γ_{a_vom1,a_sp1} that a_{vom1} has in the trust value $\tau^d_{a_{vom1},a_{sp1}}$ is calculated as shown below:

Using Equation 3.19:

$$\gamma_{a_{vom1},a_{sp1}} = \frac{\int_{\tau^d_{a_{vom1},a_{sp1}}+\epsilon}^{\tau^d_{a_{vom1},a_{sp1}}-\epsilon} B^{\alpha-1}(1-B)^{\beta-1}dB}{\int_0^1 U^{\alpha-1}(1-U)^{\beta-1}dU} = \frac{\int_{0.95}^{0.55} B^{\alpha-1}(1-B)^{\beta-1}dB}{\int_0^1 U^{\alpha-1}(1-U)^{\beta-1}dU} = 0.98$$

Agent	α	β	τ_{a_{vom1},a_x}	γ_{a_{vom1},a_x}
a_{sp1}	18	6	0.75	0.98
a_{sp2}	3	16	0.16	0.98
a_{sp3}	19	6	0.76	0.98
a_{sp4}	10	15	0.40	0.97
a_{sp5}	4	1	0.8	0.87
a_{sp6}	19	12	0.61	0.98

TABLE 4.4: Agent a_{vom1}'s calculated trust and associated confidence level for HTML content and phone call service provider agents.

The shape parameters, trust values and associated confidence for each agent, a_{sp1} to a_{sp6}, which a_{vom1} computes using TRAVOS, are shown in Table 4.4. From this, it is clear that the trust values for agents, a_{sp1}, a_{sp2} and a_{sp3}, all have a confidence above θ_γ (0.95). This means that a_{vom1} does not need to consider the opinions of others for these three agents. Agent a_{vom1} is able to determine that a_{sp3} is the most trustworthy out of the three phone call service provider agents and chooses it to provide the phone call service for the VO.

4.2.2.2 Calculating Reputation

The process of selecting the most trustworthy HTML content service provider is not as straightforward. Agent a_{vom1} has calculated that out of the possible HTML service providers, a_{sp5} has the highest trust value. However, it has determined that the confidence it is willing to place in this value is 0.87, which is below θ_γ and means that a_{vom1} has not yet interacted with a_{sp5} enough times to calculate a sufficiently confident trust value. In this case, a_{vom1} has to use the opinions from other agents that have interacted with a_{sp5}, and form a reputation value for a_{sp5}

that it can compare to the trust values it has calculated for other HTML providers (a_{sp4} and a_{sp6}).

Let's assume that a_{vom1} is aware of three agents that have interacted with a_{sp5}, denoted by a_{vom2}, a_{vom3} and a_{vom4}, whose opinions about a_{sp5} are $(15, 46)$, $(4, 1)$ and $(3, 0)$ respectively[2]. Agent a_{vom1} can obtain beta shape parameters based solely on the opinions provided, by using Equations 3.8 and 3.7, as shown below:

Opinions from providers: $a_{vom2} = (15, 46), a_{vom3} = (4, 1)$ and $a_{vom4} = (3, 0)$

Using Equation 3.7: $N = 15 + 4 + 3 = 22, M = 46 + 1 + 0 = 47$

Using Equation 3.8: $\alpha = 22 + 1 = 23, \beta = 47 + 1 = 48$

Having obtained the shape parameters, a_{vom1} can obtain a trust value for a_{sp5} using Equation 3.6, as follows:

Using Equation 3.6: $\tau^r_{a_{vom1},a_{sp5}} = \frac{\alpha}{\alpha+\beta} = \frac{23}{23+48} = 0.32$

Now a_{vom1} is able to compare the trust in agents a_{sp4}, a_{sp5} and a_{sp6}. Before calculating the trustworthiness of a_{sp6}, agent a_{vom1} considers a_{sp5} to be the most trustworthy (see Table 4.4). Having calculated a new trust value for agent a_{sp5} (which is lower than the first assessment), agent a_{vom1} now regards a_{sp6} as the most trustworthy. Therefore a_{vom1} chooses a_{sp6} as the service provider for the HTML content service.

4.2.2.3 Coping With Inaccurate Opinions in the VO

The method a_{vom1} uses to assess the trustworthiness of a_{sp6}, as described in Section 4.2.2.2, is susceptible to errors caused by reputation providers giving inaccurate information (as discussed in Section 3.4.2). In our scenario, suppose a_{vom2} provides the HTML content service too, and is in direct competition with a_{sp6}. Agent a_{vom1} is not aware of this fact, which makes a_{vom1} unaware that a_{vom2} may provide inaccurate information about a_{sp6} to influence its decision on which HTML content provider agent to incorporate into the VO. If we look at the opinions provided by agents a_{vom2}, a_{vom3} and a_{vom4}, which are $(20, 46)$, $(4, 1)$ and $(3, 0)$ respectively, we can see that the opinion provided by a_{vom2} does not correlate with the other two. Agents a_{vom3} and a_{vom4} provide a positive opinion of a_{sp6}, whereas agent a_{vom2} provides a very negative opinion. If a_{vom2} is providing an inaccurate account of its experiences with a_{sp6}, we can use the mechanism discussed in Section 3.4.2 to allow a_{vom1} to cope with this inaccurate information, and arrive at a better decision that is not influenced by self-interested reputation providing agents (such as a_{vom2}).

Before we show how TRAVOS can be used to handle inaccurate information, we must assume the following. Agent a_{vom1} obtained reputation information from a_{vom2}, a_{vom3} and a_{vom4} on

[2]Opinions can be represented as a pair consisting of the number of successful and unsuccessful interactions the opinion provider says it has had with the trustee.

Agent	Weighting	Adjusted Values			
		μ	σ	α	β
a_{vom2}	0.0039	0.5	0.29	1.0091	1.0054
a_{vom3}	0.78	0.65	0.15	5.8166	3.1839
a_{vom4}	0.74	0.62	0.17	4.3348	2.6194

TABLE 4.5: Agent a_{vom1}'s adjusted values for opinions provided by a_{vom2}, a_{vom3} and a_{vom4}.

	$[0, 0.2]$		$[0.2, 0.4]$		$[0.4, 0.6]$		$[0.6, 0.8]$		$[0.8, 1]$		**Total**
	n	**m**	**n**	**m**	**n**	**m**	**n**	**m**	**n**	**m**	
a_{vom2}	2	0	11	4	0	0	0	0	2	3	25
a_{vom3}	0	2	1	3	0	0	22	10	6	4	30
a_{vom4}	1	3	0	2	0	0	18	8	5	3	25

TABLE 4.6: Observations made by a_{vom1} given opinion from a reputation source. *n* represents that the interaction (to which the opinion applied) was successful, and likewise *m* means unsuccessful.

several occasions, and each time a_{vom1} recorded the opinion provided by a reputation provider and the actual observed outcome (from the interaction with an agent to which the opinion is applied). Each time an opinion is provided, the outcome observed is recorded in the relevant bin. Agent a_{vom1} keeps information of like opinions in bins as shown in Table 4.6.

Using the information shown in Table 4.6, agent a_{vom1} can calculate the weighting to be applied to the opinions from the three reputation sources by applying the technique described in Section 3.4.2.3. In so doing, agent a_{vom1} uses the information from the bin, which contains the opinion provided, and integrates the beta distribution between the limits defined by the bin's boundary. For example, a_{vom2}'s opinion falls under the $[0.2, 0.4]$ bin. In this bin, agent a_{vom1} has recorded that $n = 15$ and $m = 3$. These n and m values are used to obtain a beta distribution, using Equations 3.13 and 3.14, which is then integrated between 0.2 and 0.4 to give a weighting of 0.0039 for a_{sp6}'s opinion. Then, by using Equations 3.13 and 3.14, agent a_{vom1} can calculate the adjusted mean and standard deviation of the opinion, which in turn gives the adjusted α and β parameters for that opinion. The results from these calculations are shown in Table 4.5.

Summing the adjusted values for α and β from Table 4.5, a_{vom1} can obtain a more reliable value for the trustworthiness of a_{sp5}. Using Equation 3.4, a_{vom1} calculates a trust value $= 0.62$ for a_{sp5}. This means that from the possible HTML content providers, a_{vom1} now sees a_{sp5} as the most trustworthy and selects it to be a partner in the VO. Unlike a_{vom1}'s decision in Section 4.2.2.2 (when a_{sp6} was chosen as the VO partner), here we have shown how a reputation provider cannot influence the decision made by a_{vom1} by providing inaccurate information.

4.2.3 Implementing TRAVOS

To test the applicability of the TRAVOS model to an agent-based virtual organisation, we choose to implement and integrate it within such a context. Again, this context is provided by the

CONOISE-G system. In this case the environment consists of numerous agents on a JADE platform (Bellifemine et al., 2001). Inter-agent communication is achieved using RDF and a shared ontology. In the system, agents can dynamically form VOs, which are resilient against potential perturbations. When agents form a VO they enter into a service level agreement (SLA) with the VO Manager agent. Over the lifespan of the VO, the system shows how the VO Manager monitors the level of service provision from its VO members using the infrastructure agents. If an SLA is broken, then the necessary action is taken to replace the faulty member with another agent.

The trust subsystem (TRAVOS implementation) is made up of two main parts: the trust component, which is internal to all agents, and reputation broker agents, which are reputation providing agents. Figure 4.5 shows the overall system architecture, and in particular the main parts of the trust system, including the trust component, which is internal to all agents, and the reputation broker agents, which are reputation providing agents.

The remainder of this section gives a detailed description of the trust system. It identifies the major components of the system and details interfaces between them.

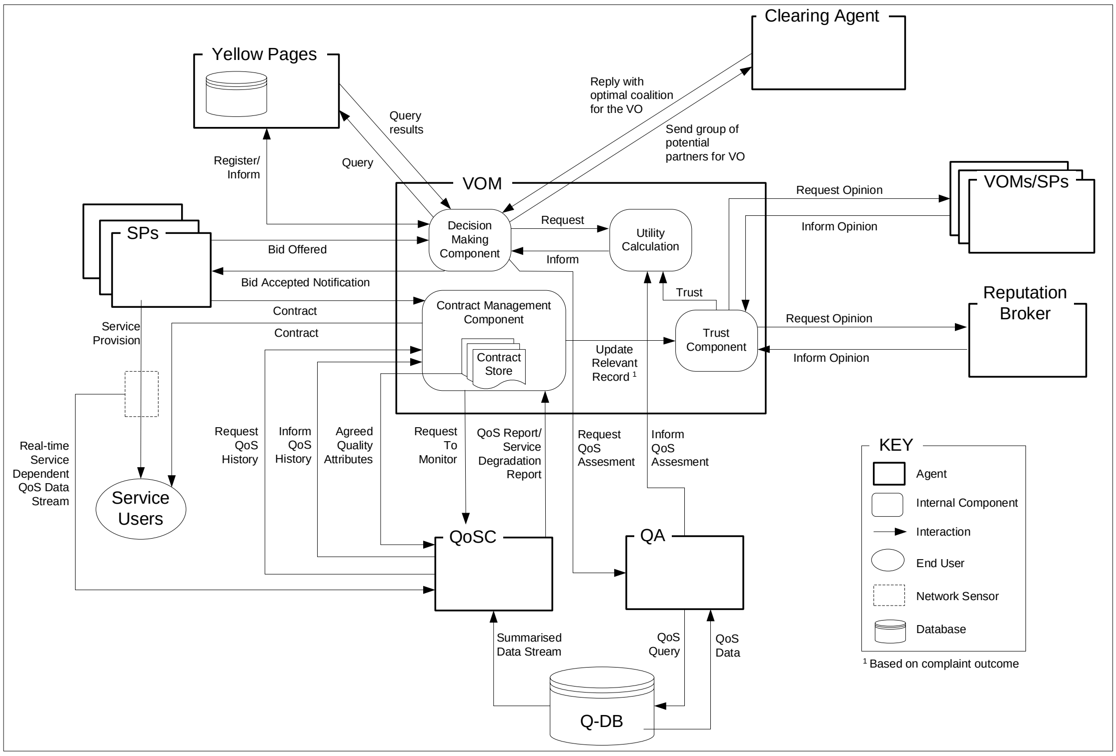

FIGURE 4.5: CONOISE-G system architecture.

4.2.3.1 Trust Component Design

The trust component is internal to agents that require a trust metric in their decision-making process, and is insulated from the external environment and other agents by the agent that embodies it. The component receives requests and messages from the agent that owns it, and if the need arises for the trust component to interact with the external environment, it does so through its owner.

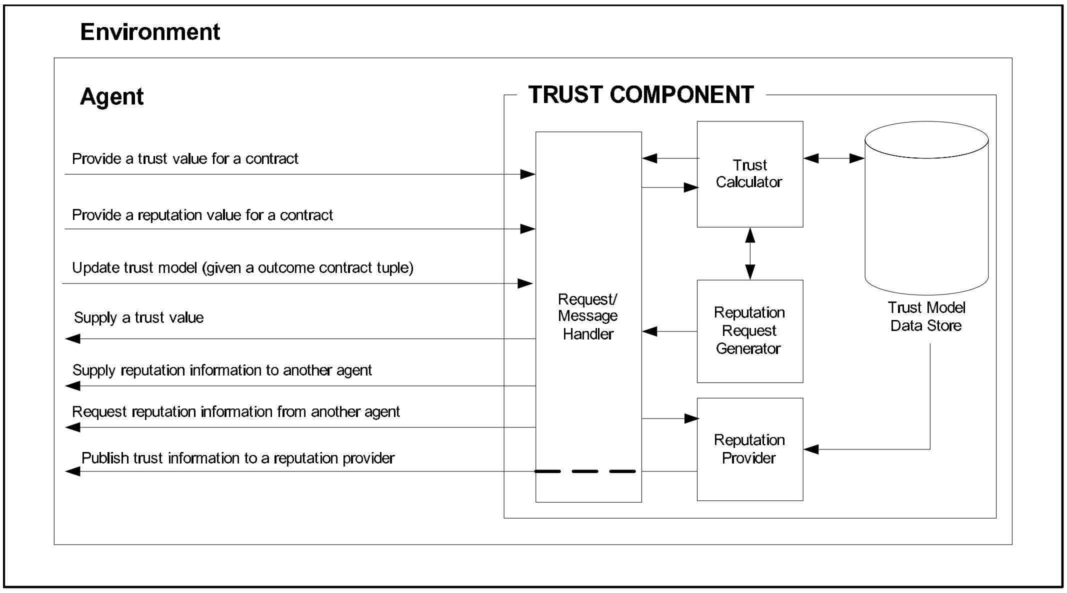

FIGURE 4.6: The subsections of the trust component and their interactions

In more detail, Figure 4.6 shows the main messages and requests that the trust component is capable of handling, and the main messages and requests that it generates. The diagram also shows the high level design of the main parts of the component, and the interactions that exist between them. These are described below:

- **Request/Message Handler:** This is the interface of the component to the external world; it correctly identifies the incoming and outgoing communications, and deals with them accordingly.
- **Trust Calculator:** This is the heart of the component and uses the TRAVOS model described in Chapter 3 to convert a collection of outcomes of past interactions into a probability representing the trustworthiness of an agent. The calculator draws relevant data for the trust calculation from the internal data store, which holds an interaction history for the agent. If the confidence in the data available does not meet a set threshold, then it uses the Reputation Request Generator to request reputation information from peers or brokers. If the agent is confident in the trust it derives from its own data then it would not use reputation information. This is because reputation information is subjective to the provider, and also may not be reliable. Therefore, if the confidence in one's own data is high, this should be the only source of information used to calculate a trust value. The trust calculator is responsible for providing the trust-related evaluation required in VO formation and restructuring (Stages 5 and 6 in the scenario presented in Section 4.2.1).

- **Reputation Request Generator:** This generates a request for other agents (peers or reputation brokers) to provide reputation information. It is used in the VO restructuring stage of the scenario described in Section 4.2.1.
- **Reputation Provider:** This services the request for reputation information that the agent may receive from other agents requiring information about a certain agent.

Having described the intra-agent components of the TRAVOS implementation, the next section describes how these components are used by an agent to calculate trust values for others in a number of different ways. This includes an agent using internal components internal or by interacting with other agents that share similar components.

4.2.3.2 Interaction Protocols

This section describes the communication protocols between the components of the system in the following scenarios:

1. Obtaining a trust value from direct interaction alone (User query and VO formation stages of the agent-based virtual organisation scenario, Stage 3 and Stage 5 in Section 4.2.1).
2. Obtaining reputation information from peers (VO restructuring stage of the agent-based virtual organisation scenario, Stage 7 in Section 4.2.1).
3. Obtaining reputation information from reputation brokers (VO restructuring stage of the agent-based virtual organisation scenario, Stage 7 in Section 4.2.1).

Direct Interaction Based Trust

The agent does not need to interface with any other agents in this scenario. Figure 4.7 shows how the trust component responds to a request from the decision-making component (within the agent) to provide a utility for a given contract. The trust component can serve this request by using the prior values available to the agent, or by using the past interaction history with the partner in question.

Obtaining Reputation Information from Peers

Figure 4.8 shows message-passing between agents in order to exchange reputation information. This occurs when the trust component does not have enough of an interaction history to serve the request made by the decision-making component, or when the confidence in the trust value that it generates is very low. If the agent that is being asked for trust information does not have enough information, then it returns a failure message, otherwise it provides the requester with the appropriate information.

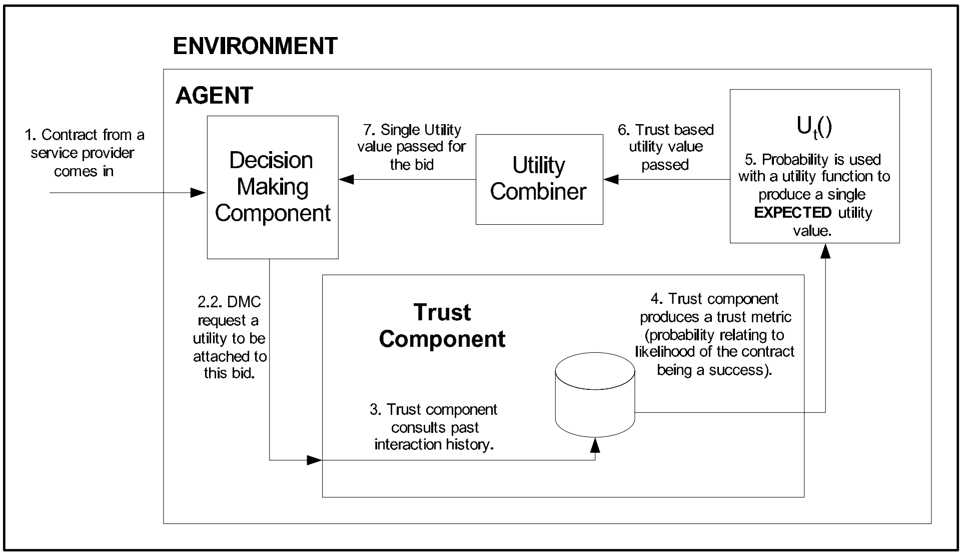

FIGURE 4.7: Direct interaction based trust.

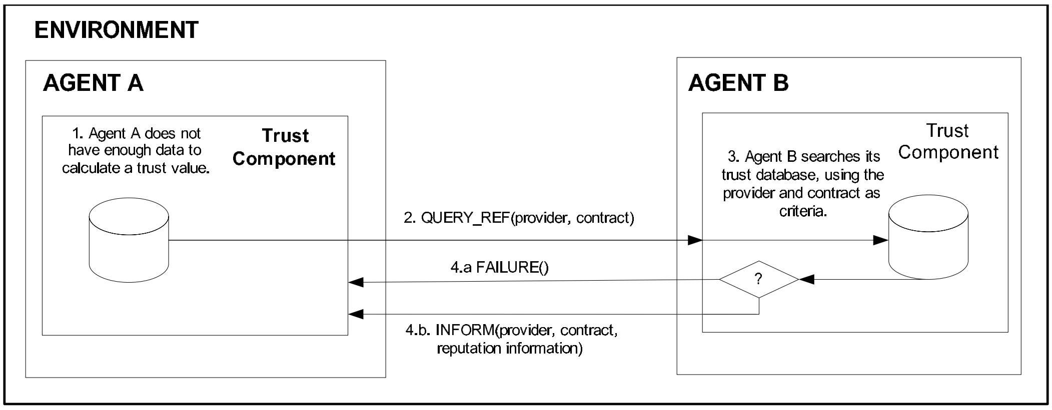

FIGURE 4.8: Obtaining reputation information from peers.

Obtaining Reputation Information from Reputation Brokers

In response to the issue of scalability, we propose the use of several broker agents to facilitate the propagation and storage of reputation information in an accessible manner. The mechanism is simple and serves its purpose, but we are aware that there may be complications in other environments where there may be too many agents present.

Before reputation information can be obtained from reputation broker agents, the broker agents must obtain the information from the distributed agents. We propose that a subscription mechanism be used, where reputation brokers subscribe to trust information messages from agents in their domain. Each agent then provides trust information to the broker agent at a set time interval, as shown in Figure 4.9.

Figure 4.10 shows how the reputation broker services requests, by other agents, to provide trust information. This mechanism is similar to the reputation information provision by peers.

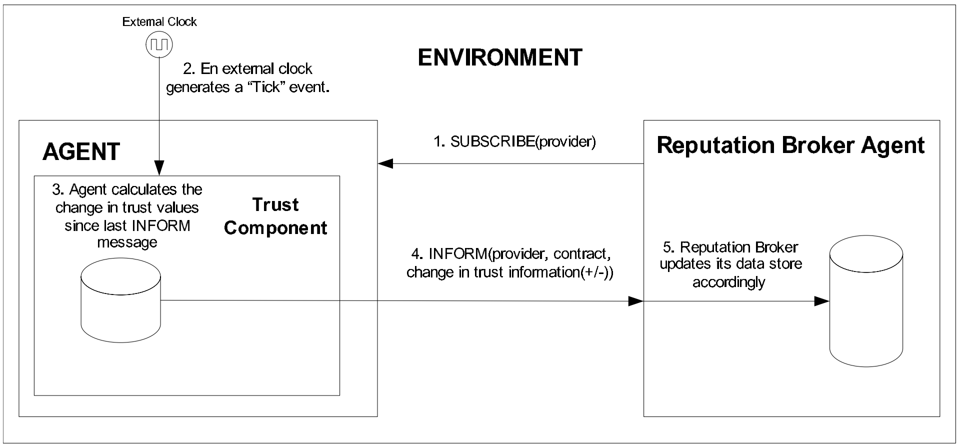

FIGURE 4.9: Populating the reputation brokers — A subscription mechanism.

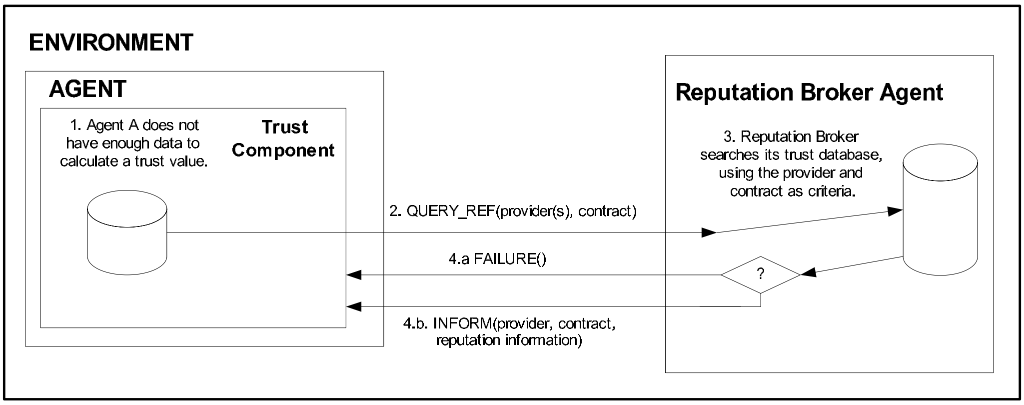

FIGURE 4.10: Obtaining reputation information from the reputation brokers.

4.3 Summary

In this chapter we have evaluated TRAVOS in two different ways, and in doing so we have shown that our model does allow agents to make better decisions in agent-based VO systems.

Firstly, through empirical evaluation we have demonstrated that our probabilistic approach is effective. Furthermore, we have shown that an agent using our exogenous opinion filtering method is able to outperform one employing an endogenous approach. Through the evaluations, ultimately, we have shown that the mechanism we have designed, not only performs, but perform better than other approaches.

Secondly, the system evaluation of TRAVOS has showed that the model can be applied to an application that supports the dynamic formation and management of agent-based VOs. The system evaluation was carried out in the context of an application and, whilst the boundaries of the system were not entirely open, the system was far more complex than the simulated environment used for the empirical evaluation. Whereas the empirical evaluation supported the validity of the model to be used for estimating the trustworthiness of others, the system evaluation has shown that the model can be implemented and used by an agent-based system which requires agents to assess the trustworthiness of potential partners.

This evaluation chapter has shown many positive aspects of our model, but it does highlight the fact that the model falls short of one of our key aims. The current model does not factor, into the trust calculations, any social information that may exist in the environment (Aim 6 in Section 1.3). In the next chapter, we extend the state of the art in the computational trust research domain by providing mechanisms that extend the basic TRAVOS model to include precisely such social information.

Chapter 5

Extending TRAVOS to Incorporate Social Relationships

This chapter considers how social information can be obtained by an agent, and, moreover, how it can then be incorporated into the trust calculation. At the end of Chapter 3 we highlighted the aims that were achieved by the TRAVOS model presented in that chapter. However, we also identified that the TRAVOS model lacked the incorporation of social information in the trust model. This limitation is addressed in this chapter. Here, we present an extension to the TRAVOS model that enables it to find social information and to subsequently use it in its trust calculation. Specifically, the extension, called TRAVOS-R (for TRAVOS Relationships), contains mechanisms that allow an agent to learn the social relationships that exist between agents in a multi-agent system and then factor these into its calculations.

The chapter begins by providing motivation for using social relationships in trust calculations, and discusses how such information can be used in a trust model (Section 5.1). We then present an analysis of the type of interactions that occur between agents in a multi-agent system, with the aim of illustrating the main types of relationships between a pair of agents. We ground our analysis in a relationship-based scenario in Section 5.3. The mechanisms that form TRAVOS-R are presented in two main parts. Firstly, we present the mechanisms concerned with identifying and learning the type of relationship present between a pair of agents (Section 5.4). Secondly, we present a set of relationship-based heuristics which show how knowledge about the presence of a particular type of relationship can be used to adjust biased opinions given by providers in an attempt to mislead the truster. This chapter presents only the theory for TRAVOS-R, and does not demonstrate the superiority of using social information in trust calculations. It is in the following chapter (Chapter 6) that we demonstrate this through empirical and system evaluation.

5.1 Background

Within a multi-agent system, agents are able to interact with others, and through these interactions they form a type of relationship, which dictate the manner in which agents behave towards each other. Furthermore, in most cases, the relationships that exist between the stakeholders of the agents in the real world are usually mirrored by the relationships that exist between the agents in the virtual world. For example, consider a scenario in which there are two real world companies capable of providing a telephony service, each one being represented in a virtual marketplace by its autonomous agent. Due to the competitive relationship between the real world companies, their agents are likely to have a competitive relationship in the virtual marketplace, leading the agents to behave in particular ways towards each other, such as bidding against each other in a particular auction, or propagating false information about the capabilities of the other. In the latter case, it is important for an agent who receives the misleading information to recognise the fact that the information may be misleading, because the information provider has a competitive relationship with the agent the information is about. From the observation of human interactions it is evident that there is an intrinsic connection between relationships and trust (Williams et al., 1988). However, in contemporary trust models (for example see discussion in Sections 2.4.2.3, 2.4.2.5 and 2.4.2.6) this connection is often omitted as the designers of such models choose to concentrate on the link between experience and trust, or the use of reputation to determine trust.

Like other contemporary trust models, TRAVOS makes use of direct experience and the experiences of others, with an interaction partner in determining the level of trust to be assigned to a particular individual. The limitation of TRAVOS, and others, as discussed above, is that it overlooks a key component of determining trust in a particular trustee, namely the intra-agent social relationships within a multi-agent system. In human societies, social relationships, such as whether an individual is a collaborator or a competitor, play a crucial role in determining the trustworthiness of others and, using this as inspiration, this chapter aims to incorporate social relationships into the existing TRAVOS model. In particular we wish to capture the type and strength of particular relationships that may exist between two agents. For example, two individuals may find themselves competing against each other, or collaborating with each other. In the former case they have a relationship type that is competitive, and in the latter a type that is collaborative. Furthermore, suppose that a pair of individuals are competing for the first time and another pair that often finds themselves competing. In the former case we can say that there may be a weaker form of competition than that which is present in the latter case. The result of this is a more socially tuned trust metric that is based not only on the performance of individuals but their place in society as well. This leads individual agents to more accurately predict the behavior of others in the society by observing their social relations, and to the possibility of not risking a potential loss in utility by actually interacting with them. There are three primary ways in which relationships may be used with the TRAVOS model: (i) setting prior information in the model, (ii) selecting (or ignoring) possible agents from whom to obtain opinions, and (iii) adjusting the actual opinion provided by another. The following sections discuss each of these in more detail.

5.1.1 Setting Prior Information

Relationships can be used in *setting* the priors of the basic TRAVOS model. TRAVOS assumes configurable parameters as priors to the model, which allow it to function if an agent has no interaction history. There is a range of capabilities which require prior information, but for the purpose of this work, we will concentrate on two main capabilities.

1. **Calculating the trustworthiness of a trustee.** When a truster has limited or no experience of a particular trustee it uses prior information to make the best decision it can given the uncertainty. In this case, the prior information is the belief the truster has about the trustee's trustworthiness in the absence of any evidence.

2. **Calculating the ability of an opinion provider to provide an accurate opinion.** A truster may require prior information to assess the reliability of the opinion given by a particular opinion provider. Here, the prior information represents the belief the truster has about the opinion provider's reliability in the absence of any evidence.

In TRAVOS, prior information is encoded in the form of a prior distribution, which represents the agent's knowledge and beliefs before observing any first hand evidence. Once an agent begins to obtain evidence, this prior distribution is updated to reflect what has been observed. The use of prior information in TRAVOS is limited in the following ways.

- **Manual configuration** — The system designer has to manually set the prior knowledge that an agent will use. Typically this involves an adhoc approach to bring about a particular outcome in the community of agents. This method is not very scalable, for example when many different manual configurations are needed.

- **Designer expertise** — If the system designer is to set the prior distribution in a model then they are likely to require a large amount of domain specific knowledge and a deep understanding of the decision-making process of the agent.

- **Domain specific** — Manually configured prior distributions are very domain specific and for this reason they are not portable.

- **Non-Distinctive** — Prior information is typically used at a universal level, and is applied in any situation where trust related evidence is absent. This means that even if an agent is aware of some other knowledge (for example the category of agent that the trustee belongs to) the prior information is used in the same manner as if it did not have this extra knowledge.

- **Static** — The prior knowledge that the agent applies, in situations where there is inadequate evidence, does not change based on expertise gained by the agent over time. For example, over time an agent can learn that the majority of agents in the system are not malicious and so it may benefit by using a prior that reflects this learning, instead of continuing to use a uniform distribution with which it was initially configured.

By using relationships we can enable TRAVOS to adjust trust priors automatically given the knowledge of inter-agent relationships within the agent society. For example, suppose that two service-providing agents belong to the same organisation (we can assume this leads to collaboration between them) and that have common goals, and one consumer agent wishes to hire one of the service providing agents. Now, suppose that the consumer agent asks one of the service providers for its opinion about the other service provider, and that collaboration between service providers incentivises them to provide exaggerated opinions about their partners. If the consumer is not aware that the service-providing agents are in a collaborative relationship then it may blindly use prior knowledge[1], subsequently taking the opinion provided and making no adjustments to it. However, if the consumer is aware of the relationship between the two service providers, then it can set its prior knowledge to reflect this and therefore make adjustments to cope with the fact that the service provider's opinion may be exaggerated.

5.1.2 Selecting the Best Source of Opinions

Relationship information can also be used to *filter* for appropriate and inappropriate agents from whom to request opinions regarding the trustee. Again, the inspiration for this comes directly from human societies, where we are unlikely to ask for an opinion about the capabilities of a person from their friends or family. Simply, this is because we assume that friends and family have an incentive to provide an exaggerated opinion, which leads us to believe that the person is more capable than they actually are.

As an example consider again the situation where there are two service providers and a consumer agent. In this case we assume that the consumer agent does request an opinion about one of the service providers from the other. Now, if the consumer is aware of the relationship and believes that it could lead the opinion provider to give an exaggerated opinion, then it may try to find other sources of opinion (mitigating the risk that a false opinion will be provided).

5.1.3 Adjusting Trust Levels and Opinions

Finally, relationship information can be used to adjust the final trust metric, or the opinions used in calculating them, by either increasing or decreasing the level of trust represented by them. By this, we mean that if an agent has knowledge of the presence of a particular relationship, and

[1] In most cases this prior knowledge is uniform and leads the agent to behave ignorantly.

knows that this relationship is likely to cause another to behave in a biased way, then it can take appropriate action to mitigate any risks caused by this behaviour.

To demonstrate this, we again use the example stated in the previous two sections. Here, the consumer is aware of the relationship between the two service providers. More specifically, the consumer knows that both the service providers are in collaboration, so that if any one of the two are asked to provide an opinion about the other, they are likely to provide a positive opinion regardless of the actual abilities of the other. Using this knowledge, the consumer agent is able to ask[2] for an opinion from one of the service providers about the other, and adjust the opinion accordingly.

To this end, we extend the basic TRAVOS model described in Chapter 3 to include relationship information in its trust calculations. In more detail, we equip agents using TRAVOS to learn relationships between other agents in the system, and then use this information to adjust opinions obtained from others. We choose to concentrate on using the relationship information in adjusting the opinions provided by others as this allows us to relax an underlying assumption of the filtering mechanism described in Section 3.4.2. In the TRAVOS model, we assumed that the opinions provided by an opinion provider are comparable regardless of trustee to which they apply. This allowed us to learn the opinion providing ability of agents over time. Here, we remove this assumption and say that an opinion provider is likely to provide certain types of opinions, dependent on the type of relationship it has with the trustee.

We begin this chapter by enumerating and analysing the types of relationships that can be found in a multi-agent system, and support this analysis with a virtual organisation based scenario. In light of the relationships identified through the analysis, we then present TRAVOS-R, a mechanism for learning such relationships and a set of relationship-based heuristics to adjust opinions.

5.2 Relationship Analysis

In human societies, relationships are a very difficult concept to characterise, and this problem is not simplified by abstracting to an agent-based system. Here, relationships between agents are difficult to define. Consequently they are created, classified, defined and modeled in a number of different ways in the literature (López and Luck (2002), Banerjee et al. (2000) and Ashri et al. (2005)). We therefore simplify the concept of a relationship to suit our needs and, for the purpose of this research, we define a relationship as "*a connection existing between two social agents having interactions with each other*" (adapted from (Trencansky and Cervenka, 2005)). This definition does not tell us about how relationships can impact the trustworthiness of and the opinion provided by another. In addition, we assume that all relationships have three particular characteristics.

[2]Assuming that the consumer cannot find a better and more reliable source of reputation information.

- **Type** — Each relationship between agents can be classified as belonging to a particular type. Consider two agents share a relationship, this relationship can be of type X or it can be of type Y, for example competitive or cooperative. This allows agents in a multi-agent system to have different types of relationships with different agents, much like individuals in the real world.

- **Strength** — In addition to a relationship having a particular type, it also has a strength. By this we mean that there are various degrees of each particular type of relationship. For example, a pair of agents may have a *weak* competitive relationship, or they may have a *strong* competitive relationship. Here, we do not present a taxonomy of different strengths as it is not our aim to do so, but a discussion of how this particular characteristic of relationships is modeled in TRAVOS-R is discussed later in Section 5.4.4.

- **Impact** — Lastly, a relationship has associated with it *an implicit and explicit impact on the trust-related decisions of the two interacting agents between whom this relationship exists*. (We discuss this in detail later in Section 5.5).

Before we move on to the discuss the impact of relationships on trust and how we will model the characteristics of relationships, it is important to enumerate possible relationship types that can exist between agents in a multi-agent system. To do this we examine the outcomes of interactions, undertaken to achieve a specific goal, between *rational* agents in a multi-agent system. Rationality in this sense implies that agents act in ways that maximise their individual utility gain (see discussion in Chapter 1). Outcomes of all interactions can then be classified by the amount of utility gained by the interaction partners after a particular interaction episode, and it is this classification from which relationships can be derived. The classification for two agents (A and B) interacting to achieve their own individual goals G_A and G_B is shown in Figure 5.1, in which five distinct classifications can be seen. It is important to note that these individual goals may be complementary (for example, one agent wishes to buy an item that the other is trying to sell) or conflicting (for example, both agents are bidding to win a particular service provision contract). We choose to concentrate only on how these outcomes may be a result of *bilateral* relationships, involving only two agents. The alternative would be to consider that the outcomes are a result of relationships which are *compounded* to form small social networks of more than two agents. However, the notion of compounded relationships introduces unnecessary complexity, and so we only consider bilateral relationships (in Chapter 7 we discuss how our work on relationships can be taken further). Each of the five classes is discussed in more detail below.

Class A: $Utility_A > 0, Utility_B \leq 0$

> The first class is one where, after achieving a goal, agent A gains some positive utility and agent B is left with no utility gain. Let's assume that both agents A and B are rational agents, each with the capabilities of achieving its own goals G_A and G_B respectively. In this case, the goals are conflicting; that is to say that if an agent

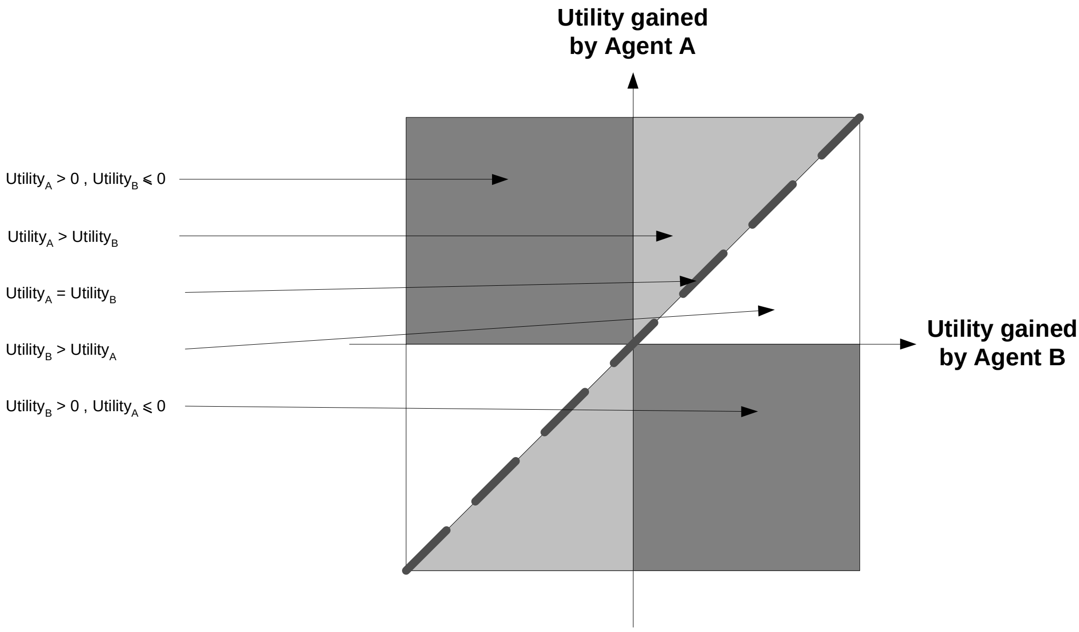

FIGURE 5.1: The utility gained by two interaction partners (agents A and B) after interacting to achieving a goal G.

achieves its own goal then the other cannot achieve it in that instance. Furthermore, if a rational agent can achieve the goal by itself, it will not share the reward for achieving it. This class of outcomes shows that in achieving its goals, A uses its capabilities and so earns the reward, and B misses out. This class of outcomes is thus a result of a *competitive* relationship.

Class B: $Utility_A > Utility_B > 0$

Here, both agents A and B gain some utility, but A gains more. This outcome is most likely when one of the agents is dependent on the other. Consider the example where B wishes to achieve the goal of providing a composite service (consisting of two services) to a third party, but it has only one of the component services. To achieve its goal, B seeks out another agent, A, to provide it with the other component service. Agent A's goal is to try and sell its service to the highest bidder. In this case, A has a stronger negotiation position (strategy), and during the negotiation, A is able to demand the majority of the reward that B is likely to gain from fulfilling the goal[3]. In some cases, this *upper hand* leads to A only getting an equal share of the reward, but benefiting in other ways such as earning credits (Rodrigues and Luck, 2006). If A agrees to provide the service so that B can achieve the goal, B becomes dependent on A's services. This class of outcomes

[3]Note: That this is most likely in monopolistic or oligopolistic scenarios where there is one major (or a few major) service providers so that the consumers have little choice.

is a result of a *dependence* relationship between A and B. The relationship is *directional*, unlike the competitive relationship described above, and in this class the dependence is from B towards A; that is to say that B depends on A.

Class C: $Utility_A = Utility_B$

Here the utility gained by agents A and B is equal. This outcome is most likely when two agents willingly cooperate to achieve a complementary goal and share the utility gained equally. This class of outcomes is a result of a *cooperative* relationship and, since both agents gain equally, the relationship is not directional.

Class D: $Utility_B > Utility_A > 0$

Class D is similar to class B, as both agents gain some utility, but agent B gains more. Therefore in this class it is agent B that has the upper hand in any interaction. Thus, we can say that these outcomes are a result of an *inverse* of the relationship that exists for Class B and, agent A depends on B.

Class E: $Utility_B > 0, Utility_A \leq 0$

Class E is similar to Class A, but it cannot be said that the the relationship that gives rise to these outcomes is the inverse of that which is present for Class A outcomes. The competitive relationship is not directional, but is *symmetric*; if two agents are competing in a zero-sum game then one has to win. Therefore both Class A and Class E are caused by competitive relationships.

It is evident from Figure 5.1 that for a given interaction the outcome is dependent on only one of the relationships described above. Although this simplifies the problem of linking relationship types to outcomes of interactions, it raises one question. Does the above classification capture all possible relationship types that can exist? Ashri et al. (2003) presents an alternative method of identifying relationships present in multi-agent systems. They use the manner in which agents interact with the environment to infer the relationships that exist in the environment. Building upon this generic relationship identification process, Ashri et al. (2005) use an agent-based market model to identify basic types of relationships in a multi-agent system. More specifically, they identify the following relationships, which they consider to be most prominent with regards to trust.

- **Trade** — A trade relationship exists between two agents that have traded in the same market. It is the most basic relationship type, and signifies that in a market certain agents have interacted at least once.

- **Dependency** — A dependency relationship exists between two agents, if one has a goal to sell items and the other has a goal to buy such items. Here, one agent is dependent on another selling particular items.
- **Competition** — A competitive relationship is one where two agents compete to either sell or buy the same items in the same market.
- **Collaboration** — A collaboration relationship exists between two agents when each agent is dependent on the other.
- **Tripartite** — A tripartite relationship is a combination of the above relationships formed between two or more agents.

Through our analysis we can see that our approach identifies a subset of their relationships, but does not identify two particular types. Firstly, our taxonomy does not consider a trade relationship. A trade relationship is simply a connection representing a past interaction between two agents. From a trust viewpoint, this information is useful when an agent needs to identify potential opinion providers, but the relationship is not strong enough to have an impact on trust by itself. A trade relationship can be seen as a *super-class* of relationship that can have many types (for example, competitive, cooperative and dependency). Secondly, our approach does not identify a tripartite relationship. Again, from a trust perspective, it is the basic types of competitive, cooperative and dependency relationships that have an impact on trust. Basic types exist between two agents, whereas tripartite relationship are composite social structures existing between a group of agents. In our work we do not consider complex relationships, as it is crucial to first understand how basic relationships can be incorporated into trust calculations (see Section 7.2 for a discussion on how our work can be extended to incorporate complex social structures). We argue that the relationships captured in our analysis are most influential on a trust model, and others that do exist, in virtual organisations, can be mapped to the three described in this section.

5.2.1 Transient Relationships

In human societies, two individuals may compete with, cooperate with, or depend on each other based on the context that they find themselves in. However, underlying these context dependent relationships there are stronger more *permanent* relationship bonds which dictate the behaviour of one individual towards another. For example, consider two individuals belonging to one university in which they are colleagues. The permanent relationship between them is cooperative, and it can be observed that they both work on the same project and co-author papers. On some occasions, however, they may also be seen competing, for instance in a departmental tennis competition. The manner in which the individuals behave towards each other is based on their permanent cooperative relationship, so that even when they compete (or appear as if competing) they do so amicably. Here, we assume that over time, due to the permanent cooperative relationship, there are many more instances of cooperation than competition. Furthermore, in all instances it is the permanent relationship that governs the behaviour of each individual towards the other. So, regardless of whether they find themselves competing or cooperating with each other, in this example the individuals behave amicably, honestly and respectfully towards each other.

Similarly in the case a multi-agent system, it is likely that two agents share more than one type of relationship over a number of interaction episodes. Consider two agents A and B which, in the context of selling a particular house, act as estate agents and compete to sell the house, and in the context of repairing a house, cooperate to carry out repairs to the house. In these two interaction episodes the agents share two different relationships. An agent may learn what type of relationship exists between two interacting agents for a particular interaction episode, but this information would be useless in a trust model because it is only valid for the interaction episode that just took place. Thus, due to the variability of relationships that can occur between the two interacting agents (given the many different contexts in which they can meet) it is necessary to model these *temporary* relationships (those occurring on a per interaction basis) as different from the permanent underlying relationship between the two interacting agents. We call the temporary relationships *transient* relationships.

This form of modelling makes the problem more complex and interesting. Through observing many transient relationships we can learn the presence of a particular type of permanent relationship. As illustrated in the example of the two university colleagues, it is the permanent relationship that dictates the type of transient relationship the two individuals will most often find themselves working under. More specifically, due to the presence of a particular type of permanent relationship between two agents A and B, they are most likely to interact under a transient relationship of the same type. In addition, when A and B interact it is only possible for them to spare one particular type of transient relationship; in a given context and time, A and B cannot be both competing and collaborating. This is true of human societies, from which we draw inspiration. Given a context, as humans, we do not compete with *and* cooperate with an individual. In fact, in legal contexts where this is likely to occur, the individuals simply back away from the interaction, describing it as a conflict of interest. More formally, the introduction of transient relationships imposes two strong assumption on our work.

Assumption 1 *True Correlation* — The true permanent relationship between two agents is likely to give rise to transient relationships of the same type in future interaction episodes.

Assumption 2 *Exclusivity* — Only one transient relationship can hold between two agents in a single interaction episode.

The analysis presented so far has shown how we conceptualise relationships. We have presented a simple classification of relationships that can exist between agents in a multi-agent system, and more specifically, we have introduced the notion of permanent and transient relationships. An important limitation of the analysis and its results are that they are not based on a real system, and therefore it is necessary for us to instantiate the relationships discussed in the this section in a scenario.

5.3 A Scenario for Relationships

Here, we present examples of relationships (identified in the previous section) that exist in multi-agent systems. The examples are illustrated through the use of a service provision scenario (much like the one used for the system evaluation of TRAVOS in Section 4.2) in which there are two types of agents to demonstrate how the three different relationships (competitive, co-operative and dependence) may arise. In this scenario there are service provider agents (SPs) and consumer agents (CAs). CAs seek various services, which the SPs offer. The SPs have capabilities to provide a number of services, but their service provision resources are bounded. For example, a particular agent may have the capability to provide only a maximum of 1000 text messages a month. Based on their individual capabilities, SPs can decide whether or not to make a bid for a service provision request, generated by a CA when it requires a certain service or set of services. A single SP can make a bid or, in circumstances where the SP does not have sufficient resources, it can combine its services with other SPs and form a virtual organisation (VO).

Competitive Relationships — To demonstrate a competitive relationship, suppose that a particular CA requires a text messaging service, and there are two service providers, agent A and agent B, in the system, that have the capability of providing text messaging. Both A and B wish to bid for this contract and to be the sole provider of text messaging to the CA. In this situation, A and B are both competing for the contract of being text messaging service provider to the CA, and therefore they are in a competitive relationship. If one of them wins the contract, then the winner will receive the payment for the service from the CA, whereas the other will receive no gain.

Dependence Relationships — To show a dependency relationship we consider a CA that requires a text messaging service that can allow it to send 100 texts per month. In response to this service provision request, a service provider, agent A, decides to make a bid. However, A only has the capability of providing 80 texts a month, which means that for it to achieve its goal it has to obtain a further 20 texts per month. It does this by requesting another service provider, agent B, to provide 20 texts per month (this may be seen as subcontracting). Agent A contracts the additional 20 text messages from B, and now has enough resources to make the bid and win the contract, but it is dependent on B for part of those resources. If B fails to deliver, then A will end up breaking the contract and will fail too. In this example, the CA has a contract with A for 100 texts, and A has a contract with B for 20 texts.

Cooperative Relationships — A cooperative relationship can exist when the need arises for two service providers to pool their resources and build a VO. Here, suppose that a CA wishes to be provided with text messaging and streaming audio services as one package. Now, agent A (a text provider) would like to bid for this contract and become the text service supplier for the CA, but due to the fact that it can only provide one of the services in the package it cannot achieve this goal by itself. On the other hand, agent B (a streaming audio provider) is faced with a similar problem. To achieve their goals A and B must cooperate with each other to pool their resources and create a VO. This VO, with the combined service, will be able to offer a suitable package to the CA. The CA then holds a contract with the VO, and if there is a failure in service provision the CA will hold the VO as a whole responsible.

In this section we have demonstrated, through use of a scenario, how the three types of relationships we identified can arise in agent-based VOs. We now move on to presenting the first mechanism in TRAVOS-R, which describes how an agent in a multi-agent system is able to learn the inter-agent relationships present within the system.

5.4 Learning Inter-agent Relationships

The analysis and scenario presented above has identified and demonstrated the key relationship types that may exist in a multi-agent system. Due to the openness, sheer scale and the number of stakeholders that own agents within such systems it is impossible to give knowledge of all relationships present, at any one time, to an individual agent. However, some may be known, and when such information is available it should be exploited to speed up matters. Given this situation, we describe mechanisms that equip the agents within these systems to learn the relationships that are present, between others in the virtual society, through observation of actions and outcomes of interactions.

This section describes this process of learning the relationships in three parts. Firstly, we describe how an agent can observe certain actions or outcomes, providing support to the validity of the mechanisms described in the subsequent sections. Secondly, we describe how an agent is able to learn the type of transient relationship that is present, between two other agents, in a single interaction episode. Finally, we describe how the knowledge of the transient relationship observed over a number of interaction episodes is used to modify the belief an agent has about the *type* of and the *strength* of the permanent relationship between two interacting agents.

5.4.1 Basic Notation

Before we describe the three main parts of the relationship learning mechanism, we introduce some notation that will be used in the explanation of the different aspects.

Let us consider a society $\mathcal{A}$ of n agents containing agents $a_1, a_2, ..., a_n$. In this society, two agents a_x and a_y are related to each other by a *permanent* relationship of a certain type, denoted

$\mathcal{L}^{a_x,a_y}_{type}$. The *type* of the relationship is drawn from the set of all possible types of relationships, which in our case is $= \{com,\ cop,\ dep\}$.[4] For example, the competitive relationship between agent a_1 and a_2 is represented as $\mathcal{L}^{a_1,a_2}_{com}$.

Over time, distinct pairs of agents $\{a_x, a_y\} \subseteq \mathcal{A}$ may interact with one another in an *interaction episode*. During this episode, agents a_x and a_y may perform a number of actions, produce a number of signals or have a number of communications between each other, all of which are observable by all other agents in $\mathcal{A}$. For simplicity, we do not differentiate between signals, actions or communications; rather we group all the observable results of interactions into a single set $\mathcal{P}$, consisting of observable actions $p_1, p_2, .., p_n$.

Furthermore, as described in Section 5.2.1, during a particular interaction episode, the interacting agents may share a particular type of *transient* relationship. To differentiate between transient and permanent relationships we use the hat notation, representing a transient relationship of a certain type between a_x and a_y as $\hat{\mathcal{L}}^{a_x,a_y}_{type}$.

Having described the basic notation, we now present our relationship learning mechanism in three parts, starting with observing actions produced when agents interact.

5.4.2 Observing Actions

When related agents interact, they are likely to perform a subset of actions from $\mathcal{P}$ more frequently, depending on the type of relationship that they share. Therefore, by observing actions that are produced by the interacting agents we can begin to learn the type of relationships that exist in the system. In this section, we illustrate that certain actions, which are present in a multi-agent system, give clues to the types of relationships present.

In the previous section, for simplicity, we classified a number of different types of observations as actions. However to show that actions and relationships are related, we now consider inter-agent communication. In a multi-agent system it is necessary for agents to communicate during an interaction to achieve a goal. These communication actions can broadly be classified into the following (Singh, 1998): *assertives* which inform agents about information they do not have; *directives* which request information from agents; *commissives* which promise something to agents; *permissives* which give permissions to perform certain actions on information; *prohibitives* which ban some action; *declaratives* which cause events themselves and *expressives* which express emotions and evaluations of a particular agent. Along with communications, in a system there may be certain actions associated with each of the communicative acts. For example, the *act* of providing a security certificate may be associated with permissive communications. We assume that these actions are observable by the agents taking part in the interaction (between whom the communication is taking place), and other agents witnessing the interaction.

[4] In our case the relationships are competitive, cooperative and dependence, as discussed in Section 5.2.

Assumption 3 *Visibility* – Certain actions and outcomes are present and visible to the interacting agents and other third parties observing the interaction.

By observing one of these communicative acts, or more generally a certain action, an agent can reason about the possible transient relationship between the agents the communication was between. For example, consider an agent a_1 sending a permissive communication to agent a_2, which is witnessed by an agent a_3. In this example a_1 has authority and allows a_2 to gain access to something that it would otherwise have no access to. Agent a_3, the observer, can conclude that a_1 and a_2 have cooperated, since a_1 has given a_2 access to some private material.

Having identified that certain communicative acts and actions may occur due to the presence of certain relationships, it is important to note that so far we have only described seven abstract types of communicative acts. It is very difficult to consider which would be present under which relationship. To do this it is necessary to consider the type and content of a particular communicative act category. For example, intuition may lead one to conclude that in a cooperative relationship the majority of communication will be in the form of assertives. However, consider the case in which a_1 asks a_2 for a particular piece of information. In response a_2 sends an assertive communication back to a_1, informing a_1 that it does not wish to supply the information, or that it does not have the requested information. In this case, we can see that a_2 is not cooperating with a_1, illustrating that it is necessary to examine the nature and content of the communicative acts.

5.4.3 Learning a Transient Relationship

Here, we present a method by which an agent (a_3) can learn the *type* of a transient relationship, ($\hat{\mathcal{L}}^{a_1,a_2}$), between two agents ($a_1$ and a_2), through observation of the actions generated from their interaction in a particular interaction episode.

Firstly, we model $\hat{\mathcal{L}}^{a_1,a_2}$ as a random variable that can take on a number of values. These, of course, are limited to the domain of possible relationship types: $\{com, \ cop, \ dep\}$. As previously discussed, an agent can observe actions produced by the interacting parties. Since these actions provide clues to the type of relationship that the interaction is taking place under, the observer (a_3) can use them to calculate the most likely value for $\hat{\mathcal{L}}^{a_1,a_2}$. We use a simple Bayesian process to allow the observing agent to perform such calculations. If an agent is aware of the probability of a certain action being observed, given the fact that a certain type of relationship is present, then after observing the action, by using Bayes rule, it can calculate the posterior probability for that type of relationship.

In more detail, each agent has a conditional probability table (CPT) that specifies the probability of a certain action being observed, given the fact that a certain type of relationship is present. We denote this table $P(\mathcal{P}|\hat{\mathcal{L}}^{a_1,a_2})$, and an example of such a CPT for three observable actions is shown in Table 5.1. This particular example shows that observing action p_1 indicates that there is definitely a relationship of type competitive between two agents in an interaction episode that produced p_1. The CPTs for all actions in $\mathcal{P}$ form parameters of our model (the prior information an agent requires), which have to be carefully set by the system designer.

Relationship	**Actions**		
	p_1	p_2	p_3
$\hat{\mathcal{L}}^{a_1,a_2}_{com}$	1	0.1	0.1
$\hat{\mathcal{L}}^{a_1,a_2}_{cop}$	0	0.6	0.1
$\hat{\mathcal{L}}^{a_1,a_2}_{dep}$	0	0.2	0.1
$\hat{\mathcal{L}}^{a_2,a_1}_{dep}$	0	0.1	0.7

TABLE 5.1: Conditional probability table for actions p_1, p_2 and p_3, $P(p_1, p_2, p_3|\hat{\mathcal{L}}^{a_1,a_2})$.

Having defined the CPT it is feasible, using Bayes's rule, to produce a posterior distribution for the random variable, $P(\hat{\mathcal{L}}^{a_1,a_2})$, given a prior distribution[5] (denoted $P'(\hat{\mathcal{L}}^{a_1,a_2})$) and the set of observed actions during the interaction episode ($p_1, p_2, ..., p_n$), as shown in Equation 5.1. Here, we have assumed independence between the observations of each action, because an action produced by two agents interacting has no direct relation to future and past actions produced by these two interacting agents. Agents are likely to meet in a number of contexts and actions produced when they interact are not a function of the actions that have been produced, but rather they are a function of the context in which they are interacting and the problem they are trying to solve through the interaction.

$$P(\hat{\mathcal{L}}^{a_1,a_2}|p_1, p_2, ..., p_n) \propto P'(\hat{\mathcal{L}}^{a_1,a_2}) \prod_{i=0}^{n} P(p_i|\hat{\mathcal{L}}^{a_1,a_2}) \tag{5.1}$$

The resulting posterior distribution, $P(\hat{\mathcal{L}}^{a_1,a_2})$, is then used to calculate the type of transient relationship that a_3 believes to have existed between the two interacting agents a_1 and a_2. More specifically, it is the most likely value from $P(\hat{\mathcal{L}}^{a_1,a_2})$.

$\hat{\mathcal{L}}^{a_1,a_2}$	Prior $P(\hat{\mathcal{L}}^{a_1,a_2})$	Posterior $P(\hat{\mathcal{L}}^{a_1,a_2}))$
$\hat{\mathcal{L}}^{a_1,a_2}_{com}$	0.25	0.7
$\hat{\mathcal{L}}^{a_1,a_2}_{cop}$	0.25	0.2
$\hat{\mathcal{L}}^{a_1,a_2}_{dep}$	0.25	0.05
$\hat{\mathcal{L}}^{a_2,a_1}_{dep}$	0.25	0.05

TABLE 5.2: Example prior and posterior distribution of the random variable $\hat{\mathcal{L}}^{a_1,a_2}$, as calculated by a_3 after making a number of observations from an interaction episode.

For example consider Table 5.2 which shows an example prior and posterior distribution for a_3's $P(\hat{\mathcal{L}}^{a_1,a_2})$ distribution for the relationship that exists between a_1 and a_2 during a particular interaction episode. From the table, we can see that in the prior distribution, all types of relationship are equally likely, which represents a state where a_3 has not observed any actions.

[5]We assume a uniform prior, so that all types of transient relationships are equally likely before making any observations.

Having observed certain actions, a_3 can determine a posterior distribution, which clearly shows that the most likely type of transient relationship witnessed is that a_1 and a_2 are competing.

In this example, a_3 believes that a_1 and a_2 formed a transient relationship of type competitive – $\hat{\mathcal{L}}^{a_1,a_2}_{com}$. In the next section we explain how this can be used to allow a_3 to learn the type of *permanent* relationship present between a_1 and a_2.

5.4.4 Learning the Permanent Relationship

In Section 5.2.1 we discussed how transient relationships and permanent relationships are related. Here, we describe a mechanism that allows an agent to calculate the *type* and *strength* of a permanent relationship through observation of transient relationships.

We begin by defining a structure that captures an agent's beliefs about the type of permanent relationship between two others. In the earlier discussion, we defined the relationship between transient and permanent relationships, namely that the type of a transient relationship is dictated by the type of permanent relationship present. For example, if two agents are in a permanent competitive relationship, then they are more likely to find themselves interacting in a particular episode under a competitive transient relationship. Therefore, keeping a count of the times when a particular type of transient relationship occurs and when it does not allows an agent to calculate the expectation of it occurring in the next time step. This count must be maintained for all types of transient relationship, and the type that yields the highest expectation can then be deemed to be the type for the permanent relationship that is present.

For reasons outlined in Section 3.3.1, Beta distributions naturally lend themselves to a binary domain and allow effective methods to predict future outcomes based on previous binary outcomes (for example the presence or absence of a particular type of transient relationship). For this reason, the belief about the type of permanent relationship is represented as a set of beta distributions.

More specifically, we define beta distributions that correspond to an agent a_3's belief about each particular type of permanent relationship that can exist between two agents a_1 and a_2. Thus, a_3 has a permanent relationship vector, $\mathbf{R}^{a_1,a_2}$, containing four beta distributions, which represent the four types of relationships between a_1 and a_2, as shown below[6]:

$$\mathbf{R}^{a_1,a_2} =< (\alpha^{a_1,a_2}_{com}, \beta^{a_1,a_2}_{com}), (\alpha^{a_1,a_2}_{cop}, \beta^{a_1,a_2}_{cop}), (\alpha^{a_1,a_2}_{dep}, \beta^{a_1,a_2}_{dep}), (\alpha^{a_2,a_1}_{dep}, \beta^{a_2,a_1}_{dep}) >$$

Now, the parameters of the various beta distributions in the vector correspond to counts of the corresponding types of transient relationships a_3 believes to have existed in previous interaction episodes between a_1 and a_2. In particular, initially the permanent relationship vector is configured with prior values. Then, after observing a particular type of transient relationship, the observer agent (a_3) simply increases the α parameter for the corresponding element, and increases the β parameter for all the other elements.

[6]We choose to represent a beta distribution as a set of its parameters, (α, β).

For example, suppose, that agent a_3 has the following permanent relationship vector[7]:

$\mathbf{R}^{a_1,a_2} =< (1,1),(1,1),(1,1),(1,1) >$.

Table 5.3 shows how this vector changes subject to the transient relationships that agent a_3 calculates over time from observing the interaction episodes between agents a_1 and a_2. The bold numbers in the table show beta distributions where the α parameter is increased in response to observing the corresponding type of transient relationship.

Time	Transient Relationship	$\mathbf{R}^{a_1,a_2}$
0		$< (1,1),(1,1),(1,1),(1,1) >$
1	$\hat{\mathcal{L}}^{a_1,a_2}_{com}$	$<$ **(2,1)**, $(1,2),(1,2),(1,2) >$
2	$\hat{\mathcal{L}}^{a_1,a_2}_{com}$	$<$ **(3,1)**, $(1,3),(1,3),(1,3) >$
3	$\hat{\mathcal{L}}^{a_1,a_2}_{cop}$	$< (3,2)$, **(2,3)**, $(1,4),(1,4) >$

TABLE 5.3: Example of how the permanent relationship vector $\mathbf{R}^{a_1,a_2}$ changes as an agent observes transient relationships.

At any given point in time, the actual type of permanent relationship between agents a_1 and a_2 ($\mathcal{L}^{a_1,a_2}$), as believed by the observer a_3, is defined as follows. It is of the type which has a corresponding beta distribution, in the vector, with the greatest expected value. The expected value of a beta distribution is calculated as shown in Equation 5.2.

$$E = \frac{\alpha}{\alpha + \beta} \tag{5.2}$$

For example, if we refer back to Table 5.3, then we can see that the expected values of the beta distributions at time $t = 3$ are <0.6, 0.4, 0.2, 0.2>. The beta distribution with the highest expected value is found in the element corresponding to the relationship type $\mathcal{L}^{a_1,a_2}_{com}$. This means that at time $t = 3$ agent a_3 believes that the permanent relationship type between agents a_1 and a_2 is competitive.

Storing the knowledge about the type of permanent relationship in vector **R** as a set of beta distributions gives us additional benefits. It provides us with an elegant way of representing the observing agent's belief about the strength of the permanent relationship. Here, we assume that the strength of the relationship, as *perceived* by the observer, is equal to the *confidence* the observer agent has in its belief about the type of the permanent relationship. More formally, we define the confidence metric, δ^{a_1,a_2}, as the integral of the corresponding beta distribution around its expected value (E) by a certain configurable parameter, ε.

Having described how an agent can determine at a given point in time, the type and strength of the relationship between two agents in the multi-agent system, we can now describe how this knowledge can be used in adjusting the opinions provided by agents that are related to a trustee.

[7] Once again, we assume uniform prior values.

5.5 Relationship-Based Heuristics

In this section we discuss how relationships can *impact* the accuracy of opinions provided by opinion providers to trusters. More specifically, we determine exactly how a truster (a_1) is able to adjust opinions, to remove added bias based on the presence of certain relationships between the trustee (a_2) and the opinion provider (a_3).

Assume that when an opinion provider is asked for an opinion by a truster (about a certain trustee) it can respond in a number of ways. For example, it can convey accurate information, it can convey misleading information, or it can not respond at all. The manner in which it responds is dictated by the nature of the relationship it shares with the trustee. Given the relationship analysis (see Section 5.2), the following relationships can exist between an opinion provider and a trustee.

1. The opinion provider and the trustee share a competitive relationship.
2. The opinion provider and the trustee share a cooperative relationship.
3. The opinion provider is dependent on the trustee.
4. The trustee is dependent on the opinion provider.

In Section 3.4 we described a method, using a *probability of accuracy*, to adjust an opinion prior to arriving at a reputation level for an individual. Here, we present a set of relationship-based heuristics that allow an agent to manipulate the reported opinion in a particular way (dependent on the relationship the opinion provider has with the trustee) before combining it with others. The aim is to adjust the opinion so that it does not mislead a truster. We choose heuristics due to the level of subjectiveness in translating the impact of relationships on trust. It our case, the impact of a relationship cannot be represented by a function derived from a mathematical proof.

In more detail, each heuristic adjusts the trust level (in a trustee) portrayed by an opinion provider through its opinion. Previously, we have represented the opinion about a_2 *reported* to a_1, from a_3, as $\hat{\mathcal{R}}_{a_3,a_2}$. More specifically, the opinion was described as a tuple $(\hat{m}_{a_3,a_2}, \hat{n}_{a_3,a_2})$ consisting of reported counts of the number of successful ($\hat{m}_{a_3,a_2}$) and unsuccessful ($\hat{n}_{a_3,a_2}$) interactions the opinion provider has had with the trustee. From these values, using the techniques described in the TRAVOS model (Section 3.4.1), we can create a beta distribution that is representative of the opinion. This distribution has an expected value $E_{\hat{\mathcal{R}}^t_{a_3,a_2}}$, which represents the trust level portrayed by the opinion, and a standard deviation $\sigma_{\hat{\mathcal{R}}^t_{a_3,a_2}}$ that represents the confidence the opinion provider has in its opinion. The heuristics presented in this section adjust the trust level portrayed by an opinion. We denote this adjusted trust level as $\bar{E}_{\hat{\mathcal{R}}^t_{a_3,a_2}}$.

5.5.1 Opinion Provider Competes with Trustee

In the context of this work, as stated in Section 5.2, an opinion provider competes with a trustee if both agents bid for the same contract. In this case, whoever wins the contract gets an increase in utility. For this reason, given the chance, the opinion provider will behave in a way that causes the trustee to lose contracts, thus maximising the chance for itself to win. A strategy the opinion provider can adopt is to provide false negative opinions about the trustee, which involves providing opinions to others that make the trustee seem less trustworthy, regardless of the trustee's true nature. This will lead the truster to calculate a low reputation value for the trustee, and ultimately to the decision that the trustee is too untrustworthy to be awarded the contract. We can assume that the opinion provider is likely to use this strategy due to the negative effect it has on the trustee's reputation.

There is, however, a special case in which the opinion provider may have an incentive to adopt a false positive strategy, in which it provides opinions that make the trustee appear more trustworthy. Consider a situation where the trustee wins a contract and fails to deliver. In this case, the level of trust the truster will place in it next time will be less, making it harder for the trustee to compete in future bids. The opinion provider will find this beneficial because if it was also competing, then it stands a better chance of being awarded the contract against an untrustworthy trustee. To encourage this situation, therefore, the opinion provider may provide false positive information, hoping the trustee will fail. The problem with this strategy is that success requires the trustee to fail, and requires the truster to learn that the opinion provider was not accurate in providing the opinion. Therefore this strategy may enable the opinion provider to compete effectively against the trustee, but will be at the cost of reducing its own reputation as an opinion provider. Given this negative impact on its own reputation the opinion provider is less likely to adopt the false positive strategy.

ALGORITHM 1: How to adjust the trust level portrayed by an opinion provider that competes with the trustee.

$a_1 \leftarrow Truster$
$a_2 \leftarrow Trustee$
$a_3 \leftarrow Opinion\ Provider$
if $\mathcal{L}^{a_3,a_2} = \mathcal{L}^{a_3,a_2}_{com}$ **and** $\delta^{a_3,a_2} \geq \delta_{min}$ **then**
    $E_{\hat{\mathcal{R}}^t_{a_3,a_2}} \leftarrow Trust\ level\ portrayed\ by\ opinion\ \hat{\mathcal{R}}^t_{a_3,a_2}$
    $Amount\ to\ adjust \leftarrow (\delta^{a_3,a_2} \times (E_{\hat{\mathcal{R}}^t_{a_3,a_2}} \times (1 - E_{\hat{\mathcal{R}}^t_{a_3,a_2}})))$
    $\bar{E}_{\hat{\mathcal{R}}^t_{a_3,a_2}} = E_{\hat{\mathcal{R}}^t_{a_3,a_2}} + \ Amount\ to\ adjust$
end if

Given the argument above, if the truster is aware of a competitive relationship ($\mathcal{L}^{a_3,a_2}_{com}$) then it must compensate for the false negative strategy of the opinion provider. That is to say that the truster (a_1) needs to compensate for the incorrect low trust level portrayed in the opinion ($\hat{\mathcal{R}}_{a_3,a_2}$) reported by the opinion provider (a_3). It can do this by increasing the portrayed trust level (increasing $E_{\hat{\mathcal{R}}^t_{a_3,a_2}}$) by a certain factor. We believe that the confidence, δ^{a_3,a_2}, the truster has in its belief about the relationship should be the compelling factor in deciding the magnitude of the adjustment, and if it is below a certain amount δ_{min}, no adjustment should be made. To this end, we present Algorithm 1, which allows a truster to cope with opinions provided by agents that are competing with the trustee. More specifically, it shows how the portrayed level of trust is increased, based on a function of the agent's confidence that a competitive relationship is present, and the actual portrayed level of trust from the opinion. Since the assumption for this algorithm is that the opinion will be biased towards a low trust value, it is better to measure the amount we wish to adjust the opinion by considering the difference in magnitude of the trust level portrayed by the opinion and a trust value of 1 (this is shown in on Line 6 as $(1 - E_{\hat{\mathcal{R}}^t_{a_3,a_2}})$).

5.5.2 Opinion Provider Cooperates with Trustee

A cooperative relationship between two agents means that the two agents are sharing resources towards and profit from achieving a goal. Due to the mutual gain between the agents, they may find it beneficial to promote each other's trustworthiness. So, given the presence of this relationship between the opinion provider and the trustee, the opinion provider is likely to adopt a false positive strategy to make the trustee appear more trustworthy, increasing the trustee's chances of winning contracts. The presence of the cooperation means that it is likely that the trustee will pool its resources with the opinion provider to meet this increase in demand, thus increasing the opinion provider's gain.

A cooperative relationship may mean that the opinion provider and the trustee belong to the same VO, providing an additional incentive for the opinion provider to propagate false positive information about the trustee. In so doing, the opinion provider will increase the overall reputation of the VO, thus increasing the VO's ability to win contracts and maximising its own individual gain.

On the other hand, the opinion provider may pursue a false negative strategy, making the trustee appear less trustworthy. The reason for this may be that the opinion provider wishes the trustee to free up some resources, which can be utilised by the opinion provider in future cooperative work. Propagating false negative information about the trustee will result in the trustee having greater resource availability, but it will also damage the trustee's reputation and the reputation of any VO that the trustee may form with the opinion provider in the future.

Finally, there is one last strategy the opinion provider may adopt, the honest strategy. Here, the opinion provider provides accurate (to the best of its knowledge) opinions that reflect the true beliefs that it has about the trustee's behaviour. This is the most likely strategy when the opinion provider has a high level of trust in the trustee, and believes that the trustee can fulfil the requirements of the truster. Giving an honest opinion about a trustee, that the opinion provider knows to be very trustworthy, will not only increase the chances of the trustee winning contracts but will also mean that the opinion provider does not harm its reputation for providing opinions. However the honest opinion strategy is likely to cause the trustee not to win contracts if the opinion provider does not have much trust in the trustee. In this case, it is necessary for the opinion provider to provide an opinion that portrays a high trust level for the trustee (and as we have said before, this is achieved using a false positive strategy).

Given the arguments above, in the presence of a cooperative relationship, the truster needs to accommodate for the false positive strategy that an opinion provider is likely to adopt. A truster is able to do this by decreasing the trust level portrayed by an opinion. Algorithm 2 shows how this can be achieved.[8] More specifically, the algorithm shows how the portrayed level of trust is reduced by an amount. This amount is a function of the actual portrayed trust level, and the confidence the agent has in the fact that a cooperative relationship is present between the trustee and the opinion provider. In this algorithm, since we are assuming that the opinion will be biased towards a high trust value, it is better to measure the amount we wish to adjust the opinion by considering the difference in magnitude of the trust level portrayed by the opinion and a trust value of 0 (this is why Line 6 of this algorithm differs from that of Algorithm 1).

ALGORITHM 2: How to adjust the trust level portrayed by an opinion provider that cooperates with the trustee.

1: $a_1 \leftarrow Truster$
2: $a_2 \leftarrow Trustee$
3: $a_3 \leftarrow Opinion\ Provider$
4: **if** $\mathcal{L}^{a_3,a_2} = \mathcal{L}^{a_3,a_2}_{cop}$ **and** $\delta^{a_3,a_2} \geq \delta_{min}$ **then**
5: $\quad E_{\hat{\mathcal{R}}^t_{a_3,a_2}} \leftarrow Trust\ level\ portrayed\ by\ opinion\ \hat{\mathcal{R}}^t_{a_3,a_2}$
6: $\quad Amount\ to\ adjust \leftarrow (\delta^{a_3,a_2} \times (E_{\hat{\mathcal{R}}^t_{a_3,a_2}} \times E_{\hat{\mathcal{R}}^t_{a_3,a_2}}))$
7: $\quad \bar{E}_{\hat{\mathcal{R}}^t_{a_3,a_2}} = E_{\hat{\mathcal{R}}^t_{a_3,a_2}} - \ Amount\ to\ adjust$
8: **end if**

5.5.3 Opinion Provider Depends on Trustee

The presence of a dependency relationship means that one agent requires another's abilities to achieve its individual goal. When the opinion provider is dependent on the trustee, the opinion provider requires some of the trustee's services, and is likely to pay slightly more due to the trustee's stronger negotiation position (see Section 5.2). An opinion provider that finds itself in such a situation may be indifferent to the choice of providing false opinions that make the trustee seem more or less trustworthy. By increasing the perceived reputation of the trustee, the opinion provider does not stand to gain an increase in utility through contracts won or lost by the trustee (as in Sections 5.5.1 and 5.5.2). In this case, the strategies that the opinion provider is likely to adopt are likely to be related to the trustee's resource utilisation.

It is in the interest of the opinion provider to drive down the market demand for the trustee's resources, which is advantageous in two ways. Firstly, a low demand would mean that the trustee would not be utilising its capabilities, allowing the dependent opinion provider to use

[8] We have used the notation introduced in the previous section.

them when the need arises. Secondly, the low demand might mean that the trustee will not charge so much for its resources, thereby allowing the opinion provider to use its services at a discounted rate (and reducing the negotiation position of the trustee). The opinion provider can achieve this by using a false negative strategy, which reduces the perceived reputation of the trustee, creating a low market demand for its resources.

If the truster is aware of the fact that the opinion provider depends on the trustee, then it needs to accommodate for the false negative strategy that an opinion provider, in such a relationship, is likely to adopt. A truster is able to do this by increasing the trust level portrayed by the opinion provider's opinion. Furthermore, in this case, the heuristic should take into consideration that an opinion provider that reports an opinion portraying a high level of trust[9] is not likely to have exaggerated its opinion as much as one that provides a very low level[10] of trust. Therefore, we introduce a factor ($\eta \in [0, 1]$) which allows us to adjust high trust and low trust opinions separately. To this end, we present Algorithm 3, which shows in detail how an agent can adjust an opinion given the knowledge that the opinion provider depends on the trustee. More specifically, Algorithm 3 adjusts the opinion in two ways depending on wether the portrayed trust level is below a neutral level. If the portrayed level of trust is below a neutral level, then it increases the portrayed trust level using a function of η, the actual portrayed trust level and the confidence the agent has in the presence of a dependence relationship. If the portrayed level of trust is above a neutral level then the increase is a function of the actual portrayed trust level and the confidence only.

ALGORITHM 3: How to adjust the trust level portrayed by an opinion provider that depends on the trustee.

$a_1 \leftarrow Truster$
$a_2 \leftarrow Trustee$
$a_3 \leftarrow Opinion\ Provider$
if $\mathcal{L}^{a_3,a_2} = \mathcal{L}^{a_3,a_2}_{dep}$ **and** $\delta^{a_3,a_2} \geq \delta_{min}$ **then**
    $E_{\hat{\mathcal{R}}^t_{a_3,a_2}} \leftarrow Trust\ level\ portrayed\ by\ opinion\ \hat{\mathcal{R}}^t_{a_3,a_2}$
    $Amount\ to\ adjust \leftarrow (\delta^{a_3,a_2} \times (E_{\hat{\mathcal{R}}^t_{a_3,a_2}} \times (1 - E_{\hat{\mathcal{R}}^t_{a_3,a_2}})))$
    if $E_{\hat{\mathcal{R}}^t_{a_3,a_2}} \leq 0.5$ **then**
        $\bar{E}_{\hat{\mathcal{R}}^t_{a_3,a_2}} = E_{\hat{\mathcal{R}}^t_{a_3,a_2}} + (\eta\ \times\ \ Amount\ to\ adjust)$
    else
        $\bar{E}_{\hat{\mathcal{R}}^t_{a_3,a_2}} = E_{\hat{\mathcal{R}}^t_{a_3,a_2}} + Amount\ to\ adjust$
    end if
end if

[9] A trust level above that of a neutral trust level, which in TRAVOS is 0.5.
[10] A trust level below that of a neutral trust level, which in TRAVOS is 0.5.

5.5.4 Trustee Depends on Opinion Provider

If the trustee is dependent on the opinion provider, then the opinion provider stands to profit if there is an increase in market demand for the trustee. An increase in demand would mean that the trustee has more contracts to fulfil, and the probability of it depending on the opinion provider to successfully fulfil those contracts increases. As discussed in the previous section, the opinion provider can try and influence the trustee's market demand by manipulating the trustee's reputation. By falsely increasing the trustee's reputation the opinion provider can increase the potential for the trustee to obtain more contracts. This false positive strategy, however, involves some degree of risk on the opinion provider's part. Since the opinion provider is falsely increasing an untrustworthy trustee's reputation, there is a strong possibility it will fail in fulfilling the many contracts it has been awarded. Such failures will reflect poorly on the ability of the opinion provider to provide true opinions. This associated risk will be an important factor the opinion provider will consider when judging to what extent it exaggerates the opinion before delivering it to the truster.

The opinion provider is unlikely to adopt a false negative strategy, which would prevent the trustee from winning contracts, reducing the amount of work it gives to the opinion provider. One may argue that the opinion provider will then be able to pursue its own goals. However, considering that an opinion provider is rational, it would rather be depended upon as this gives it a stronger negotiation position, allowing it to drive the price it charges for its resources above the normal market price.

Given the above arguments, and knowing that this relationship is the opposite of the opinion provider depending on the trustee, we present Algorithm 4. This heuristic allows a truster to decrease the trust level portrayed by the opinion provider's opinion.

ALGORITHM 4: How to adjust the trust level portrayed by an opinion provider that is depended upon by trustee.

$a_1 \leftarrow Truster$
$a_2 \leftarrow Trustee$
$a_3 \leftarrow Opinion\ Provider$
if $\mathcal{L}^{a_2,a_3} = \mathcal{L}^{a_2,a_3}_{dep}$ **and** $\delta^{a_3,a_2} \geq \delta_{min}$ **then**
  $E_{\hat{\mathcal{R}}^t_{a_3,a_2}} \leftarrow Trust\ level\ portrayed\ by\ opinion\ \hat{\mathcal{R}}^t_{a_3,a_2}$
  $Amount\ to\ adjust \leftarrow (\delta^{a_3,a_2} \times (E_{\hat{\mathcal{R}}^t_{a_3,a_2}} \times E_{\hat{\mathcal{R}}^t_{a_3,a_2}}))$
  if $E_{\hat{\mathcal{R}}^t_{a_3,a_2}} \leq 0.5$ **then**
    $\bar{E}_{\hat{\mathcal{R}}^t_{a_3,a_2}} = E_{\hat{\mathcal{R}}^t_{a_3,a_2}} - \ Amount\ to\ adjust$
  else
    $\bar{E}_{\hat{\mathcal{R}}^t_{a_3,a_2}} = E_{\hat{\mathcal{R}}^t_{a_3,a_2}} - (\eta\ \times\ \ Amount\ to\ adjust)$
  end if
end if

5.6 Summary

In Chapter 3 we presented a novel computational model of trust, that met the majority of our research aims specified in Section 1.3. More specifically, it enabled an agent to account for uncertainty in the actions of its interaction partner, by allowing the agent to determine a trust level for the partner. However, the model presented fell short of addressing one key aim of this research, namely to incorporate social information into the trust calculations in a computational trust model. Agents in VO environments are not required to have relationships, but over time, and through regular interactions, certain relationships between such agents may emerge. The result of such relationships is that they may incentivise agents to behave in a particular manner, more specifically, to provide exaggerate opinions about related agents. To this end, in this chapter we presented TRAVOS-R, an extension to TRAVOS that incorporates social information into trust calculations. By doing so, we extend the state of the art in computational trust models in the following ways.

- Certain types of relationships may exist between agents in an agent-based VO. We have (in Section 5.2) provided a general taxonomy of such relationship types.
- Where other models assume that a sociogram (or similar data structures) are available at the disposal of agents, to determine the relationships between others in the virtual community, we present a novel method of equipping an individual agent with the ability to learn the presence of such relationships (Section 5.4). This means that an agents no longer has to have access to a centralised store of relationship information, and that a third party does not have to keep updating this store of relationship information as each agent can maintain their own relationship information. This is a clear advantage in open systems.
- Finally, based on our taxonomy of relationships, we present a set of relationship-based heuristics that can be used within trust calculations to prevent a truster being misled by biased opinions provided by self-interested opinion providers who are related to the trustee in a particular way.

In this chapter we have shown that an agent can obtain and use social information in trust calculations, and we believe that this allows an agent to determine a more accurate trust level, and give an agent protection from being misled by others in the system. In the next chapter we show through empirical and system evaluation that this is indeed the case, and that our approach enhances the ability of an agent to calculate a trust level for another.

Chapter 6

Evaluation of TRAVOS-R

We achieved the aim of incorporating social information into trust calculations in the previous chapter, but we still need to consider exactly how this social information affects the trust model's performance. In this chapter we present the empirical evaluation of the mechanisms that make up TRAVOS-R. The primary focus of the evaluation is to identify how the TRAVOS-R approach performs compared with TRAVOS, and draw conclusions about the nature of its performance under different configurations and in different environments. To do this, we evaluate the approaches in two distinct ways. Firstly, we explore the performance of TRAVOS-R under different configurations of the prior information that is required to bootstrap the model (in Section 6.2). Secondly, we explore the performance of TRAVOS-R in a variety of different environments, and compare it directly to the performance of TRAVOS (in Section 6.3). Furthermore, as per Section 4.2 we provide a system evaluation that shows how the TRAVOS-R mechanisms, described in the previous chapter, can be used in a system to ensure that agents are not misled by biased opinions (in Section 6.4).

6.1 Empirical Evaluation Methodology

The TRAVOS-R model is evaluated using a simulation testbed consisting of interactions between consumer agents and service provider agents. It is similar to that described for the TRAVOS evaluation in Chapter 4, but there are key differences in the roles of the consumer and the service provider agents. These differences arise from the need to simulate an environment which supports the added functionality of TRAVOS-R (namely the identification of relationships between service and opinion providers). In this section we detail the exact settings of the simulation testbed and its components. Firstly, we describe the agents that are found within the simulation, and then, we go on to describe how we simulate the various relationships between them. We conclude this section with an outline of the interactions that are simulated between the agents present, and a description of the measurements that can be made from running the simulation.

6.1.1 Consumer and Service Provider Agents

The simulation consists of two main types of agents, which represent the broad types that may be found in real life systems. Firstly, there are consumer agents representing agents that consume services and need to evaluate the trustworthiness of the various service-providing agents. These service-providing agents make up the second type, representing those agents that have the capabilities to provide certain services.

In this simulation, the consumer agents are able to calculate the trustworthiness of a given service provider using three different models of trust, resulting in three types of consumer agents, as described below:

1. *Zero Intelligence Consumer Agent* — The Zero Intelligence trust model is a simple one; it evaluates the trustworthiness of any agent to be neutral (0.5), regardless of any other information obtained by the consumer agent. [1]
2. *TRAVOS Consumer Agent* — This agent uses the TRAVOS model, as described in Chapter 3, to evaluate the trustworthiness of the service providers.
3. *TRAVOS-R Consumer Agent* — This agent uses the TRAVOS-R mechanisms (described in Chapter 5) to evaluate the trustworthiness of service providers.

Here, we must note that in each case (for TRAVOS-R and TRAVOS consumer agent types) the trustworthiness is calculated purely from opinions provided by others, and does not include the direct interaction component. The reason for this is simply that the difference between TRAVOS-R and TRAVOS is the manner in which they adjust, filter and combine opinions.

Within this simulation, service providers are expected to behave in a certain way with regard to their service-providing capabilities. Specifically, the behaviour of the service provider agents is characterised by a variable $B \in [0, 1]$, which dictates the nature of the outcome from an interaction with them. Thus, B represents the probability that the service provider will provide the service as specified (as described in Section 3.2). For example, an interaction with a service provider with $B = 1$ will always result in a positive or successful outcome. On the other hand, only 50% of interactions with a trustee with $B = 0.5$ will result in a positive outcome. We set the population of service providers in the simulation to be uniformly distributed across all values of B.

Unlike the simulation environment described in Chapter 4, in which service provider agents and opinion provider agents were distinct, with each addressing only one functionality (providing services or opinions), we now consider the combination of functionalities so that a service provider delivers both services and opinions.[2] Given this, there are now two types of service providers, created through differences in the way they provide opinions:

[1]The name Zero Intelligence name is inspired by Gode and Sunder (1993), and conveys the fact that this agent does not use any knowledge or intelligence in calculating a level of trustworthiness for others.

[2]For the remainder of this chapter we use the term service provider and opinion provider interchangeably.

1. *Accurate Service Provider (ASP)* — Service providers of this type provide their opinions based on their direct observations. They do not exaggerate or distort the events that they have witnessed.

2. *Biased Service Provider (BSP)* — The service providers belonging to this category provide a biased opinion based on the relationship they share with the service provider in question (see the next section for more details on service providers and relationships). This bias is either positive or negative noise added to the actual opinion, again based on the relationship shared. For example, if a service provider A is requested to provide an opinion to a service provider B about another service provider C, and A shares a competitive relationship with C, then A adds negative noise to the opinion. This makes C appear less trustworthy than A believes it is (as discussed in Section 5.5.1).

Having outlined the different agents in the simulation, we now move to describe the functionality of the simulation testbed that emulates relationships between these agents.

6.1.2 Relationships Within the Service Provider Agent Population

The largest difference between the simulation environment presented here and that in which TRAVOS was evaluated (Chapter 4) is the presence of relationships between agents. More specifically, the service provider population is configured so that there are relationships between all the service providers, resulting in each service provider having the option to be incentivised to provide biased opinions. Specifically, the relationship between service provider A and service provider B is chosen uniformly from a distribution consisting of four relationship types (discussed in more detail in Section 5.3):

1. A competes with B.

2. A cooperates with B.

3. A depends on B.

4. A is depended upon by B.

This means that each member of the service provider population has a relationship with every other member of the population. As discussed in Section 5.4.2, if agents have a particular relationship they are likely to behave in a certain way towards each other, resulting in actions and signals that a third party agent may observe. In this simulated environment, as a result of a shared relationship, two agents generate such signals. We create a set of 12 signals, so that any one relationship type can cause one of three signals to be produced by two interacting service providers. More specifically, within the simulation, the signals are determined by the distribution shown in Figure 6.1, which shows how the probability of a signal being produced varies based

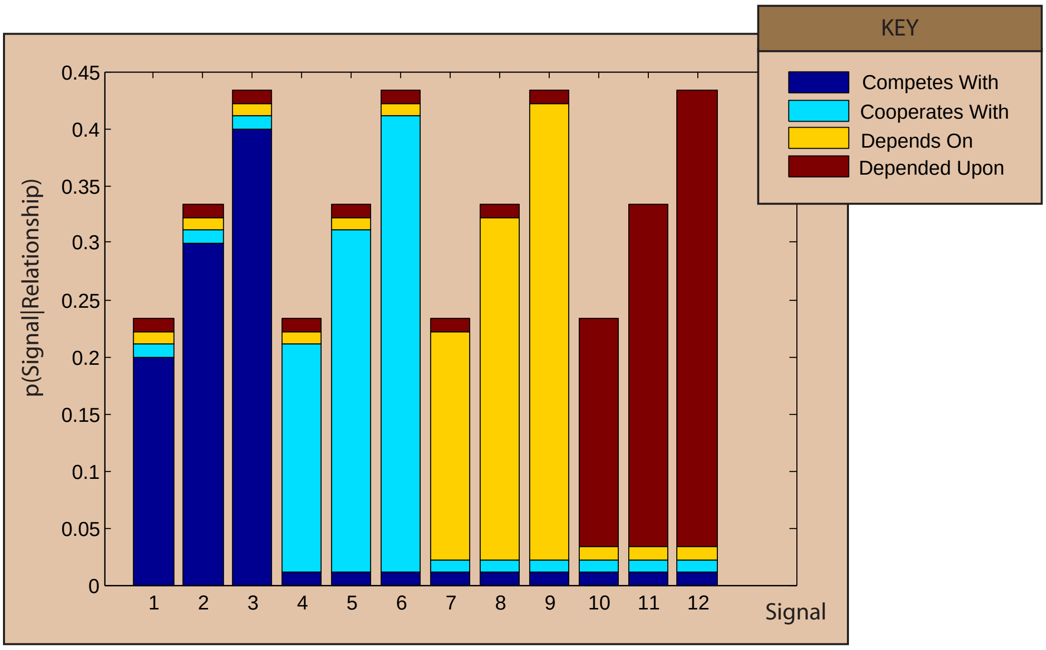

FIGURE 6.1: The probability distribution used by the simulation to generate one of twelve signals based on the relationship present between two interacting agents.

on the relationship type. This distribution is selected as it enables certain distinct subsets of signals to be attributed to specific relationships, and adds a small element of randomness to the generation of signals from interacting service providers.[3] For example, if two agents share a competitive relationship, then an interaction between them is most likely (with a probability of 0.9) to produce Signal 1, 2 or 3, but there is a small probability (0.1) that it can produce any of the other signals from Signal 4 to 12 inclusive.

Having described the agents found in the simulation, the relationships between them, and how the relationships influence certain types of opinion provision, we now outline the simulated interactions between the agents.

6.1.3 Simulating Agent Interactions

The simulation begins by configuring a consumer agent (or multiple consumer agents) and a population of service providers. Then the simulation assigns relationships between agents in the service provider population. Following this, the simulation begins to emulate a predetermined number of interactions that can occur between the mix of agents. These interactions fall into one of the following categories:

- *Experience Interaction* — A experience interaction is one where agents within the service

[3]The randomness makes the simulation more like the real world, where agents may interact under a particular relationship, but produce signals that might be produced by the presence of another relationship.

provider population interact amongst themselves to generate experience of each other. Essentially, this allows them to form opinions about each other, which they use to service requests for opinion provision made by the consumer.

- *Opinion Experience Interaction*[4] — An opinion experience interaction is one where a consumer agent asks a service provider agent for an opinion about another service provider. The opinion and the outcome of the interaction form experience that the consumer agent gains, and it is used to allow the consumer agent to form a better evaluation of the opinion-providing capabilities of the agents providing opinions.
- *Signal Experience Interaction*[5] — A signal experience interaction occurs when a consumer agent observes a signal generated from an interaction between two service providers. The signal depends on the relationship shared by the two service providers, and is generated from a distribution (as shown in Figure 6.1, and described above in Section 6.1.2).

The simulation concludes with the consumer agent evaluating the trustworthiness of each individual service provider, using opinions provided by all the other service provider agents in the population. Similarly to the approach described at the end of Section 4.1.1, in each simulation run we measure the *mean estimation error* for each consumer agent. Where multiple runs are used for particular experiments, an average of the error over all the runs is taken as a measurement for the experiment.

Against the above background, the evaluation of TRAVOS-R is carried out as a series of experiments. Each experiment consists of 50 simulation runs, which allows the results to be tested for statistical significance. More specifically, for each experiment, we test whether the difference between the mean error values obtained from the independent samples is statistically significant, using a t-test at the 95% confidence level. Finally, to ensure independence between simulation runs, any history or experience the consumer agents and the service provider agents build up over an episode is cleared at the end of that episode.

6.2 Evaluating Different TRAVOS-R Bootstrap Configurations

The TRAVOS-R model needs to be bootstrapped with a certain amount of relationship information before it can be used in a system. This prior information allows the agents to effectively learn the relationships that are present, and thereby choose the correct relationship-based heuristic to apply to the opinions they encounter. Here, we describe the evaluation of the performance of TRAVOS-R consumer agents under different forms of prior information.

[4] This form of interaction provides experience that is utilised by the TRAVOS mechanisms.

[5] This form of interaction provides experience that is utilised by the TRAVOS-R mechanisms.

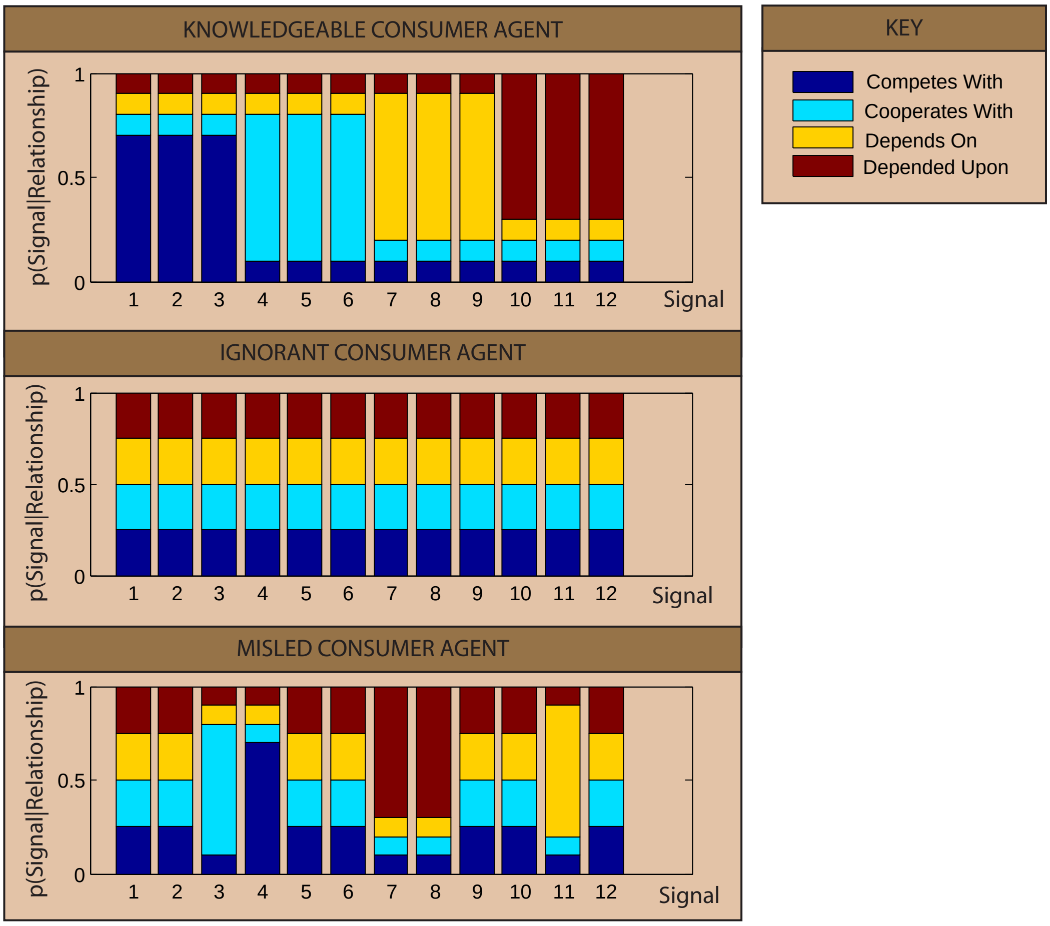

FIGURE 6.2: Relationship given signal distributions used in three different configurations of TRAVOS-R.

6.2.1 Types of TRAVOS-R Agents

In detail, the prior information about relationships that a TRAVOS-R consumer agent has is held as conditional probability tables (CPTs, see Section 5.4.2), which represent the probability of observing particular signals given the presence of particular relationships. We create types of the TRAVOS-R consumer agent that have different CPTs, representing different forms of prior information. The three subtypes are described below, and the their associated prior distributions are shown in Figure 6.2:

1. *Knowledgeable Consumer Agent (KCA)* — This agent is configured with an accurate distribution, which is very similar to the actual distribution used to generate the signals in the simulation. This type of configuration is representative of a system where the designer is aware of the possible actions and signals that can be produced by the presence of a particular relationship.

2. *Ignorant Consumer Agent (ICA)* — This agent is configured with a uniform distribution,

so that it is unable to make any sense of the signals that it observes. The ICA is representative of systems where the designer does not know the connection between the presence of a particular relationship type and the signals and actions produced by interacting agents.

3. *Misled Consumer Agent (MCA)* — This agent is configured with an inaccurate distribution. More specifically, it is configured with CPTs that either give it completely incorrect information about what relationship a particular signal represents, or uniform prior information (indicating that it has no knowledge). This agent represents the scenario where the system designer may, mistakenly, assign incorrect CPTs due to lack of (or incorrect) knowledge about the relationships and signals in a system.

6.2.2 Experimental Process

The three types of TRAVOS-R consumer agents, as described above, are tested in two environments: (i) where the entire service provider population consists of ASPs, and (ii) where the entire population is made up of BSPs. These configurations are selected as they represent environments in which the TRAVOS-R mechanism is required and one in which it is not required. In each case, we vary the number of *experience interactions*, $n_{ei} \in \{0, 5, 10, 15, 20\}$, and for each setting of experience interaction we vary the number of *signal experience interactions*, $n_{si} \in \{0, 5, 10, 15, 20\}$.

In this set of experiments the ACA is expected to be able to identify the correct relationships that are present, and should therefore be able to apply the correct relationship-based heuristics. Ultimately, this should lead it to *outperform* the other two. By outperform we mean that it produces a lower mean estimation error, meaning that its estimates for the trustworthiness of others are more accurate. More formally, we state the aim of this experiment in a hypothesis as follows.

> **Hypothesis 1**
> When varying the prior information about relationships that a TRAVOS-R agent has, the KCA will outperform both the MCA and the ICA. The MCA's performance will degrade as the number of signals increase, and it will be poorer than the ICA.

The results clearly show that more informative prior information, such as the knowledgeable distribution, enables a consumer agent to perform better (see Figure 6.3). In both environments (100% BSP and 100% ASP populations) the KCA is able to outperform the other two, a result that validates Hypothesis 1. In the best case it is able to outperform the ICA by a mean error of 0.2, and the MCA by a mean error of 0.05. The results also show a notable change in the performance of the models. As can be seen in Figure 6.3, the models produce a larger mean error in environments where there is a 100% BSP population, because each agent adds a lot of biased noise to the opinions before supplying them to the consumer. This causes the consumer to be misled.

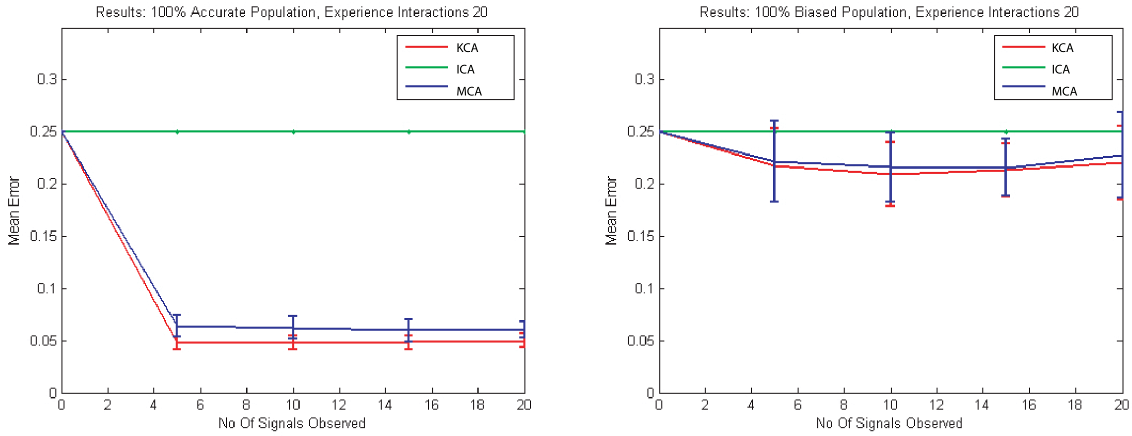

FIGURE 6.3: Plots showing how different configurations of TRAVO-R performed in environments with 100% ASP population (on the left) and a 100% BSP population (on the right).

However, contrary to our expectations, even though the performance of the MCA is worse than the KCA, the performance of each does not differ significantly (as can clearly be seen in the full set of results shown in Appendix A). Although not an obvious result, the MCA is able to outperform the ICA (by a mean error of 0.15). It also closely follows the performance of the KCA because much of the *signal given relationship* distribution of the MCA (see Figure 6.2) is uniform, which results in a reduction of confidence in the relationships it perceives are present.

More specifically, the aspects of the distribution that are not uniform (for example Signal 3) cause TRAVOS-R to use incorrect relationship-based heuristics. However, signals such as Signal 1 in the MCA's prior information have a uniform CPT, leading the MCA to believe that every relationship is equally likely, and hence it lowers its confidence in the relationship type it believes is present. Such a low confidence implies a minor impact on the adjustment the relationship-based heuristic makes on the opinion. Therefore, after the adjustment, the opinion contains much more of its original information. In environments with 100% ASPs this clearly presents a benefit, which ultimately results in the MCA's performance somewhat following that of the KCA with little variance. However, in 100% BSP populations, this leads to little adjustments of opinions that contain a large degree of bias and so the variance in the performance increases (as can be seen in Figure 6.3).

In general, the results suggest that the end performance of an agent using TRAVOS-R is affected by the configuration of the TRAVOS-R model by the system designer. It is important that the system designer is able to accurately represent the probability of observing a particular signal given a particular relationship. In cases where this is not possible, it is better for the agent designer to use prior information that is composed mainly of uniform parts, rather than completely uniform prior information.

Having examined how different configurations of TRAVOS-R perform, in the next section we examine how its performance changes in a variety of environments, and how it compares with the TRAVOS mechanism.

6.3 Evaluating TRAVOS-R in Different Environments

The evaluation of the TRAVOS-R model in a variety of environments is critical to ascertain in which cases it would be appropriate to use it. We choose to evaluate the model against TRAVOS, as the mechanisms in TRAVOS-R aim to be a direct improvement on those contained in TRAVOS and TRAVOS has already been benchmarked against comparable state of the art models in Chapter 4.

6.3.1 Creating Different Environments

To comprehensively evaluate TRAVOS-R we create a variety of environments, ranging from those with limited or no trust information, to those rich in trust information. *Trust information* in this context means the presence of information in the system that can allow an agent to make an accurate judgement about the trustworthiness of another. For example, consider a system in which no agents have interacted, so that there is no information in the system for these agents to accurately form opinions or evaluate trustworthiness of other agents. On the other hand, an environment in which agents have been interacting with, forming opinions about, and obtaining opinions from others for some time, is rich in trust information. Here, agents have a significant amount of information (experience) that can be used to accurately evaluate the trustworthiness of others. More specifically, we create a variety of environments by varying four parameters of the simulation testbed; the values of the parameters are chosen such that they create a spectrum of environments from low trust information to high trust information environments. In addition, we vary the parameters in a modular way (i.e. from 0 to 5, as opposed to from 0 to 5 inclusively) so that we can determine the general effect of changing the variable on the model without running many unnecessary experiments. The following summarises the parameters and their domains:

- Service provider population composition (see Table 6.1).
- Number of experience interactions, $n_{ei} \in \{0, 5, 10, 15, 20\}$.
- Number of opinion experience interactions, $n_{oi} \in \{0, 5, 10, 15, 20\}$.
- Number of signal experience interactions, $n_{si} \in \{0, 5, 10, 15, 20\}$.

Given the above, in each possible environment we test the performance of the three types of consumer agents: a TRAVOS consumer, a TRAVOS-R consumer (configured with the knowledgeable distribution shown in Figure 6.2) and a Zero Intelligence consumer. The Zero Intelligence consumer acts as the the control for the experiments, and produces a baseline performance against which the other two models can be compared. Here, we present a subset of the results

Service Provider Population	
% Accurate	% Biased
100	0
75	25
50	50
25	75
0	100

TABLE 6.1: Different service provider population configurations.

that represent the general trends found in all the results. The entire set of results are shown in Appendix B, and confirm the findings stated in the rest of this subsection. We now present a series of hypotheses alongside their discussion in light of the results.

6.3.2 Changing the Service Provider Population Composition

It is critical to evaluate the performance of the model in environments where the population, from which it is obtaining opinions, is changed with respect to the number of ASPs and BSPs it contains. It is feasible to assume that in a real life application it is unlikely that the entire population of the agent system will consist completely of agents that are either accurate or biased in their opinion provision. For this reason, evaluating the mechanisms in a range of preconfigured populations establishes when it is beneficial to use TRAVOS-R instead of TRAVOS. TRAVOS requires a number of opinion provision instances to learn which agents are providing good opinions and which agents are providing bad opinions. On the other hand, TRAVOS-R does not rely on past opinion provision episodes and is able to adjust opinions based on heuristics. For this reason, when the number of opinion experience interactions is low, TRAVOS is likely to be outperformed by TRAVOS-R, and *viceversa* when the number of opinion experience interactions observed by the TRAVOS agent increases. More formally, we state the hypothesis for this part of the evaluation as follows.

> **Hypothesis 2**
> Increasing the percentage of BSPs in the population will result in TRAVOS-R outperforming both TRAVOS and the Zero Intelligence consumer agents. For a given number of *experience interactions*, n_{ei}, and a given number of *opinion experience interactions*, n_{oi}, as the percentage of ASPs increases, TRAVOS will outperform the other two.

Results have been obtained to test this hypothesis by running experiments where the service provider population composition is varied from 100% ASPs to 0% ASPs in decrements of 25%, as shown in Table 6.1. The results clearly show one trend, which is that as the percentage of BSPs increases, the performance of both TRAVOS and TRAVOS-R consumer agents falls. This is largely due to the increase in the number of false opinions that the consumer agents receive.

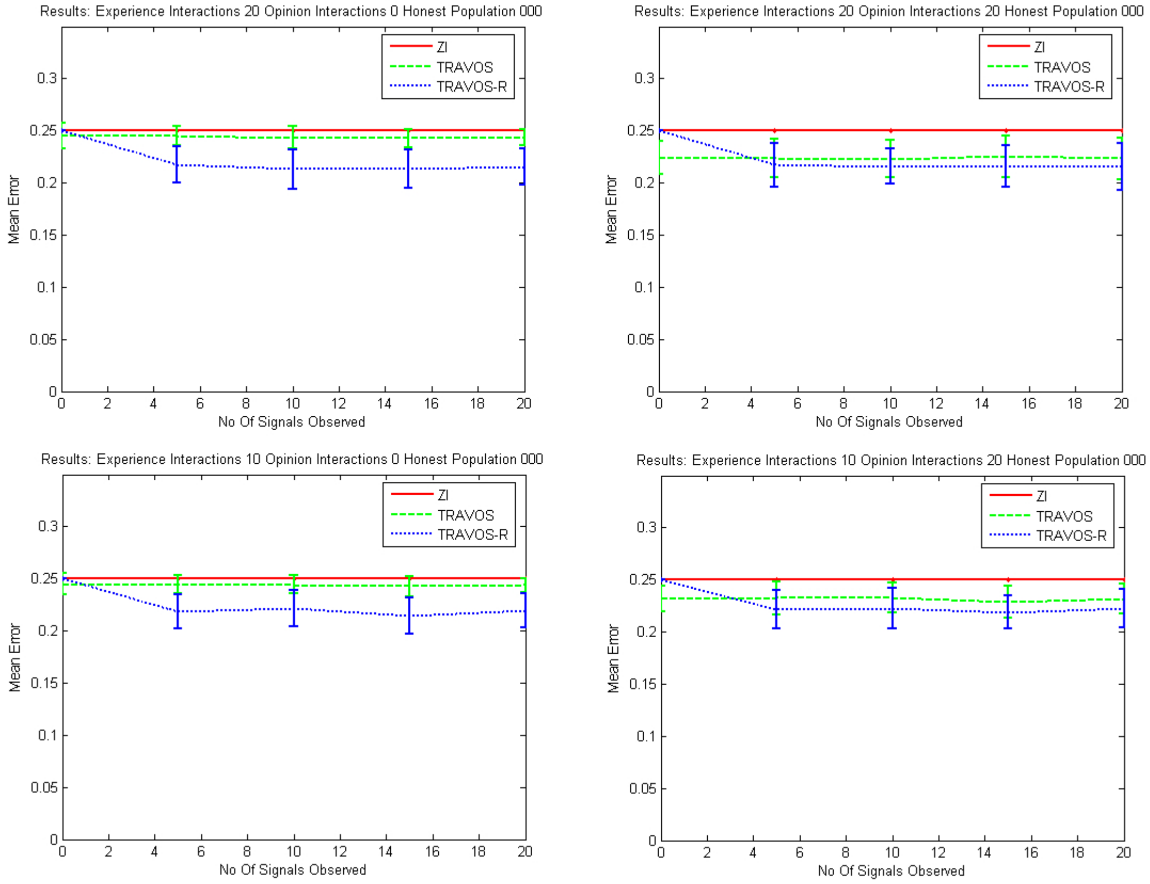

FIGURE 6.4: Plots showing the results obtained from environments containing a 100% Biased population.

However, it should be noted that, in the majority of cases, when the population consists of only BSPs, TRAVOS-R outperforms TRAVOS as expected. Results obtained for a subset of such environments are shown in Figure 6.4, which shows TRAVOS-R outperforming TRAVOS in environments containing 100% BSPs. The reason for the superior performance by TRAVOS-R in such environments is simply that it learns the relationships between agents, and so selects the appropriate relationship-based heuristics to combat the biased opinions. On the other hand, TRAVOS has no such method of combatting biased opinions, and as a result is misled by the opinions and therefore produces a larger mean error in its estimations. The process by which TRAVOS is sufficiently misled by the biased opinions is described in more detail below.

The strength of the TRAVOS mechanism for adjusting opinions relies on the assumption that an opinion provider agent employs a *fixed* strategy in providing opinions. That is, it always provides the same kind of opinion such as always false positives, or always false negatives, for all opinion requests. Simply, this means that as long as an opinion provider uses the same strategy (regardless of the agent requesting the opinion and the agent to whom the opinion applies) to provide all of its opinions, TRAVOS is able to perform relatively well. This is because it is able to *learn* a fixed strategy and, as shown in Chapter 4, by doing so it performs well.

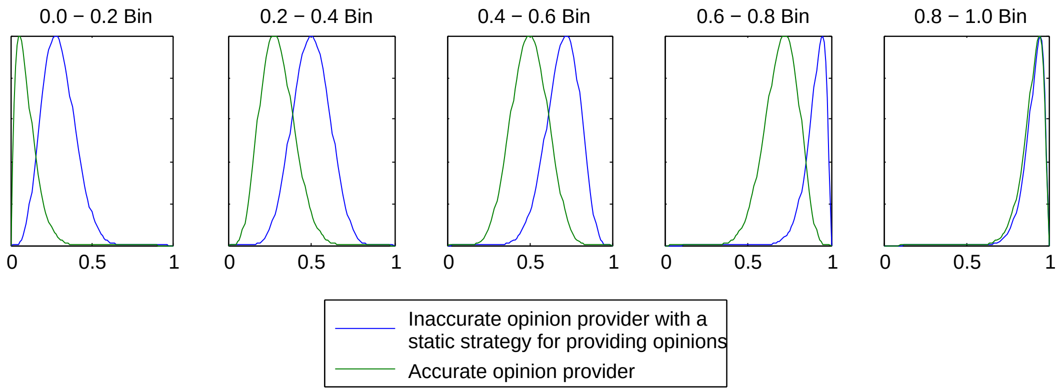

FIGURE 6.5: A TRAVOS consumer agent's opinion provision history bins for an opinion provider that provides inaccurate opinions using a static strategy.

More specifically, its learning strategy associates the opinion with the corresponding outcome observed in a set of bins that group together similar opinions (as discussed in Section 3.4.2). Such fixed behaviour leads to the bins recording useful information, which can be used to adjust opinions appropriately. For example, Figure 6.5 shows a set of bins belonging to a TRAVOS consumer agent that has encountered a particular opinion provider employing a fixed strategy in providing opinions. In this case, the opinion provider overestimates the trust level represented by the opinion before providing this false positive opinion. It does this for all opinion requests. As Figure 6.5 shows, the TRAVOS consumer agent's bins for this opinion provider are skewed to the right.[6]

Now, if the opinion provider employs a *range* of strategies to provide opinions, for example by providing a number of false positives, false negatives and honest opinions, it is able to confuse the TRAVOS mechanism. As a result, the information held in the bins tends to become more uniform as shown in Figure 6.6. Here, all the bins contain a similar plot, indicating that the TRAVOS consumer is unable to accurately *interpret* the report of the opinion provider when it provides a particular opinion, and fails to adjust it accordingly.

Then, as the number of biased service agents increases, the number of agents dynamically generating opinions (based on shared relationships) increases too, creating the dynamic environment in which TRAVOS is misled. In such cases, however, the TRAVOS-R mechanism is able to continue to adjust the opinions using the appropriate relationship-based heuristic to counteract the dynamic strategies. The relationship-based heuristics (described in Section 5.5) allow the TRAVOS-R agent to modify the biased opinions by reducing or removing the bias that may have been introduced into them. The TRAVOS-R agent knows which heuristic to apply as it learns the relationships (that cause the bias) between the service provider population, through observing signals generated by interacting service providers.

[6]The plots are skewed to the right of plots compared to what would have been obtained had the opinion provider provided accurate opinions.

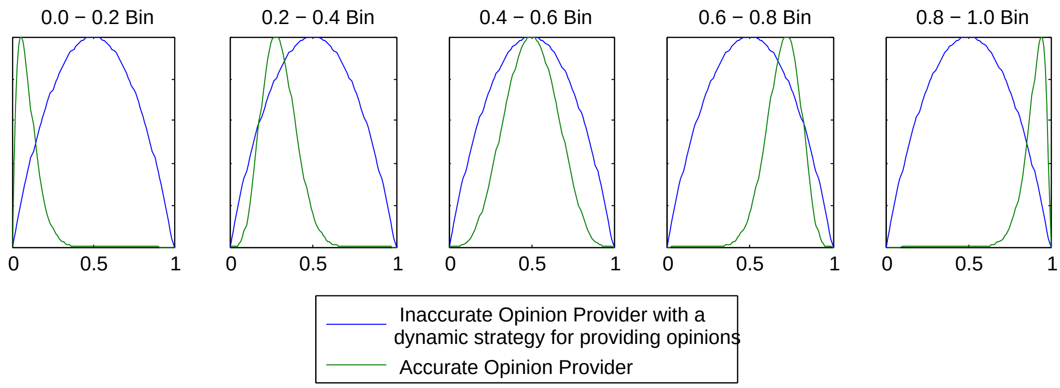

FIGURE 6.6: A TRAVOS consumer agent's opinion provision history bins for an opinion provider that provides inaccurate opinions using a dynamic strategy.

However, this ability of TRAVOS-R to adjust opinions using relationship-based heuristics has its limitations. As the percentage of accurate agents increases in the service provider population, the relative performance of TRAVOS improves, and in environments where there are sufficient opinion experience interactions (here this is $n_{oi} > 5$), it is able to outperform TRAVOS-R (the impact of varying experience interactions on the performance of TRAVOS and TRAVOS-R is discussed further in the discussion of Hypothesis 5). The poorer performance by TRAVOS-R is a result of its inability to distinguish accurately between ASPs and BSPs. In particular, the model assumes all opinions are biased, and therefore applies the appropriate relationship-based heuristic to the adjustment of the opinion. This adds noise to an otherwise honest and accurate opinion, causing it to calculate a level of trust with increased error. On the other hand, the TRAVOS mechanism prevents this from happening, keeping the information contained within the honest opinion intact, and thus allowing the agent to calculate a level of trust with less error.

6.3.3 Changing the Number of Signals Observed

TRAVOS-R learns the existence of relationships that can cause bias in opinion provision through the observation of interacting agents. More specifically, in our evaluation it does so by observing signals that are produced by interacting service providers. In real applications, the agent employing the TRAVOS-R mechanisms may or may not have the opportunity to observe many such signals. For this reason, it is important to examine how well it performs as the number of signals that it observes is varied ($n_{si} \in \{0, 5, 10, 15, 20\}$), and in what cases is it able to outperform TRAVOS.

In high trust information environments, as n_{si} increases, the performance of TRAVOS-R should increase. The strength with which the agent applies the heuristic, to adjust the opinion, is related to the confidence the agent has in its belief that a particular relationship type exists. As confidence reaches its maximum, so too does the strength at which the heuristic is applied, after which

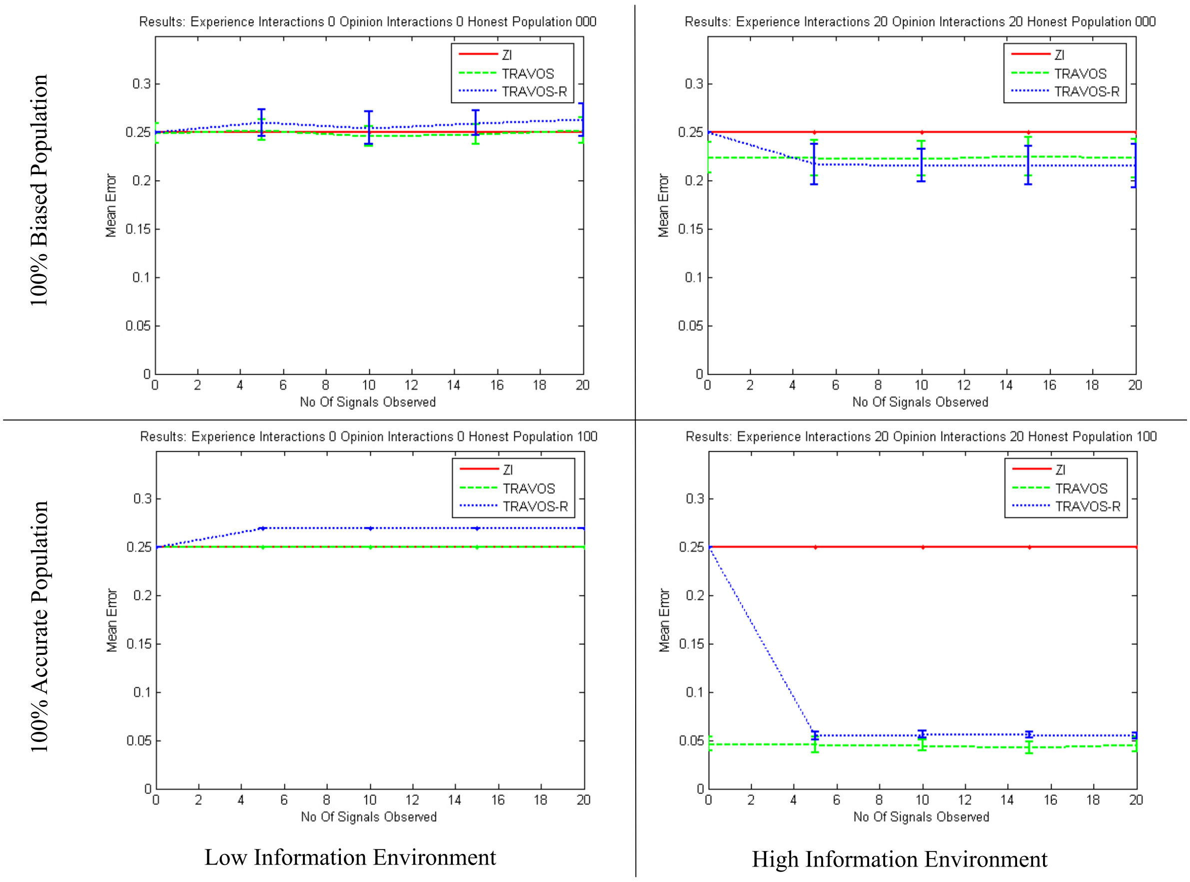

FIGURE 6.7: Plots showing the results obtained from varying the number of signals observed in environments where there is high and low trust information.

there is no further change in the adjusted opinion. Therefore, as more signals are observed, the improvement in performance will plateau as it gains maximum confidence in the relationships that it identifies after observing signals. More formally, we state this as a hypothesis below.

Hypothesis 3
In high trust information environments, initially, as n_{si} increases, the performance of TRAVOS-R will increase. This improvement will plateau after a certain level of n_{si} and will remain stable thereafter. In environments with no trust information, the performance of the TRAVOS-R agent will be similar to that of TRAVOS and the Zero Intelligence models.

The results that have been obtained from testing this hypothesis show that as the number of signals observed by the TRAVOS-R consumer agent increases, there is a change in its performance. Initially, as predicted in Hypothesis 3, there is a dramatic increase in performance (as much as a decrease of 0.2 in the mean error in its estimates). Figure 6.7 shows a number of result plots from different environments, where this initial increase in performance can be seen.

Observing just one signal is sufficient to trigger the TRAVOS-R mechanism into making conclusions about what relationships may be present. As more signals are observed, the conclusion about existing relationships are reinforced, and the confidence the mechanism has in these beliefs is increased. However, once the TRAVOS-R consumer agent believes that a particular relationship exists with a certain degree of confidence, then more signals that reinforce this do not have much of an impact. This is an inherent property of the Bayesian learning that forms part of the TRAVOS-R mechanisms. Ultimately, it is this that is the cause of the plateaus that can be seen in each of the plots in Figure 6.7 (after five signals being observed), after which the performance of TRAVOS-R is not affected by observing more signals.

In contrast to what was expected in Hypothesis 3, the results show that in environments consisting of little or no trust information and a 100% ASP population (again shown in Figure 6.7 as the plot for the low trust information environments), an increase in n_{si} actually causes the TRAVOS-R mechanism to perform worse. Initially, having observed no signals it performs as well as the TRAVOS and the Zero Intelligence consumers. However, once it starts to observe signals it begins to create a picture of relationships between agents, and uses appropriate relationship-based heuristics to adjust the opinions it receives. For an accurate opinion, this adds further error as the opinion is distorted, and leads to significantly poorer performance, as can be seen in the plot for the 100% ASPs population in a low information environment in Figure 6.7. In contrast, the corresponding plot for the 100% BSPs population shows that in a BSPs environment, the performance is more similar to the Zero Intelligence and TRAVOS-R models. Here, the heuristics enable it to adjust the biased opinions back to the neutral opinions that they should be, given that there is no trust information in the environment.

Overall, these results suggest that in an environment with a majority BSP population, and low or high trust information, TRAVOS-R is able to significantly outperform TRAVOS. In environments consisting entirely of ASPs and little trust information, TRAVOS-R performs significantly worse. That is to say, in systems where there is little or no possibility of biased opinions being provided, TRAVOS-R should not be used as it will lead to poor system performance.

6.3.4 Changing the Number of Experience Interactions

Both TRAVOS-R and TRAVOS are trust models that take into account the dimension of reputation through obtaining opinions, and then calculate the level of trustworthiness of the (target) agent to whom the opinion applied. In such models it is important to understand the impact of changing the number of experience interactions, n_{ei}, that the opinion providing agents have with the target agents. This is because if a population has little or no experience interactions, then the opinions formed are not representative of the target agent's behaviour, making even an accurate opinion provider seem as though it is providing a misleading or inaccurate opinion. This leads most trust models to disqualify and heavily adjust accurate opinions. On the other hand, in an environment where there are numerous experience interactions, the opinions contain more information about the target agent's behaviour. Our expectations of TRAVOS-R's performance, in comparison to the other two, are formally stated as a hypothesis below.

> **Hypothesis 4**
> In environments where there are few or no experience interactions, all three consumer agents will perform similarly. As n_{ei} increases, so too will the relative performance of TRAVOS and TRAVOS-R.

The results show clearly that changing the number of experience interactions shared amongst the agents in the service provider population has an impact on both TRAVOS and TRAVOS-R, as per Hypothesis 4. In both cases, as n_{ei} increases, so does the performance of both the models. The reason for this performance increase is because overall the service provider population has more trust information as n_{ei} increases. At $n_{ei} = 0$ the service providers have not interacted with each other, so they are unable to make an informative judgement when asked for an opinion about a particular service provider by the consumer agent, and simply return a neutral opinion (ASP) or an opinion based on no information (BSP).

As there is an increase in n_{ei}, the amount of information that the service providers have to base their opinion on increases too. This allows them to supply opinions that contain more information about the target agent's actual behavior. So, in environments where the population from which opinions will be requested[7] share a lot of interactions, the trust models perform better.

Figure 6.8 shows how both TRAVOS and TRAVOS-R improve their performance as n_{ei} increases from 0 to 20. Here, it is clear that the performance at $n_{ei} = 20$ is significantly better than that obtained at $n_{ei} = 0$. The improvement in performance from 0 to 5, from 5 to 10, and from 10 to 15 is of a greater magnitude than that achieved between 15 and 20. This decrease in the rate of improvement tells us that there is a n_{ei} beyond which there is no further improvement in TRAVOS and TRAVOS-R. This limiting factor arises because after a sufficient number of experience interactions, the agents forming the opinions have sufficient interaction history to form accurate and representative opinions. As they gain more experience, beyond a level which is sufficient for an accurate opinion, the relative accuracy of their opinions increases very slowly.

An interesting point to note is that as n_{ei} increases, the performance of both the models increases at the same rate. Figure 6.8 clearly shows that as the experience interactions increase, the rate at which both the curves for TRAVOS and TRAVOS-R move towards the x-axis is constant. This tells us that this variable has the same positive impact on both models.

6.3.5 Changing the Number of Opinion Experience Interactions

We have examined how changing the population composition, the number of signals the consumer agent observes, and the number of experience interactions of the service provider popula-

[7] In our case this is the service provider population.

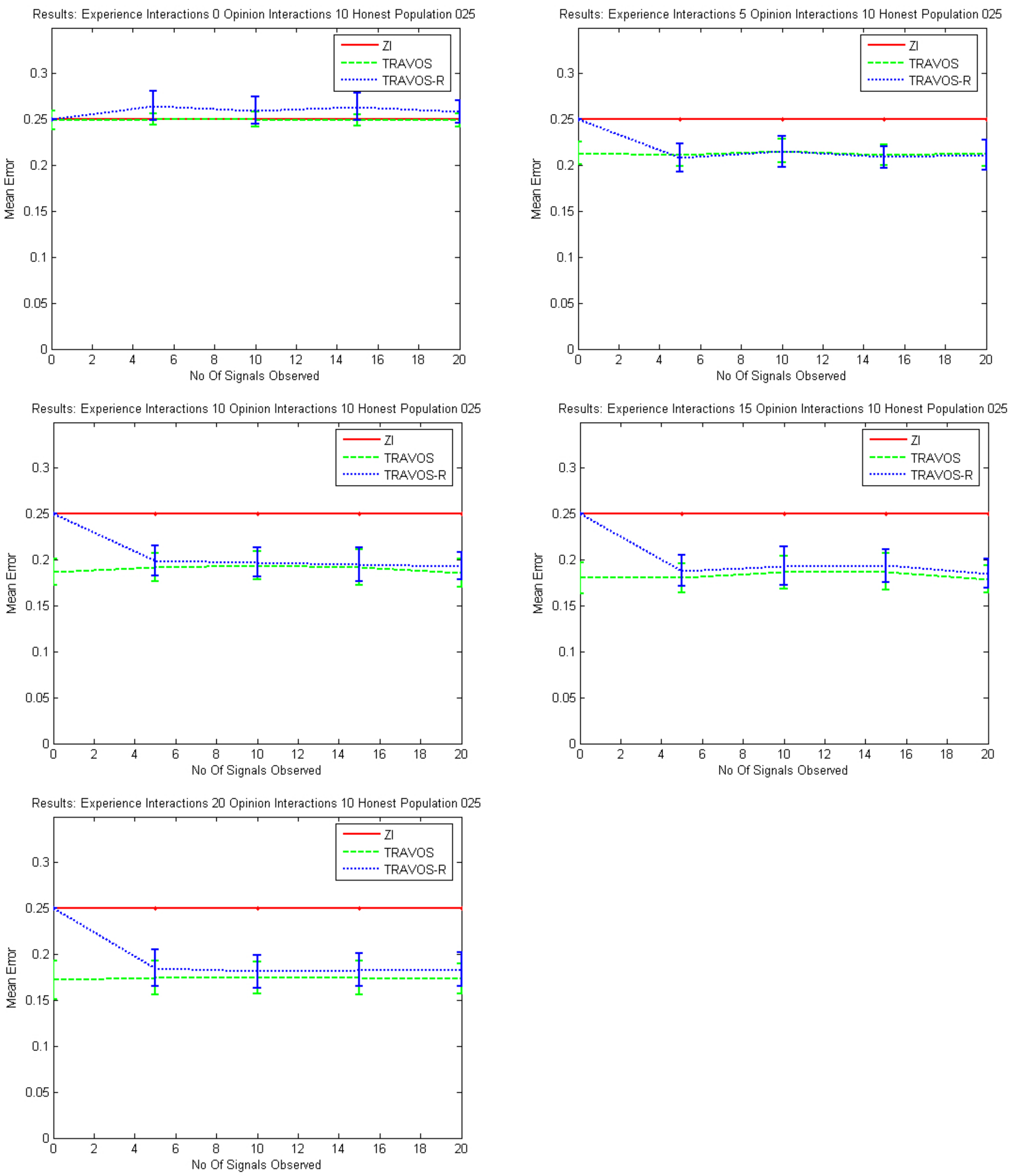

FIGURE 6.8: Plots showing a subset of results obtained from varying the number of experience interactions between the opinion providing population.

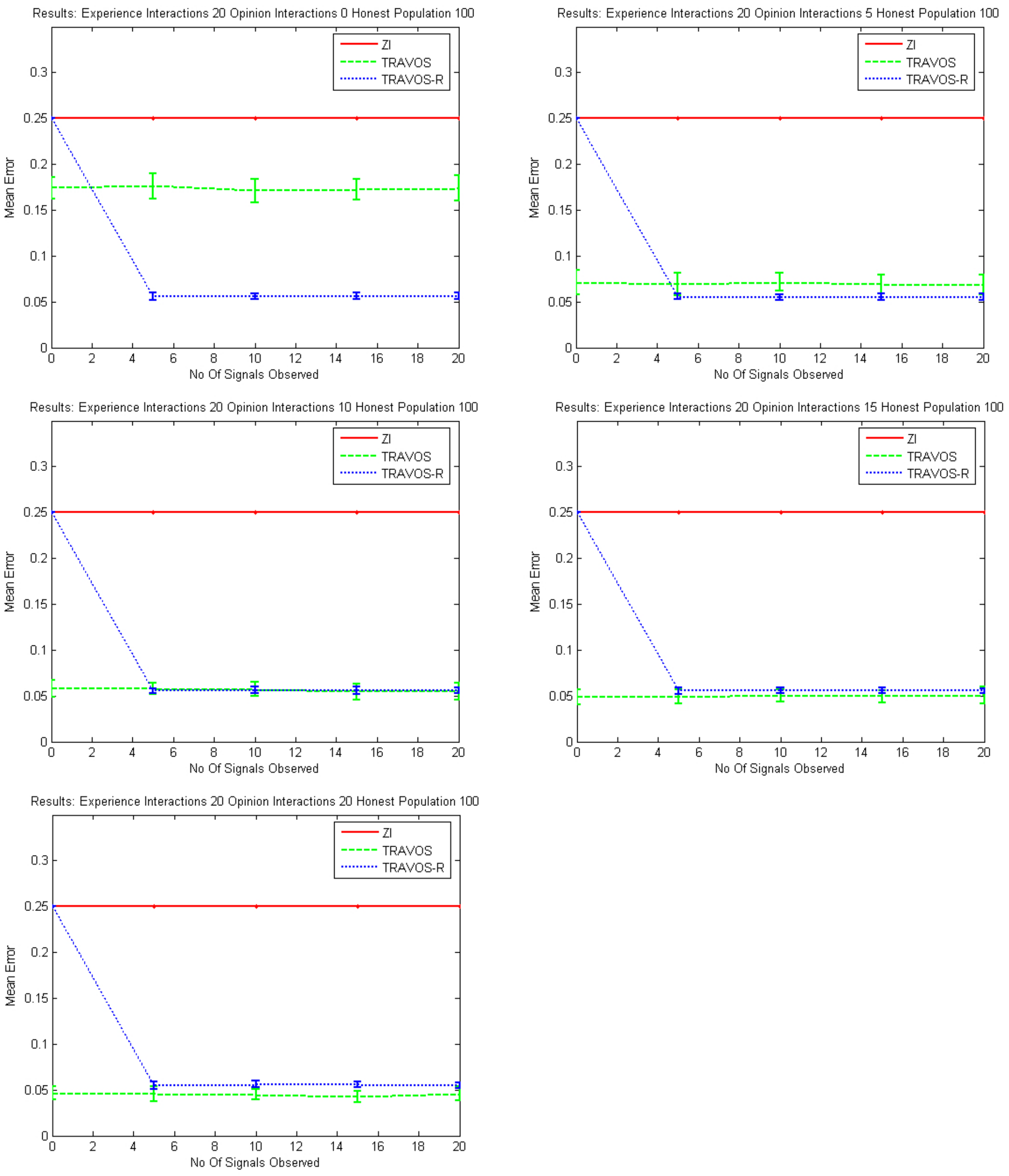

FIGURE 6.9: Plots showing how increasing opinion experience interactions leads to TRAVOS outperforming TRAVOS-R.

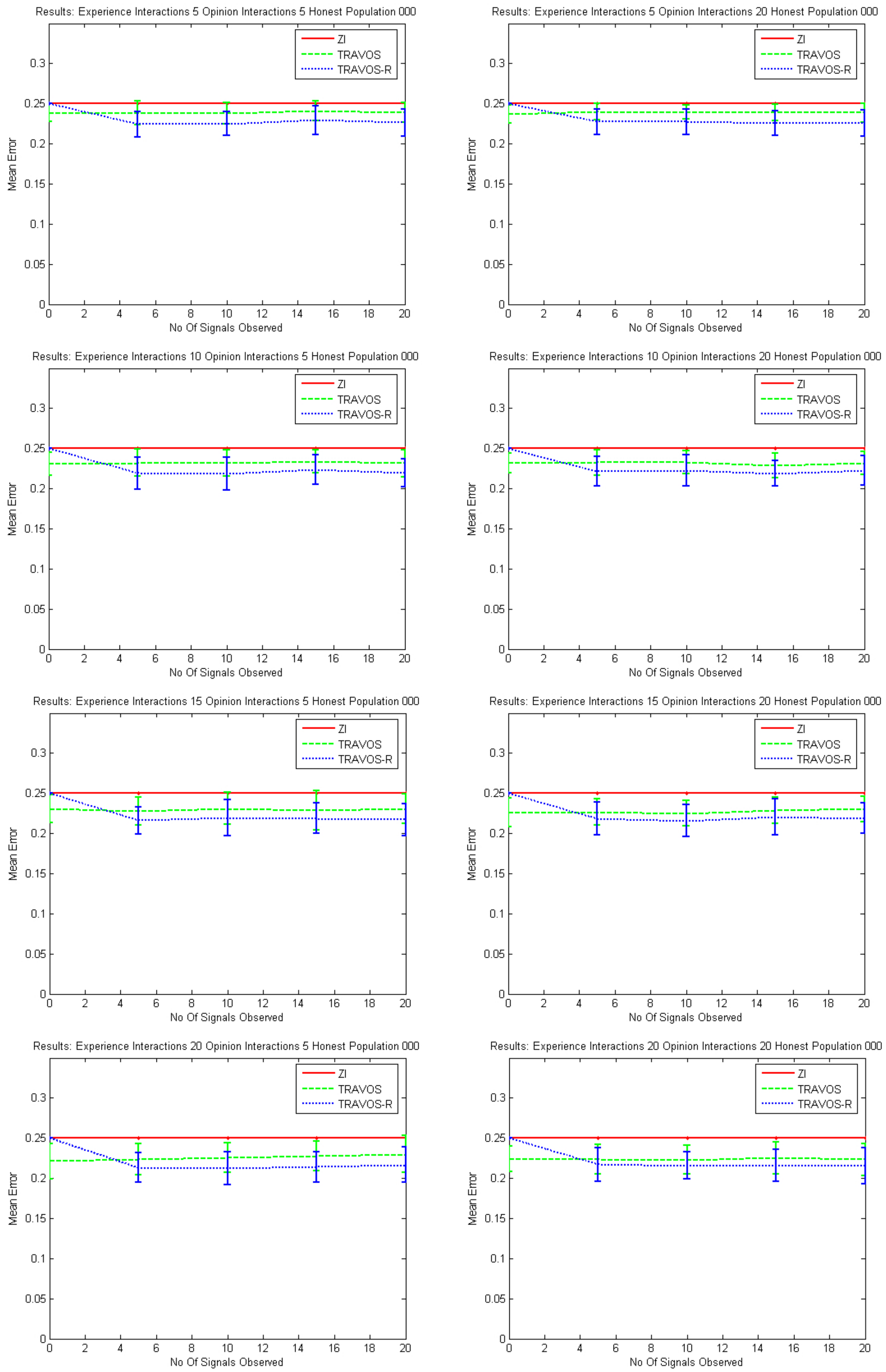

FIGURE 6.10: Plots showing how TRAVOS-R outperforms TRAVOS in environments where the majority of opinion providers are biased, regardless of increasing opinion experience interactions.

tion impacts on the performance of TRAVOS-R and TRAVOS. In doing so we have highlighted some of the differences and similarities in the two models. One particular difference in the two models is the fact that TRAVOS is able to learn from opinion provision episodes, and so improve its performance as it observes more acts of opinion provision. TRAVOS-R has no such learning mechanism, as its learning is concerned with identifying relationships from observations of agents' actions, and then selecting appropriate relationship-based heuristics to adjust opinions. To understand the impact of this, and the limitations that it may cause, we evaluate the performance of both models in a number of environments where we changed the number of opinion experience interactions, n_{oi}, observed by the TRAVOS consumer agent.

> **Hypothesis 5**
> The performance of the TRAVOS agent will increase as n_{oi} increases. After a certain n_{oi} TRAVOS will outperform TRAVOS-R.

Results obtained from testing the above hypothesis are described here. Unlike increasing n_{ei}, increasing n_{oi} leads to a faster rate of improvement in TRAVOS. TRAVOS is able to use these interactions as part of its learning process, and therefore shows an improvement in its performance. In fact, TRAVOS-R's performance remains unaffected by the change in n_{oi} as it does not use opinion experience in its mechanisms, and so does not learn anything about the nature of the opinion providers from them. Figure 6.9 shows that as n_{oi} increases from 0 to 20 TRAVOS improves its performance, and eventually after $n_{oi} = 10$ it marginally outperforms TRAVOS-R (by a mean error of 0.03).

However, there are environments where, regardless of the amount of opinion experience, TRAVOS-R outperforms TRAVOS, because TRAVOS is misled by a majority of biased opinions and cannot learn to distinguish good and bad opinion providers (as discussed in Section 6.3.2). This can clearly be seen in Figure 6.10, which shows that in environments where the majority of the service provider population consists of BSPs, TRAVOS-R outperforms TRAVOS regardless of increasing n_{oi}.

6.4 System Evaluation of TRAVOS-R

Whereas the previous sections presented a set of empirical results that identified the behavior of TRAVOS-R against TRAVOS in a variety of different environments, this section aims to show the feasibility of the TRAVOS-R mechanisms in a real application. The system evaluation of TRAVOS-R closely follows that of TRAVOS, which was presented in Section 4.2. We begin by modifying the scenario described in Section 4.2.1, and then in Section 6.4.2 we show how the TRAVOS-R mechanisms are used in such a scenario.

6.4.1 A Modified Agent-Based Virtual Organisation Scenario

We extend the scenario presented in Section 4.2.1 by introducing the notion of relationships. Relationships cannot arise in an agent-based system without reason, and in this scenario the reason for such relationships to exist is that they represent the real life relationships between the stakeholders of the agents in the system.

The modified scenario operates in exactly the same way (Stage 1 to Stage 7) except that we introduce certain relationships between agents in the system. The Yellow Pages (YP) agent and the User Agent (UA) are exempt from this relationship network. More specifically, we enable the Virtual Organisation Manager Agents (VOMs) and Service Provider Agents (SPs) to share relationships amongst themselves. The relationships that they can share are discussed below, alongside reasons for these relationships.

VOM competes with a service provider — Let us begin by considering a situation where the VOM and the SP are owned in the real world by different stakeholders. This could happen for a number of reasons. For example because the agents are owned by different individuals, or rival companies. This difference in ownership could induce a competitive relationship between the VOM and SP, because the VOM prefers not to award contracts for service provision to SPs that belong to other owners. A signal from this type of relationship may be that a VOM rejects a bid from a particular SP, even though the bid may be lower than some that have been received.

VOM cooperates with a service provider — Here, the relationship between the VOM and the SP could arise due to the similarities in their stakeholders. For example, if both the agents belong to the same company it is fair to assume that they will behave in a cooperative manner. Unlike the relationship described above, in this case the VOM may be inclined to give preference to SPs owned by its owner. A signal from this type of relationship may be that a VOM accepts a bid from a particular SP, even though the bid may inferior to some that have already been received.

VOM depends on the service provider — In our scenario, the dependence relationship arises after the VO is formed. Once this happens, the VOM is dependent on the service provider to provide the service as agreed so that the VOM can maintain its contract with the User Agent. A signal from this type of relationship may be that a VOM accepts a bid from a particular SP, and forms a contract for service provision.

VOM is depended upon by the service provider — Again, this dependence arises in a formed VO. In such a VO, the service provider is dependent on the VOM to continue using its services, and not to replace its service with others that may be more cost effective for the VO. A signal from this type of relationship may be that a VOM accepts a bid from a particular SP, and the SP requests special penalty clauses in the service provision contract.

In the following section we describe how the TRAVOS-R mechanisms can be used in a scenario like the one described above.

6.4.2 Applying TRAVOS-R in the Scenario

Here, we detail how the TRAVOS-R mechanisms can be applied in a system where agents share relationships, and where these relationships dictate some aspects of their behaviour. Initially we describe, in detail, a system which we use to demonstrate the application of TRAVOS-R. The section progresses to a walkthrough showing how a particular agent is able to learn relationships through the observation of actions and signals from the interactions of others. This is followed by an analysis of how the knowledge of the relationship that an opinion provider has, leads the agent to appropriately adjust the opinion using a relationship-based heuristic.

We begin by stating that there exists a particular multi-agent system with a variety of agents from the scenario. We concentrate on the stages of the scenarios that are concerned with evaluating the trustworthiness of a service provider based on the opinions of others (VO formation stages 4, 5 and 6 from the scenario presented in Chapter 4.2). For this reason we present a system that contains a UA, two VOMs and a number of SP agents. More specifically, we use the following agents:

- One UA, a_{ua1}.
- Two VOMs, a_{vom1} and a_{vom2}.
- Four SPs (for simplicity, we assume that each SP in the system has the ability to provide only one service):
  - a_{sp1} and a_{sp2} providing the phone call service.
  - a_{sp3} and a_{sp4} providing the HTML content service.

Furthermore, in Table 6.2 we define the relationships in the system, and observable signals that are produced as a result of these relationships. For example, from Table 6.2 we can see that a_{vom2} depends on a_{sp4}, and due to this relationship, observers watching their interaction will observe a signal of type $S4$.

Agent	**Relationship with a_{vom2}**	**Signals Produced**
a_{sp1}	Competes	$S1$
a_{sp2}	Cooperates	$S2$
a_{sp3}	Depends on	$S3$
a_{sp4}	Depended upon	$S4$

TABLE 6.2: The relationships shared by various SPs and a_{vom2}, and the signals produced as a result of the relationships.

Agent	Signal Type			
	S1	**S2**	**S3**	**S4**
a_{sp1}	6	0	0	0
a_{sp2}	0	5	0	0
a_{sp3}	0	0	6	0
a_{sp4}	0	0	0	9

TABLE 6.3: Agent a_{vom1}'s history of signals observed from interactions between the various SPs and a_{vom2}.

We begin by stating that a user requires a composite multimedia service, consisting of phone call and HTML content provision service, thereby creating a need for a VO to form. Firstly, the a_{ua1} sends the details of this requirement (which specifies the manner in which the user wishes to receive the service) as a query to a_{vom1}. Upon receiving the request to supply a composite service, a_{vom1} realises that the only way in which it can meet this requirement is if it forms a VO consisting of members that can supply the component services. Agent a_{vom1} queries the YP and finds out that currently, in the system, there are two agents capable of providing the phone call service and two agents that can provide the HTML content service. Unfortunately, a_{vom1} has shared no previous interactions with any of these four SPs. However, a_{vom1} has observed a number of signals produced from their interactions (in independent episodes) with another VOM (a_{vom2}). The history of signal observations is shown in Table 6.3 where, for example, the table shows that a_{vom1} has observed on five separate occasions signals of type $S2$ occurring between a_{vom2} and a_{sp2}.

Now, a_{vom1} has to select one of the phone call providers and one of the HTML content providers. Furthermore, a_{vom1} would like to choose the most trustworthy SP. The following sections describe how the agent is able to use the mechanisms described in Chapter 5 to select the most trustworthy partners for the VO. We begin by describing the process by which a_{vom1} learns particular transient relationships in the system.

6.4.3 Learning Transient Relationships Through Observations

Table 6.3 shows a summary of which signals have been observed by a_{vom1} from the interactions that agent a_{vom2} has had with each of the service providers that a_{vom1} is considering forming a VO with. By observing a signal in an interaction episode between two agents, a_{vom1} is able to create a picture of what transient relationship ($\hat{\mathcal{L}}^{a_{vom2},a_{sp1}}$) exists between those two agents in that one episode. However, to do this the agent has to be configured with some prior knowledge. Therefore, we assume that a_{vom1} has conditional probability tables that allows it to obtain the probability of observing a signal given the presence of a particular type of relationship, as shown in Table 6.4.

Now, using the mechanism described in Section 5.4.3, we show how a_{vom1} is able to calculate what type of transient relationship existed for an interaction between a_{vom2} and a SP. More

Relationship	p(Signal | Relationship)			
	S1	**S2**	**S3**	**S4**
Competes	0.5	0.1	0.2	0.3
Cooperates	0.2	0.6	0.1	0.1
Depends on	0.2	0.2	0.5	0.1
Depended upon	0.1	0.1	0.2	0.4

TABLE 6.4: Agent a_{vom1}'s conditional probability tables showing the probability of a signal given the presence of a certain type of relationship.

specifically, we illustrate the use of this mechanism by considering a certain interaction episode between a_{vom2} and a_{sp1} that led to a signal of type $S1$ being produced.[8]

$\hat{\mathcal{L}}^{a_{vom2},a_{sp1}}$	P($\hat{\mathcal{L}}^{a_{vom2},a_{sp1}}$)
$\hat{\mathcal{L}}_{com}^{a_{vom2},a_{sp1}}$	0.25
$\hat{\mathcal{L}}_{cop}^{a_{vom2},a_{sp2}}$	0.25
$\hat{\mathcal{L}}_{dep}^{a_{vom2},a_{sp3}}$	0.25
$\hat{\mathcal{L}}_{dep}^{a_{sp4},a_{vom2}}$	0.25

TABLE 6.5: The prior distribution of $\hat{\mathcal{L}}^{a_{vom2},a_{sp1}}$. The distribution represents a_{vom1}'s beliefs about what type of transient relationship exists between a_{vom2} and a_{sp1} in a particular interaction episode.

In this case, before observing the signal, a_{vom1} believes that each type of transient relationship between a_{vom2} and a_{sp1} is equally likely. Therefore, a_{vom1} starts with a uniform distribution for $\hat{\mathcal{L}}^{a_{vom2},a_{sp1}}$ as shown in Table 6.5. We denote this the prior distribution.

After observing a signal of type $S1$, a_{vom1} is able to calculate the posterior distribution for $\hat{\mathcal{L}}^{a_{vom2},a_{sp1}}$, and modify its beliefs about the transient relationship. This calculation involves using Table 6.4:

Using Table 6.4: $p(S1|\hat{\mathcal{L}}^{a_{vom2},a_{sp1}} = \hat{\mathcal{L}}_{com}^{a_{vom2},a_{sp1}}) = 0.5$
Using Table 6.5: $p(\hat{\mathcal{L}}^{a_{vom2},a_{sp1}} = \hat{\mathcal{L}}_{com}^{a_{vom2},a_{sp1}}) = 0.25$
Using Equation 5.1: $p(\hat{\mathcal{L}}^{a_{vom2},a_{sp1}} = \hat{\mathcal{L}}_{com}^{a_{vom2},a_{sp1}}) \propto 0.25 \times 0.5 = 0.125$

Using Table 6.4: $p(S1|\hat{\mathcal{L}}^{a_{vom2},a_{sp1}} = \hat{\mathcal{L}}_{cop}^{a_{vom2},a_{sp1}}) = 0.2$
Using Table 6.5: $p(\hat{\mathcal{L}}^{a_{vom2},a_{sp1}} = \hat{\mathcal{L}}_{cop}^{a_{vom2},a_{sp1}}) = 0.25$
Using Equation 5.1: $p(\hat{\mathcal{L}}^{a_{vom2},a_{sp1}} = \hat{\mathcal{L}}_{cop}^{a_{vom2},a_{sp1}}) \propto 0.25 \times 0.2 = 0.05$

Using Table 6.4: $p(S1|\hat{\mathcal{L}}^{a_{vom2},a_{sp1}} = \hat{\mathcal{L}}_{dep}^{a_{vom2},a_{sp1}}) = 0.2$
Using Table 6.5: $p(\hat{\mathcal{L}}^{a_{vom2},a_{sp1}} = \hat{\mathcal{L}}_{dep}^{a_{vom2},a_{sp1}}) = 0.25$
Using Equation 5.1: $p(\hat{\mathcal{L}}^{a_{vom2},a_{sp1}} = \hat{\mathcal{L}}_{dep}^{a_{vom2},a_{sp1}}) \propto 0.25 \times 0.2 = 0.05$

Using Table 6.4: $p(S1|\hat{\mathcal{L}}^{a_{vom2},a_{sp1}} = \hat{\mathcal{L}}_{dep}^{a_{sp1},a_{vom2}}) = 0.1$
Using Table 6.5: $p(\hat{\mathcal{L}}^{a_{vom2},a_{sp1}} = \hat{\mathcal{L}}_{dep}^{a_{sp1},a_{vom2}}) = 0.25$
Using Equation 5.1: $p(\hat{\mathcal{L}}^{a_{vom2},a_{sp1}} = \hat{\mathcal{L}}_{dep}^{a_{sp1},a_{vom2}}) \propto 0.25 \times 0.1 = 0.025$

[8] Table 6.3 shows that in total there were six occasions on which a_{vom1} observed a signal of type $S1$ from an interaction between a_{vom2} and a_{sp1}. We choose one of these six occasions to illustrate the use of the mechanism.

Finally, after normalising the answers we obtain the posterior distribution:

$\hat{\mathcal{L}}^{a_{vom2},a_{sp1}}$	P($\hat{\mathcal{L}}^{a_{vom2},a_{sp1}}$)
$\hat{\mathcal{L}}^{a_{vom2},a_{sp1}}_{com}$	0.5
$\hat{\mathcal{L}}^{a_{vom2},a_{sp2}}_{cop}$	0.2
$\hat{\mathcal{L}}^{a_{vom2},a_{sp3}}_{dep}$	0.2
$\hat{\mathcal{L}}^{a_{sp4},a_{vom2}}_{dep}$	0.1

From the calculations above we can see that the posterior distribution is significantly different from the (uniform) prior. The most likely value for $\hat{\mathcal{L}}^{a_{vom2},a_{sp1}}$ from the posterior distribution (after observing signal $S1$) is $\hat{\mathcal{L}}^{a_{vom2},a_{sp1}}_{com}$ with a probability of 0.5. For this reason, agent a_{vom1} believes that in this particular interaction episode agent a_{vom2} and a_{sp1} shared a transient relationship of type *competitive*. We do not show the calculations for the other observations of signals from interactions of the different SPs with a_{vom2}, and instead we simply state that:

- Agent a_{vom1} believes that on *six* occasions a_{vom2} was competing with a_{sp1}.
- Agent a_{vom1} believes that on *five* occasions a_{vom2} was cooperating with a_{sp2}.
- Agent a_{vom1} believes that on *six* occasions a_{vom2} was depending on a_{sp3}.
- Agent a_{vom1} believes that on *nine* occasions a_{sp4} was depending on a_{vom2}.

The next section illustrates how, at the end of each observed episode, agent a_{vom1} can use its beliefs about the type of transient relationship to modify its belief about the permanent relationship.

6.4.4 From Transient Relationships to Permanent Ones

Agent a_{vom1} stores the nature of the permanent relationship between a_{vom2} and SPs as a vector of beta distributions [9]. In this scenario there are four different relationship types (as discussed in Section 6.4.1), and so this vector consists of four beta distributions. Each such distribution corresponds to a particular type of permanent relationship that agent a_{vom1} believes to exist between the interacting agents.[10]

Now, prior to observing any interaction episodes of other agents, a_{vom1}'s permanent relationship vectors contain uniform beta distributions.

[9] We denote a beta distribution as (α, β).
[10] We denote the beta distribution that applies to a particular type of relationship with that type in the subscript of its parameters; for example, the beta distribution in the vector corresponding to a competitive relationship is written as $(\alpha_{com}, \beta_{com})$.

For example, a_{vom1}'s permanent relationship vector representing the relationship between a_{vom2} and a_{sp1} prior to any observations, is as follows.

$\mathbf{R}^{a_{vom2},a_{sp1}} =< (\alpha_{com}, \beta_{com}), (\alpha_{cop}, \beta_{cop}), (\alpha_{dep}, \beta_{dep}), (\alpha_{dep}, \beta_{dep}) >$
$= < (1,1), (1,1), (1,1), (1,1) >$

The method by which a_{vom1} updates each vector is simple (as described in Section 5.4.4). Going back to the calculation we started in the previous section, we showed that for a particular interaction episode that produced a certain signal, agent a_{vom1} believed that the interacting agents (a_{vom2} and a_{sp1}) were operating under a competing transient relationship. Now, after arriving at this belief, a_{vom1} must incorporate it into its belief about the permanent relationship between a_{vom2} and a_{sp1}. It does this as follows.

Agent a_{vom1}'s permanent relationship vector (for relationship between a_{vom2} and a_{sp1}) prior to observing any interactions between a_{vom2} and a_{sp1} is:

$\mathbf{R}^{a_{vom2},a_{sp1}} =< (\alpha_{com}, \beta_{com}), (\alpha_{cop}, \beta_{cop}), (\alpha_{dep}, \beta_{dep}), (\alpha_{dep}, \beta_{dep}) >$
$= < (1,1), (1,1), (1,1), (1,1) >$

Agent a_{vom1} observes an interaction episode and arrives at the belief that for that episode a_{vom2} and a_{sp1} shared a competitive relationship, and increases the α parameter of the element in the vector corresponding to competitive relationship type, giving:

$\mathbf{R}^{a_{vom2},a_{sp1}} =< (2,1), (1,1), (1,1), (1,1) >$

Furthermore, the agent increases the β parameter of all the other elements:

$\mathbf{R}^{a_{vom2},a_{sp1}} =< (2,1), (1,2), (1,2), (1,2) >$

As can be seen, the relationship vector is changed each time an agent believes it has witnessed a transient relationship of a particular type. Once again, we do not show the all the calculations for all the other observations made by a_{vom1}. Instead, based on the statements made at the end of the previous section, we summarise the permanent relationships vectors between the SPs and a_{vom2} as:

$\mathbf{R}^{a_{vom2},a_{sp1}} =< (7,1), (1,7), (1,7), (1,7) >$

$\mathbf{R}^{a_{vom2},a_{sp2}} =< (1,6), (6,1), (1,6), (1,6) >$

$\mathbf{R}^{a_{vom2},a_{sp3}} =< (1,7), (1,7), (7,1), (1,7) >$

$\mathbf{R}^{a_{vom2},a_{sp4}} =< (1,10), (1,10), (1,10), (10,1) >$

Finally, after observing the many different interaction episodes, and arriving at the permanent relationship vectors above, agent a_{vom1} is able to determine exactly what type of relationships

($\mathcal{L}_{type}$) are present between the four SPs and a_{vom2}. It does this by calculating the expected values of all the beta distributions in the relationship vectors and then selecting the corresponding type that has the highest expected value.

Using Equation 5.2:

$\mathbf{R}^{a_{vom2},a_{sp1}} =< (\frac{7}{7+1})),(\frac{1}{1+7}),(\frac{1}{1+7}),(\frac{1}{1+7}) >=< 0.88, 0.12, 0.12, 0.12 >$

$\mathbf{R}^{a_{vom2},a_{sp2}} =< (\frac{1}{1+6}),(\frac{6}{6+1}),(\frac{1}{1+6}),(\frac{1}{1+6}) >=< 0.14, 0.86, 0.14, 0.14 >$

$\mathbf{R}^{a_{vom2},a_{sp3}} =< (\frac{1}{1+7}),(\frac{1}{1+7}),(\frac{7}{7+1}),(\frac{1}{1+7}) >=< 0.12, 0.12, 0.88, 0.12 >$

$\mathbf{R}^{a_{vom2},a_{sp4}} =< (\frac{1}{1+10}),(\frac{1}{1+10}),(\frac{1}{1+10}),(\frac{10}{10+1}) >=< 0.1, 0.1, 0.1, 0.9 >$

The type of the relationship present is the one which has a corresponding element that has the highest expected value,. Therefore, a_{vom1} believes that the following types of relationship exist between the SPs and a_{vom2}:

$\mathcal{L}^{a_{vom2},a_{sp1}} = \mathcal{L}^{a_{vom2},a_{sp1}}_{com}$

$\mathcal{L}^{a_{vom2},a_{sp2}} = \mathcal{L}^{a_{vom2},a_{sp1}}_{cop}$

$\mathcal{L}^{a_{vom2},a_{sp3}} = \mathcal{L}^{a_{vom2},a_{sp1}}_{dep}$

$\mathcal{L}^{a_{sp4},a_{vom2}} = \mathcal{L}^{a_{sp4},a_{vom2}}_{dep}$

So far, we have shown how, from observation of interaction (between others in a society) an agent can learn the transient relationships, and from these transient relationship an agent can arrive at a belief about what type of permanent relationships exist. Now, in the next section, we illustrate how these beliefs are used in the trust calculations to ensure that the agent is not misled.

6.4.5 Applying a Relationship-Based Heuristic

We have demonstrated that a_{vom1} has certain beliefs about what types of relationships exist between the other VOM, a_{vom2}, and the four SPs (a_{sp1}, a_{sp2}, a_{sp3} and a_{sp4}). We stated earlier that a_{vom1} has no past interaction history with any of these SPs, so it is necessary for it to consult others' opinions to obtain a trust value for each of the SPs. Now, suppose that a_{vom1} approaches a_{vom2} and requests opinions about a_{sp1}, a_{sp2}, a_{sp3} and a_{sp4}.

Agent	$\mathcal{R}$	$E_{\mathcal{R}}$	$\hat{\mathcal{R}}$	$E_{\hat{\mathcal{R}}}$
a_{sp1}	(6,4)	0.6	(4,6)	0.4
a_{sp2}	(4,6)	0.4	(6,4)	0.6
a_{sp3}	(6,4)	0.6	(4,6)	0.4
a_{sp4}	(4,6)	0.4	(9,11)	0.45

TABLE 6.6: The opinions provided by a_{vom1} to mislead a_{vom1}.

In Table 6.2 we presented the actual relationships that a_{vom2} has with each of the SPs that it is being asked to provide opinions about. Now, due to these relationships, suppose that a_{vom2} is incentivised to provide misleading opinions about the SPs to a_{vom1}. More specifically, Table 6.6 shows the actual history of interaction ($\mathcal{R}$) between a_{vom2} and the SPs, and the trust portrayed ($E_{\mathcal{R}}$) by an honest opinion, alongside the trust portrayed ($E_{\hat{\mathcal{R}}}$) by the misleading opinions ($\hat{\mathcal{R}}$) it provides to a_{vom1}. From this table we can see that a_{vom2} provided opinions intended to mislead a_{vom1} into selecting inappropriate VO partners. For example, consider the choice a_{vom1} is faced with in choosing a phone call service provider, a_{sp1} or a_{sp2}. If a_{vom2} provided an honest opinion, then a_{vom1} would choose a_{sp1}, but due to the misleading opinion (without any adjustment), a_{vom1} will select a_{sp2} as the phone call service provider.

Agent a_{vom1} can use its knowledge of social relationships and relationship-based heuristics (described in Section 5.5) to adjust the opinions provided by a_{vom2}. We now illustrate how this can be achieved[11], and in so doing show how it results in a_{vom1} choosing the most trustworthy SP for the component service provision.

Using knowledge about the relationship between a_{vom2} and a_{sp1}

Agent a_{vom1} believes that the relationship between a_{vom2} and a_{sp1} is of type $\mathcal{L}_{com}^{a_{vom2},a_{sp1}}$. In this case, it uses Heuristic 1 to adjust the misleading opinion provided by a_{vom2}:

Using Heuristic 1 (Line 5), Table 6.6 and Equation 5.2:

$$E_{\hat{\mathcal{R}}^t_{a_vom2,a_sp1}} = \frac{4}{4+6} = 0.4$$

Using Heuristic 1 (Line 6) and $\delta^{a_vom2,a_sp1} = 0.6$:

$$Amount\ to\ adjust\ = 0.6 \times (0.4 \times (1\ -\ 0.4)) = 0.144$$

Using Heuristic 1 (Line 7):

$$\bar{E}_{\hat{\mathcal{R}}^t_{a_vom2,a_sp1}} = 0.4 + 0.144 = 0.544$$

Using knowledge about the relationship between a_{vom2} and a_{sp2}

Agent a_{vom1} believes that the relationship between a_{vom2} and a_{sp2} is of type $\mathcal{L}_{cop}^{a_{vom2},a_{sp2}}$. In this case, it uses Heuristic 2 to adjust the misleading opinion provided by a_{vom2}:

Using Heuristic 2 (Line 5), Table 6.6 and Equation 5.2:

$$E_{\hat{\mathcal{R}}^t_{a_vom2,a_sp2}} = \frac{6}{6+4} = 0.6$$

Using Heuristic 2 (Line 6) and $\delta^{a_vom2,a_sp2} = 0.6$:

[11] For simplicity we assume that a_{vom1} has a confidence of 0.6 in all its beliefs about the relationships between a_{vom1} and the four SPs, and this is above the minimum confidence level required to use a heuristic.

$Amount\ to\ adjust\ = 0.6 \times (0.6 \times 0.6) = 0.216$

Using Heuristic 2 (Line 7):

$\bar{E}_{\hat{\mathcal{R}}^t_{a_vom2,a_sp2}} = 0.6 - 0.216 = 0.384$

Using knowledge about the relationship between a_{vom2} and a_{sp3}

Agent a_{vom1} believes that the relationship between a_{vom2} and a_{sp3} is of type $\mathcal{L}_{dep}^{a_{vom2},a_{sp3}}$. In this case, it uses Heuristic 3 to adjust the misleading opinion provided by a_{vom2}:

Using Heuristic 3 (Line 5), Table 6.6 and Equation 5.2:

$E_{\hat{\mathcal{R}}^t_{a_vom2,a_sp3}} = \frac{4}{4+6} = 0.4$

Using Heuristic 3 (Line 6) and $\delta^{a_vom2,a_sp3} = 0.6$:

$Amount\ to\ adjust\ = 0.6 \times (0.4 \times (1 \times 0.4)) = 0.144$

Since $E_{\hat{\mathcal{R}}^t_{a_vom2,a_sp3}} \leq 0.5$, we use Heuristic 3 (Line 8) to perform the final adjustment (we set the value of $\eta = 0.5$):

$\bar{E}_{\hat{\mathcal{R}}^t_{a_vom2,a_sp3}} = 0.4 + (0.5 \times 0.144) = 0.472$

Using knowledge about the relationship between a_{vom2} and a_{sp4}

Agent a_{vom1} believes that the relationship between a_{vom2} and a_{sp4} is of type $\mathcal{L}_{dep}^{a_{sp4},a_{vom2}}$. In this case it uses Heuristic 4 to adjust the misleading opinion provided by a_{vom2}:

Using Heuristic 4(Line 5), Table 6.6 and Equation 5.2:

$E_{\hat{\mathcal{R}}^t_{a_vom2,a_sp4}} = \frac{9}{20} = 0.45$

Using Heuristic 4 (Line 6) and $\delta^{a_vom2,a_sp4} = 0.6$:

$Amount\ to\ adjust\ = 0.6 \times (0.45 \times 0.45) = 0.12\ (2\ decimal\ places)$

Since $E_{\hat{\mathcal{R}}^t_{a_vom2,a_sp4}} < 0.5$, we use Heuristic 4 (Line 8) to perform the final adjustment:

$\bar{E}_{\hat{\mathcal{R}}^t_{a_vom2,a_sp4}} = 0.45 - 0.12 = 0.33$

Selecting the best SPs

Having calculated the adjusted trust level for each of the opinions provided by a_{vom2} (as summarised in Table6.7) a_{vom1} can now decide which agents to form a VO with to supply the composite service consisting of phone calls and HTML content provision.

Agent	$E_{\hat{\mathcal{R}}}$	$\bar{E}_{\hat{\mathcal{R}}}$
a_{sp1}	0.4	0.544
a_{sp2}	0.6	0.384
a_{sp3}	0.4	0.472
a_{sp4}	0.45	0.33

TABLE 6.7: The trust level portrayed ($E_{\hat{\mathcal{R}}}$) by the opinions provided by a_{vom2} to a_{vom1}, and a_{vom1}'s adjusted trust level ($\bar{E}_{\hat{\mathcal{R}}}$).

From the two phone call service providers a_{vom1} selects a_{sp1} (rather than a_{sp2}), since it has calculated a higher (compared to that of a_{sp2}) adjusted trust value ($\bar{E}_{\hat{\mathcal{R}}}$) for a_{sp1}. Finally, for the HTML content provision, a_{vom1} selects a_{sp3} (rather than a_{sp4}) for the same reason (that it has a higher adjusted trust value compared to a_{sp4}). If a_{vom1} had not used the knowledge of the relationships and the appropriate heuristics, then the opinions provided by a_{vom2} would have misled it to form a VO with agents a_{sp2} and a_{sp4}.

6.5 Summary

In this chapter we have shown how the TRAVOS-R mechanisms perform, both against the TRAVOS approach and in a system where agents provide opinions to mislead the truster. The former was shown by means of an empirical evaluation, and the latter by means of a system evaluation.

The chapter began by showing how TRAVOS-R was empirically evaluated, and discussed the main trends and results obtained from this evaluation. We outlined the experimental methodology (Section 6.1), which described how the experiments were run in a simulated environment. The results obtained from the evaluation were divided into two main branches (the entire set of results from the experiments, which are consistent with the representative samples discussed in this chapter, can be found in graphical form in Appendix A and B).

Firstly, the empirical results showed how TRAVOS-R performed with different configurations of prior information. More specifically, we evaluated TRAVOS-R's performance using different signal given relationship probability distributions. Here, the results showed that using an accurate signal given relationship distribution leads to better performance, and that even though using the wrong distribution resulted in poorer performance, it was still able to outperform the uniform. In fact, the performance of both accurate and wrong configurations was very similar. These results tell us that even though some signals may be encoded incorrectly by the system designer, as long as the probability of a signal given a relationship distribution contains some uniform values (see discussion of Hypothesis 1) the performance of the model is acceptable.

Secondly, the evaluation showed how the TRAVOS-R model performed against the TRAVOS model in a variety of different environments. These results were obtained by running the two models (with a control) in a number of different environments. In doing so the following observations were made from the results produced:

- Both models show improvements in performance as the percentage of biased providers in the service provider population falls. However, in environments where the majority of the population is composed of biased providers, TRAVOS-R is able to outperform TRAVOS.
- Varying the number of signals observed by TRAVOS-R results in a dramatic increase in performance (the maximum is from a mean error of 0.25 to 0.05). However, this increase is short lived, and the performance stabilises after a certain number of signals. In environments with low opinion experience interactions and biased providers, this increase is sufficient to allow it to significantly outperform TRAVOS. However, in overall low trust environments an increase in the the number of signals leads to the opposite, and causes a decrease in performance.
- Varying the number of experience interactions that the opinion providing population has (amongst themselves) causes an increase in performance of both models in all environments. This tells us that both TRAVOS-R and TRAVOS work better in environments where members of the environment share experiences with each other. This conclusion is intuitive as agents are likely to form better opinions about others if they have shared interactions with the agents to whom the opinions apply.
- Varying the number of opinion experience interactions that the TRAVOS mechanism receives before the experiments, leads it to substantially increase its performance. In environments where the opinion providing population consists only of ASPs, TRAVOS is able to significantly outperform TRAVOS-R. However, in environments where the majority of agents are BSPs, TRAVOS-R is able to significantly outperform TRAVOS regardless of the level of opinion experience interactions and experience interactions.

Overall, the empirical evaluation highlights the ability of TRAVOS-R, as expected, to perform well (better than TRAVOS) in environments where the majority of opinion providers are providing biased opinions. In addition, the performance of TRAVOS-R improves dramatically after observing just a few signals, meaning that a TRAVOS-R agent is able to accurately adjust opinions soon after the agent enters a new system. Finally, it is important to note that the TRAVOS-R mechanisms are not suitable in environments where there is low trust information and the population of opinion providers are honest. In such cases, the relationship-based heuristic falsely leads the TRAVOS-R agent to adjust accurate opinions, and leads to poor estimates of an agent's trustworthiness.

In the system evaluation (Section 6.4) we extended the agent-based virtual organisation scenario presented earlier (Section 4.2.1) to include relationships. Using this scenario, we showed how an an agent can use the TRAVOS-R mechanism to adjust opinions. More specifically, we showed that by adjusting the opinions using the right relationship-based heuristic, the truster is able to make the right choice in VO partners.

Through evaluating the TRAVOS-R mechanisms in the two ways described above, we have shown that it is beneficial for an agent to gather social information and use it in its trust calculations (Aim 4, stated in Section 1.3). We have demonstrated that not only does our novel approach address the limitations of current trust models that try to consider social information, but it does so in a manner that performs better than a model that does not use social information, and one which allows it to be easily deployed in a real application.

Chapter 7

Conclusions

This chapter provides a summary of the research carried out and presents a number of avenues for future research. Initially, we discuss the implications of our work and enumerate the key research achievements. The chapter concludes with a number of future research areas which were identified throughout this research, and which address some of the main limitations of the work presented.

7.1 Implications of TRAVOS and TRAVOS-R

Computing systems are becoming more open and complex, especially with the advent of technologies such as the Grid (Foster and Kesselman, 2004), Peer-to-peer computing (Oram and Oram, 2001), Semantic Web (Berners-Lee et al., 2001) and E-Commerce (Kalakota and Robison, 1999). In the context of these environments, open means that agents contained within such systems are free to enter and leave at their own will. Furthermore, many of the agents are owned by different real world stakeholders, and behave in a self-interested way toward one another. In such environments, agents have to make sound decisions to achieve their goals, but are faced with a large amount of uncertainty in this decision-making process. Much of this uncertainty is caused by the self-interested nature of their interaction partners, whose behaviour is dictated by their own goals.

Given this background, it is important in such environments to assure good interactions for agents if they are to successfully complete their goals. Trust is a concept integral to human society, allowing us to effectively make decisions in the presence of uncertainty. This thesis has used this as inspiration, and presented a novel computational model of trust, TRAVOS, for use in open agent systems and shown its application in agent-based virtual organisations in the Grid. In such environments, our trust model enables an agent to make effective and sound decisions in light of the inherent uncertainty that exists in such applications.

The main benefit of using TRAVOS is that it provides an agent with a set of mechanisms, depending on the evidence at hand, with which to assess the trustworthiness of a trustee. It provides mechanisms for assessing the trustworthiness of others in situations both in which the agents have interacted before and share past experiences, and in which there is little or no past experience between the interacting agents. The ability of an agent to select the most trustworthy interaction partner means it can maximise the probability that the interaction will be carried out as agreed, and minimise the effect of any harmful action from the interacting partner.

In situations where an agent's past experience with a trustee is low, it can draw upon opinions provided by others to calculate the trustee's reputation. However, in doing so, the agent risks lowering, rather than increasing, assessment performance due to inaccurate opinions. Given this, a key feature of TRAVOS is its ability to cope with this situation by having an initially conservative estimate in reputation accuracy. Through repeated interactions with individual opinion providers, it learns to distinguish reliable from unreliable sources. By empirical evaluation (Chapter 4), we have demonstrated that this approach allows reputation to be used to significantly improve performance, while guarding against the negative effects of inaccurate opinions. Moreover, TRAVOS can extract a positive influence on performance from reputation, even when 50% of opinion sources are intentionally misleading. This effect is increased significantly through repeated interactions with individual reputation sources. When 100% of opinion sources are misleading, reputation has a negative effect on performance. However, even in this case, performance is increased by learning, and it outperforms the most similar models in the literature, in the majority of scenarios tested.

Furthermore, TRAVOS (in particular its extension TRAVOS-R) extends the state of the art by incorporating social information into its trust calculations. In more detail, it allows an agent to learn the inter-agent relationships that are present in VO-rich environments. In so doing, it allows an agent to select the appropriate relationship-based heuristic, which allows it to adjust misleading opinions provided by biased opinion sources. In fact, we have shown (in Chapter 6) that by using social information, an agent is able to perform better (in an environment where there is some social information) than an agent that does not use such information.

In summary, the research presented in the thesis has achieved the following:

1. In response to the fact that no one state of the art model is capable of meeting all the requirements of a trust model for agent-based VOs, we developed a novel computational model, TRAVOS, that meets all the basic requirement. The TRAVOS model allows an agent to effectively arrive at a trust value, which represents the trustworthiness of an individual in a particular context, using a number of methods:

    (a) Arrive at a trust level using personal experience.

    (b) In cases where an agent has no personal experience, it calculates a trust level using the opinions provided by others in the society. In particular, we have addressed the aim of assessing the reliability of opinions provided by others, in an open system, by incorporating a novel mechanism that filters out misleading opinions. More specifically, an agent using TRAVOS is able to learn, over time, how reliable (and accurate) an opinion provider is, and then use this information in adjusting the opinion provided by that opinion provider. The end result is that an opinion from an unreliable opinion provider is adjusted so that it has no (or little) impact on the reputation value of an individual. Furthermore, through empirical evaluation we have shown that our filtering mechanism is better than the one used by the most similar model in the literature.

   (c) In cases where the agent has little personal experience, it is able to combine both personal experience and the opinions of others to calculate a trust level using a confidence metric. This is a novel approach that allows an agent to make effective use of two evidence sources that it uses in its trust calculations.

2. We have described, for the first time, how a computational model of trust for agent-based virtual organisations can be deployed in a realistic application. In addition, we have demonstrated how a trust model can be employed to select appropriate virtual organisation partners, in the presence of uncertainty about the partner's behaviour and the accuracy of opinions.

3. We developed a novel taxonomy of inter-agent relationships, in agent-based VOs, through analysis of interactions between agents.

4. Using our relationships taxonomy, we have extended the state of the art in trust models by incorporating relationship information into trust calculations. In more detail, we extended TRAVOS to incorporate social information into trust calculations, so that an agent is able to do the following.

   (a) To learn beliefs about the type of temporary relationship that exists between two agents when they interact in a particular interaction episode.

   (b) Using these beliefs about the temporary relationships, we present a mechanism that allows the agents to learn the nature of the more permanent relationship that may exist between two agents (over a number of interaction episodes).

   (c) Finally, we present a novel set of relationship-based heuristics that allow an agent to adjust the opinions provided by opinion providers that it knows share a particular type of relationship with the trustee.

5. We have shown, for the first time, that a trust model can provide a more accurate trust level when it considers, in its trust calculations, the social information available in the environment.

   (a) Through empirical evaluation, we have shown that the TRAVOS-R mechanism for adjusting opinions (using relationships) results in a better empirical performance than the approach used in TRAVOS (using the perceived reliability of an opinion provider).

(b) Finally, we have shown (through a system evaluation) that using the TRAVOS-R mechanisms in agent-based VO environments prevents an agent being misled by the intention of others.

In closing, the remainder of this chapter presents ways in which the research contained in this thesis can be extended.

7.2 Further Research

The research presented in this thesis provides a solid basis for further research. Below we detail a number of avenues for further research, in which TRAVOS can be used as the base trust model upon which the proposed issues can be addressed.

Using complex social information — The TRAVOS-R mechanisms (Chapter 5) allow an agent to learn and use (in trust calculations) relationships that exist between two agents only. We believe that this approach can be extended and the impact of larger social structures on the trust calculations should be explored. As an interesting starting point, the model can be extended to learn the presence of structures such as those reviewed by Horling and Lesser (2004). We believe that the Bayesian approach used to learn inter-agent relationships in our model is limited to simple relationships. It is likely that the learning mechanism will have to be modified to allow it to learn larger social structures. We believe that this process can be replaced by a more complex Bayesian process, or by using an approach similar to that of Ashri et al. (2005).

Representing the trust of a group — In Section 2.3.6 we presented General Requirement 13 (VO-level trust), which is not addressed by any of the mechanisms presented in this thesis. To ensure that a trust model is adopted into the design of agent-based VO systems, it is important to provide this functionality. Currently, using our approach, agents can only assess other individuals. However, often in VO systems it is likely that an agent is faced with a decision of interacting with an entire VO. For example, consider an agent that wishes to consume a composite service. It can find individual service providers and form a VO with them, or it could simply find a VO that already provides this service and obtain it from the VO. In the latter case, the agent needs to be able to assess the trustworthiness of the entire VO to allow it to account for the uncertainty in whether or not the VO will successfully provide the composite service. Sabater and Sierra (2002) describe how to determine the trust level for an agent in a particular neighborhood (small social network) by forming a trust value representative of that neighborhood. This is a promising point of departure for measuring VO-level trust, but their approach is a simple weighted average function, and we believe a more complex function is required. In more detail, the function should consider the individual behaviour of the VO members, and the social ties that exist between the VO and other individuals outside the VO.

Trust Consensus — In addition, our model does not address General Requirement 12 (trust level consensus). Once again it is important to meet this requirement to ensure the success of a trust model in a VO-rich environment. Since agents in a VO work together, it can be assumed that on occasion, more than one agent may have to reach a decision together. When these decisions involve determining the trust level for a particular individual, it is important to allow the group of agents to arrive at a consensus trust level, which is representative of their collective beliefs about the trustworthiness of the individual. The study of argumentation-based negotiation (Rahwan et al., 2003) provides a point of departure. The trust model can be extended to allow an agent to use trust evidence as arguments, in negotiation with other agents in the VO, to arrive at a consensus trust level for a particular individual.

Having established, through this research, the first trust model for agent-based virtual organisations, we believe that the objectives outlined above have to be explored (and solutions developed) to ensure the adoption of such technology into real computer applications. If the challenges we have described are met, the resulting trust model will help promote the application of agent-based virtual organisation techniques to a range of open and dynamic domains.

Appendix A

Results From Evaluating TRAVOS-R with Different Bootstrap Configurations

In Section 6.2 we described how we evaluated the TRAVOS-R mechanisms with different bootstrap configurations, with the aim of understanding the impact of the pre-configured prior knowledge on the performance of the agent. More specifically, we developed three different bootstrap configurations (as described in 6.2.1), and evaluated them in a variety of environments (as described in 6.2.2). In this appendix we present the results obtained from each environment. For a detailed discussion of the general trends the results see Section 6.2.2.

The following is a description of the figures contained within this appendix.

Figure A.1	This figure shows the results obtained from evaluating three different TRAVOS-R configurations in environments consisting of 100% accurate opinion providers.
Figure A.2	This figure shows the results obtained from evaluating three different TRAVOS-R configurations in environments consisting of 100% biased opinion providers.

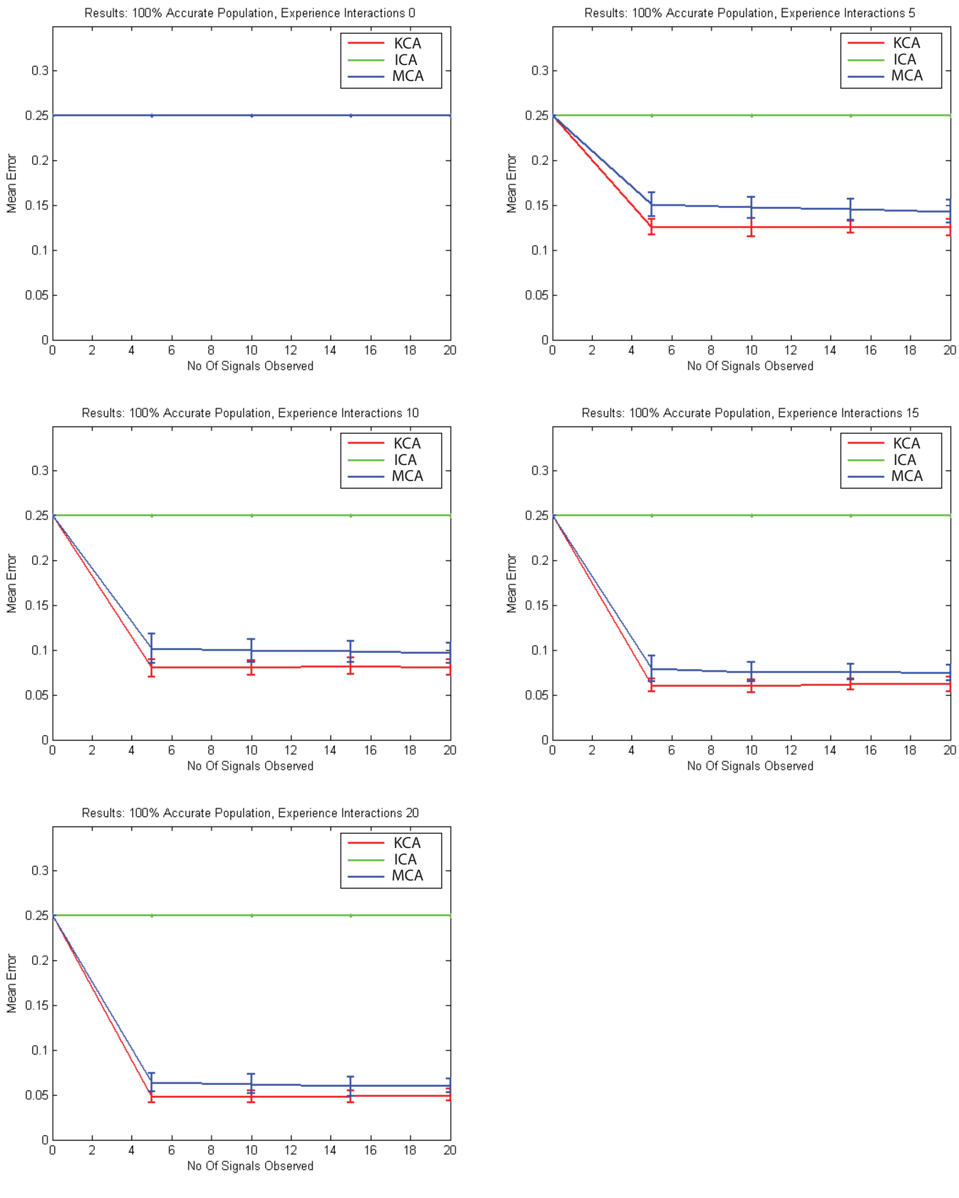

FIGURE A.1: Results of evaluating different TRAVOS-R configurations in environments consisting of 100% accurate opinion providers.

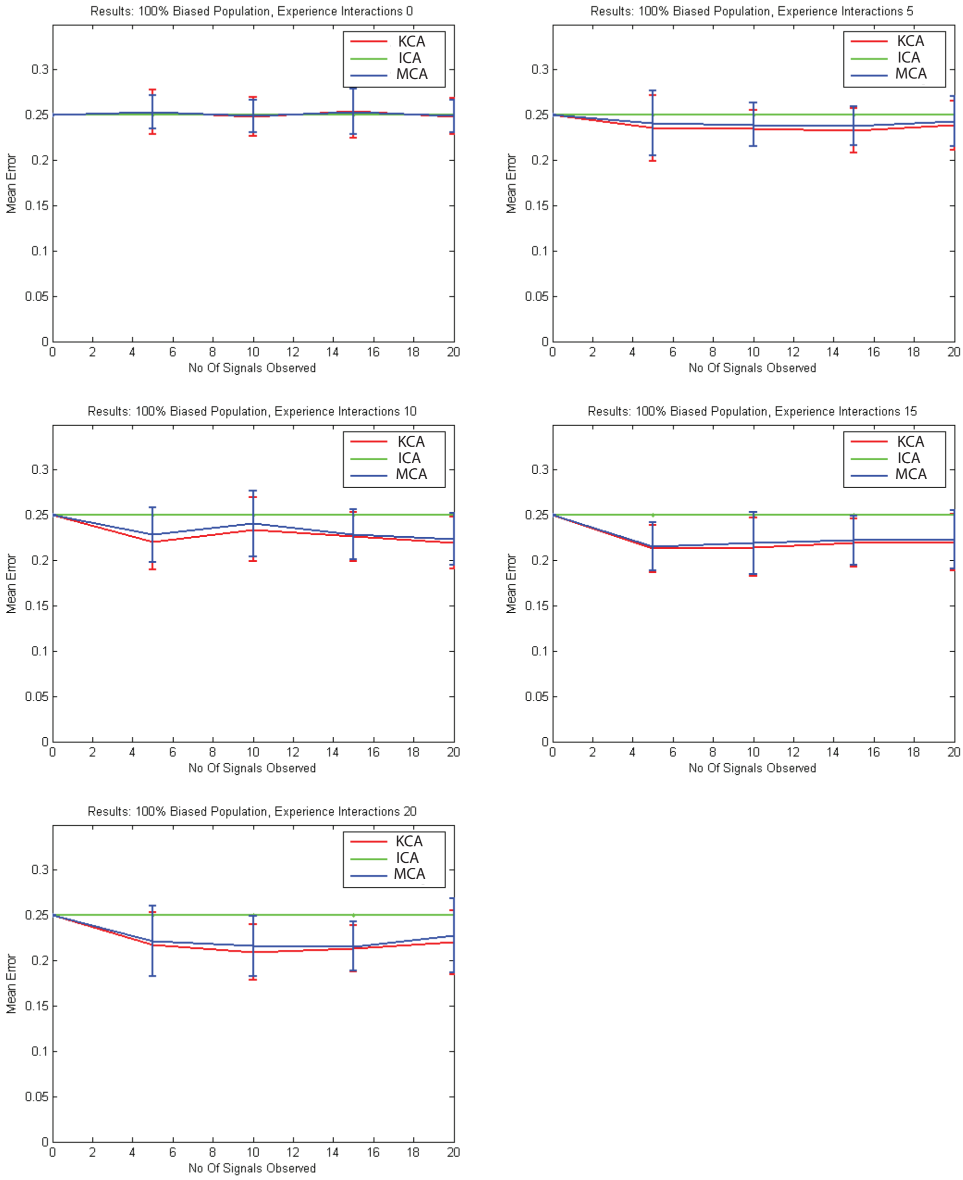

FIGURE A.2: Results of evaluating different TRAVOS-R configurations in environments consisting of 100% biased opinion providers.

Appendix B

Results From Evaluating TRAVOS-R in Different Environments

In Section 6.3 we described how we evaluated the performance of TRAVOS-R against TRAVOS, in a variety of environments, and using a subset of results we discussed the general trends observed. For completion, in this appendix we provide, in graphical form, the complete set results obtained from the environments that TRAVOS and TRAVOS-R were tested in. For a detailed discussion of the results see Section 6.3.

The following is a description of the figures contained within this appendix.

Figure B.1 This figure shows the results from evaluating TRAVOS and TRAVOS-R in environments where the number of experience interactions is 0, the number of opinion experience interactions is 0, and the percentage of accurate opinion providers varies from 0% to 100%.

Figure B.2 This figure shows the results from evaluating TRAVOS and TRAVOS-R in environments where the number of experience interactions is 0, the number of opinion experience interactions is 10, and the percentage of accurate opinion providers varies from 0% to 100%.

Figure B.3 This figure shows the results from evaluating TRAVOS and TRAVOS-R in environments where the number of experience interactions is 0, the number of opinion experience interactions is 20, and the percentage of accurate opinion providers varies from 0% to 100%.

Figure B.4 This figure shows the results from evaluating TRAVOS and TRAVOS-R in environments where the number of experience interactions is 10, the number of opinion experience interactions is 0, and the percentage of accurate opinion providers varies from 0% to 100%.

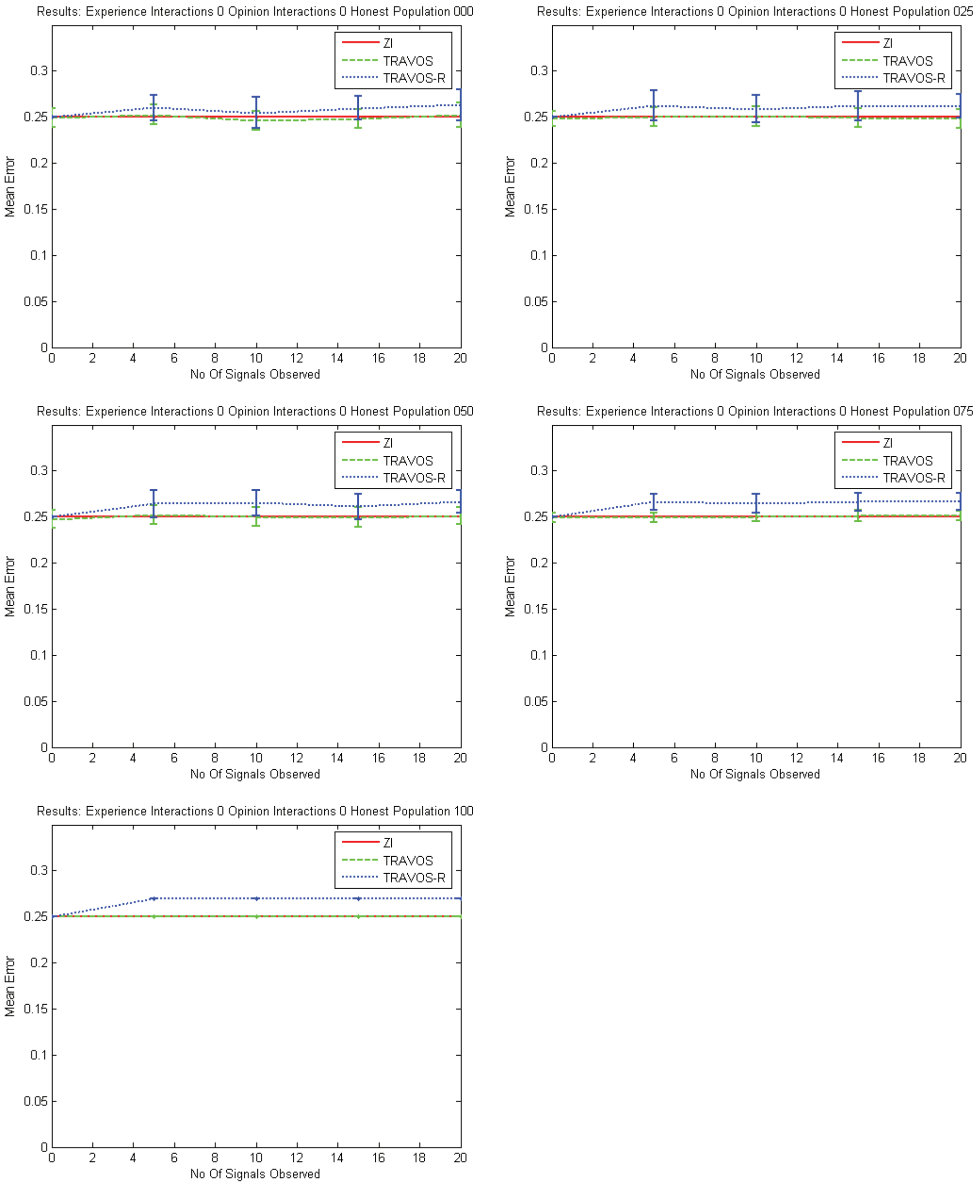

FIGURE B.1: Results of evaluating TRAVOS and TRAVOS-R in environments where the number of experience interactions is 0, the number of opinion experience interactions is 0, and the percentage of accurate (honest) opinion providers varies from 0% to 100%.

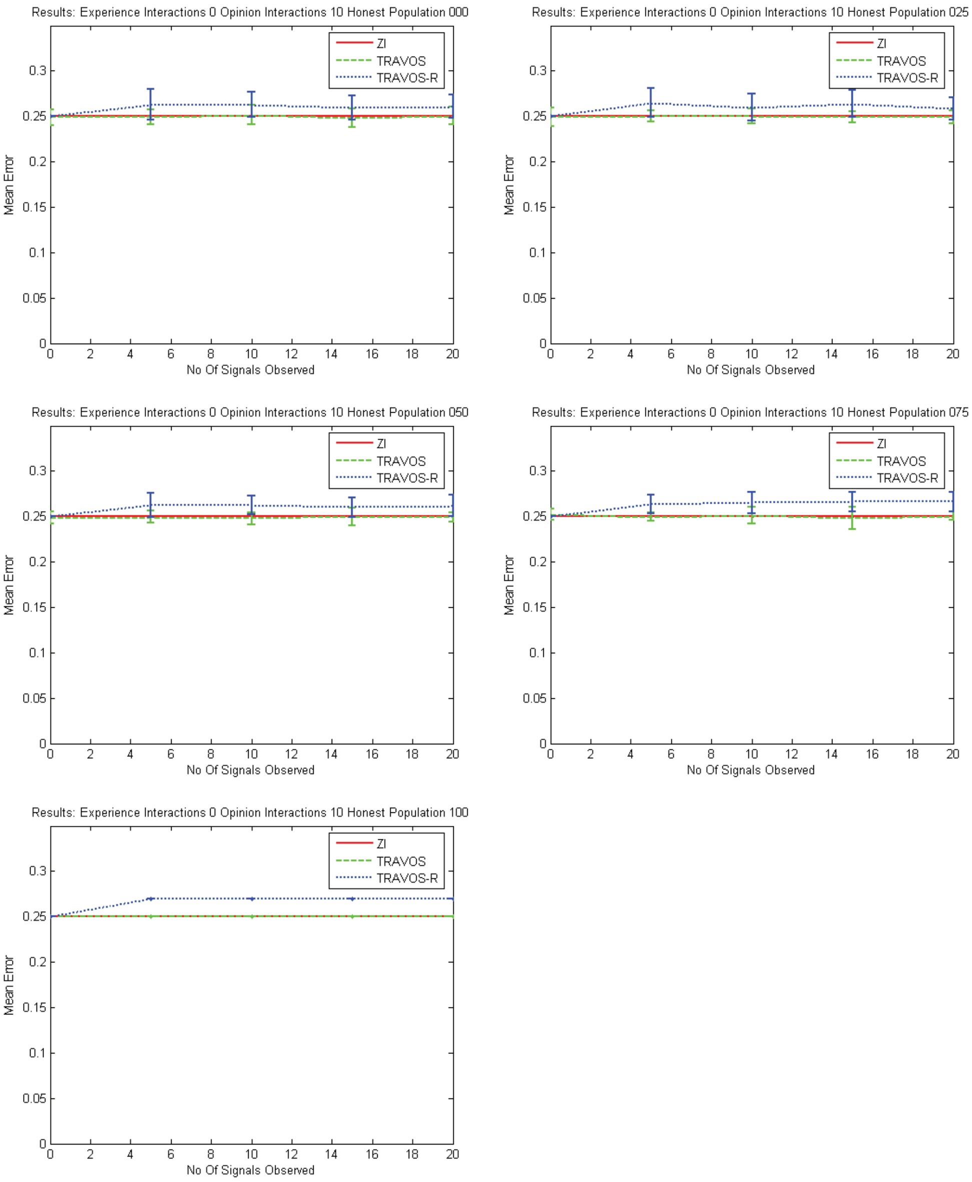

FIGURE B.2: Results of evaluating TRAVOS and TRAVOS-R in environments where the number of experience interactions is 0, the number of opinion experience interactions is 10, and the percentage of accurate (honest) opinion providers varies from 0% to 100%.

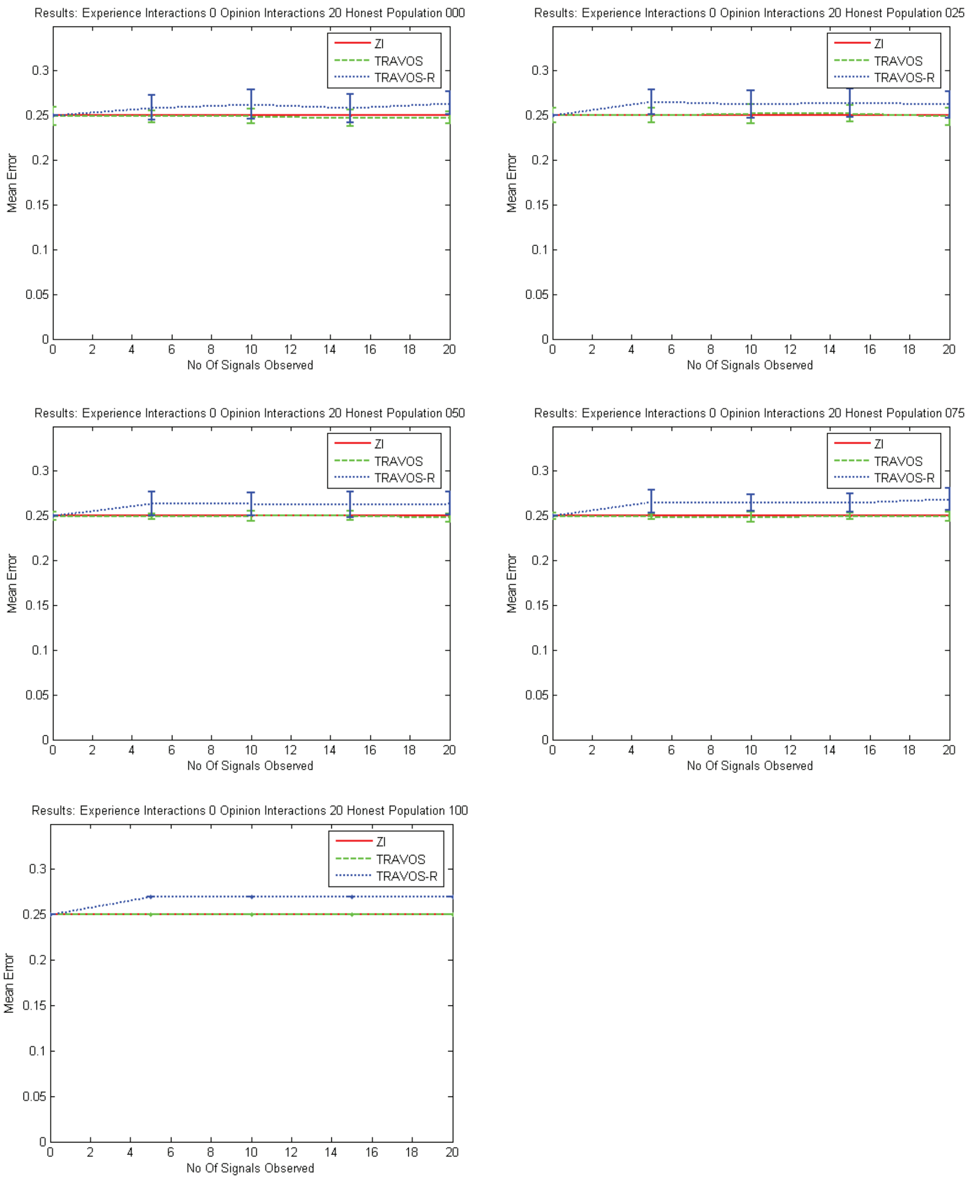

FIGURE B.3: Results of evaluating TRAVOS and TRAVOS-R in environments where the number of experience interactions is 0, the number of opinion experience interactions is 20, and the percentage of accurate (honest) opinion providers varies from 0% to 100%.

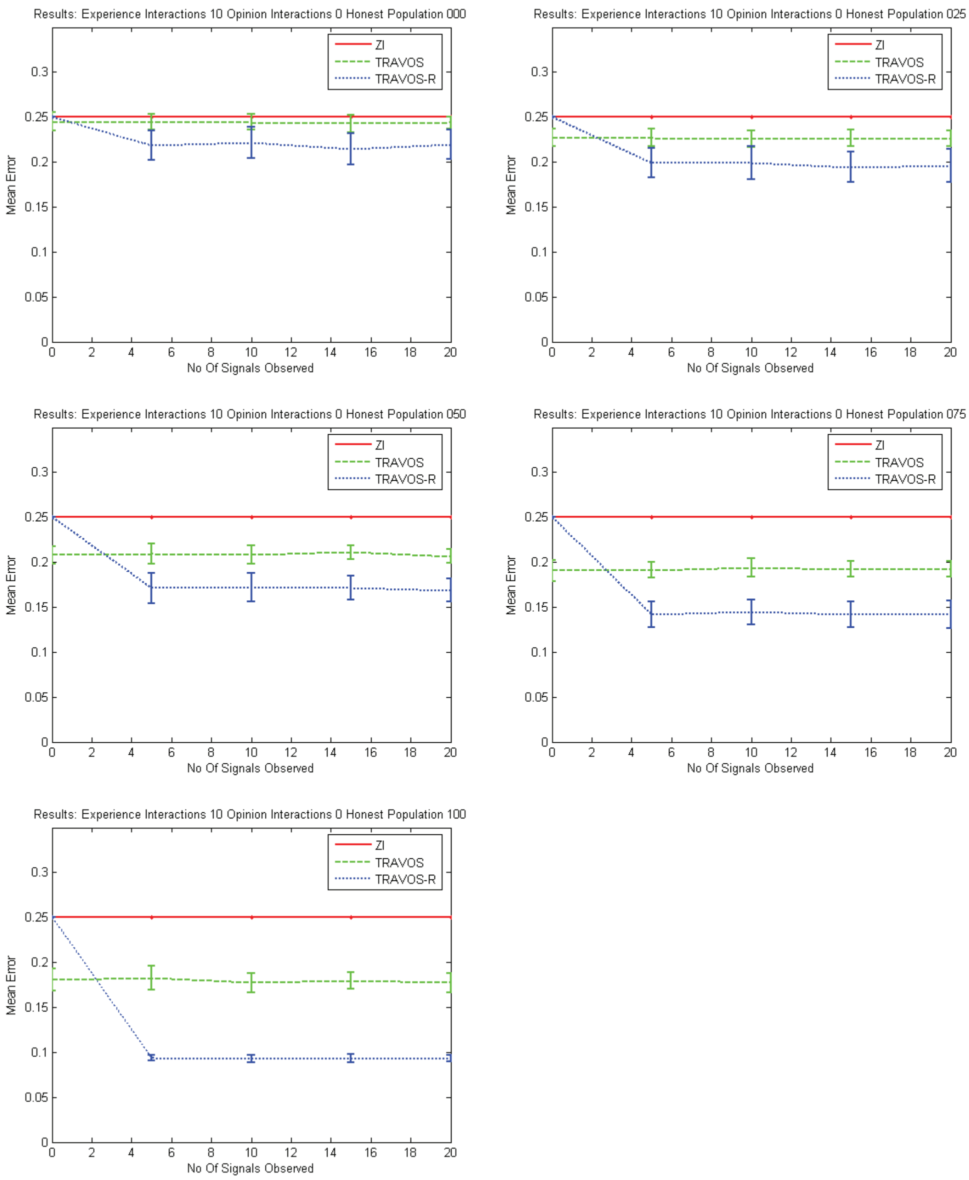

FIGURE B.4: Results of evaluating TRAVOS and TRAVOS-R in environments where the number of experience interactions is 10, the number of opinion experience interactions is 0, and the percentage of accurate (honest) opinion providers varies from 0% to 100%.

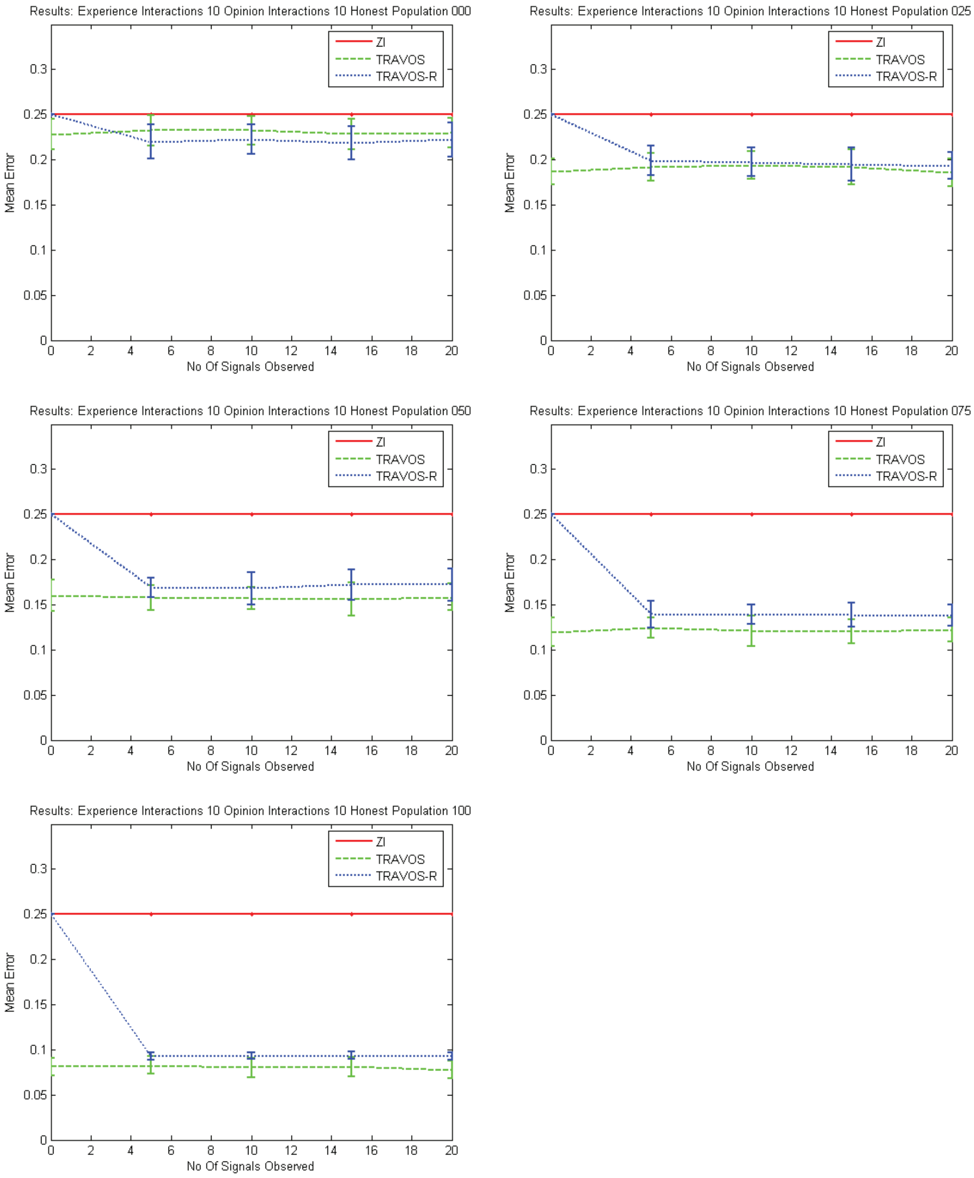

FIGURE B.5: Results of evaluating TRAVOS and TRAVOS-R in environments where the number of experience interactions is 10, the number of opinion experience interactions is 10, and the percentage of accurate (honest) opinion providers varies from 0% to 100%.

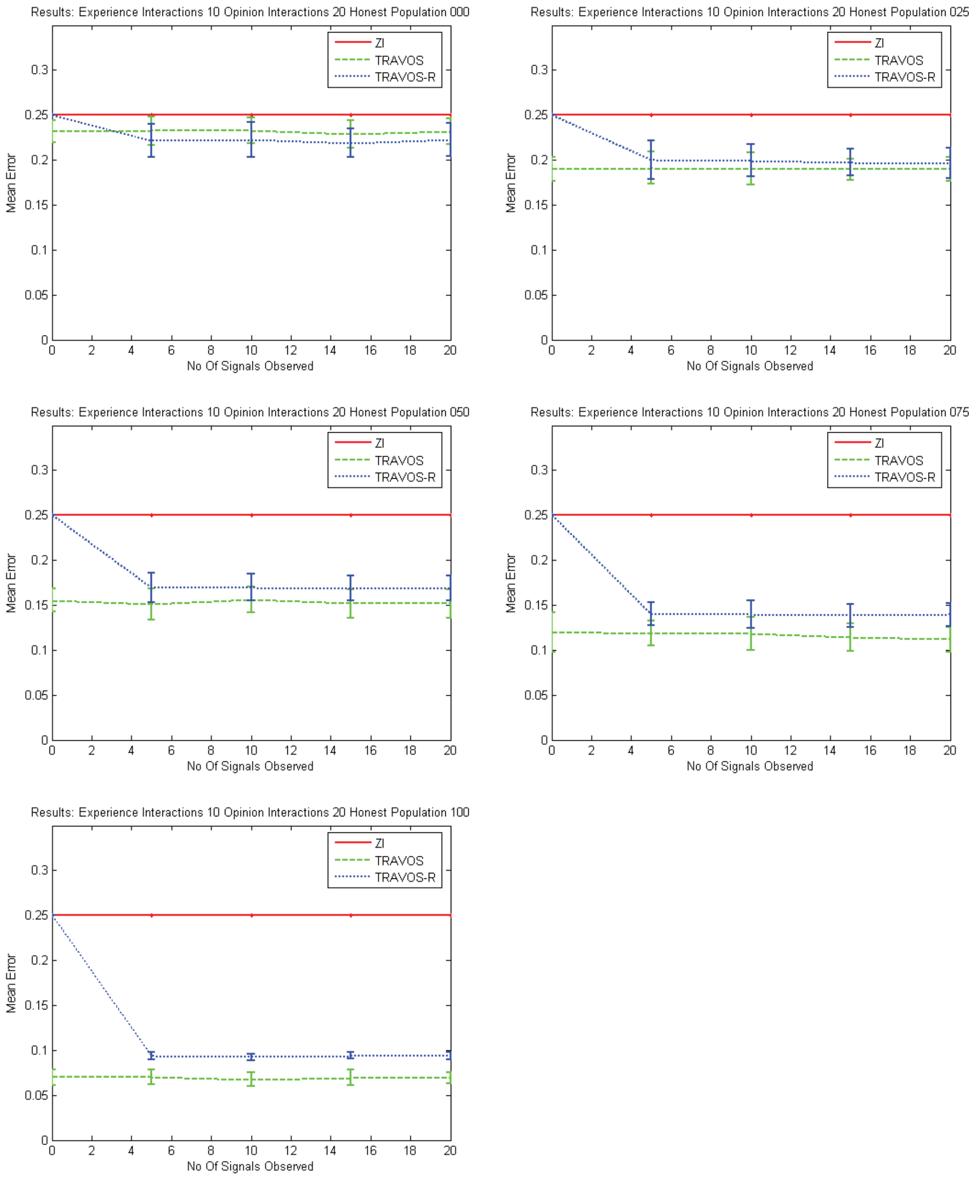

FIGURE B.6: Results of evaluating TRAVOS and TRAVOS-R in environments where the number of experience interactions is 10, the number of opinion experience interactions is 20, and the percentage of accurate (honest) opinion providers varies from 0% to 100%.

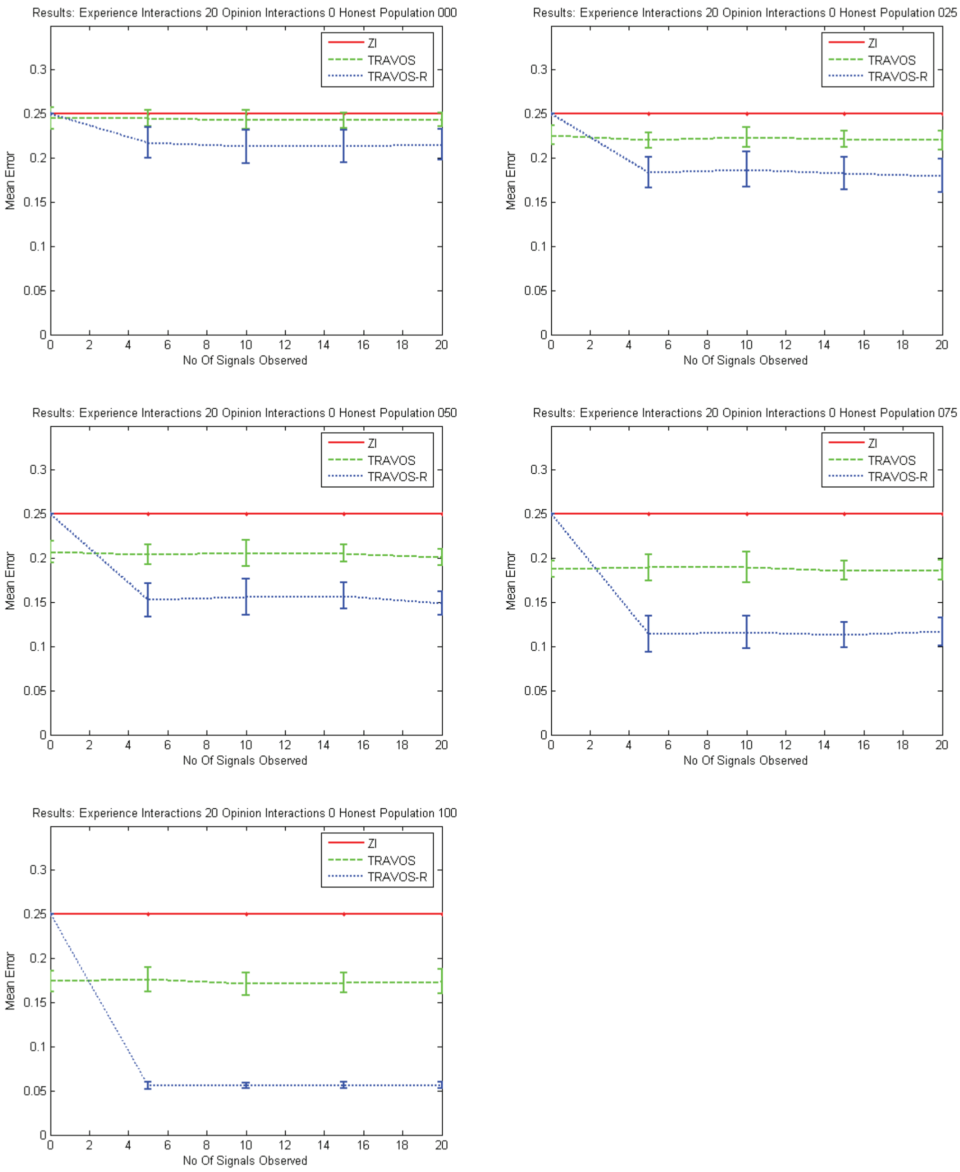

FIGURE B.7: Results of evaluating TRAVOS and TRAVOS-R in environments where the number of experience interactions is 20, the number of opinion experience interactions is 0, and the percentage of accurate (honest) opinion providers varies from 0% to 100%.

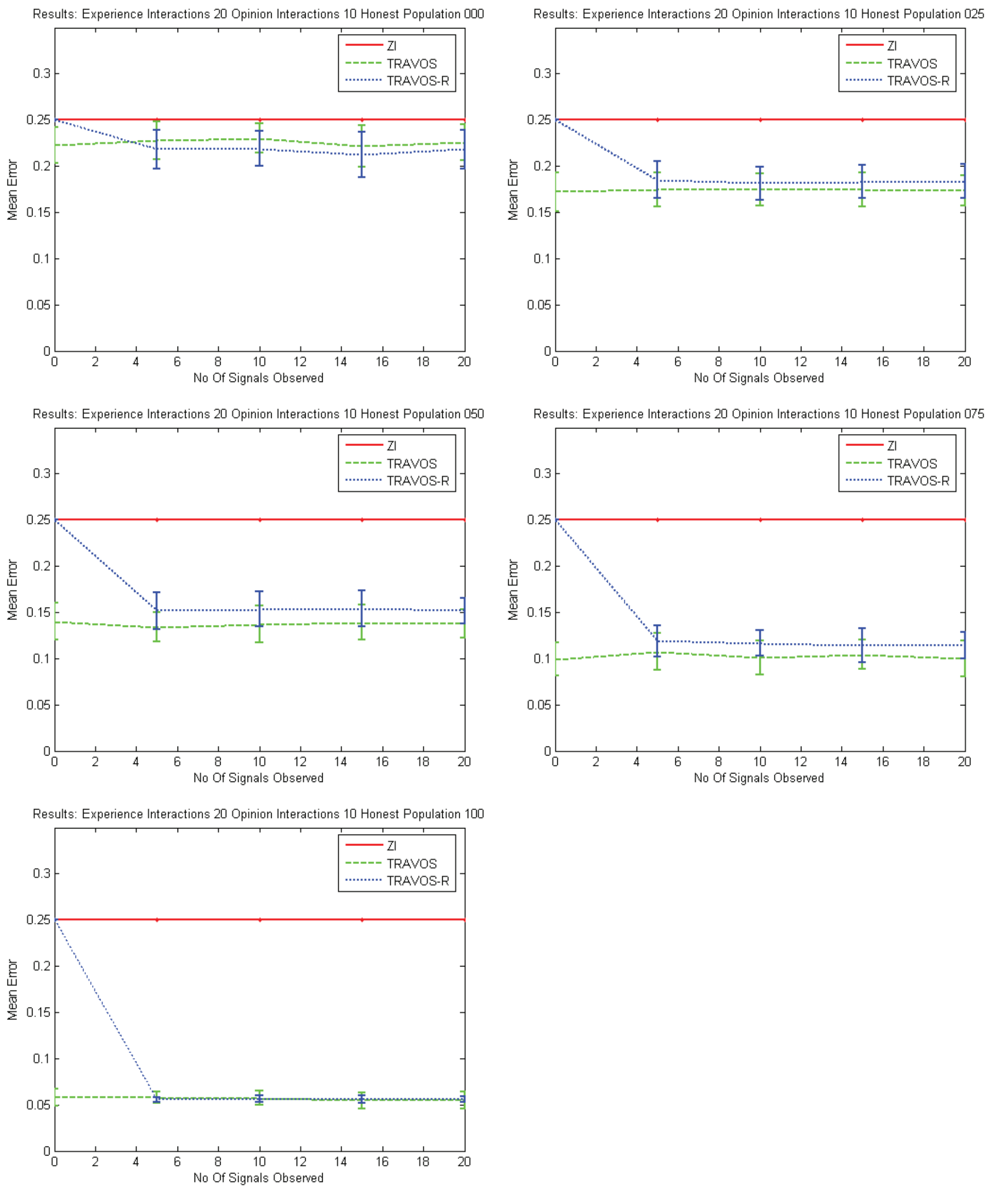

FIGURE B.8: Results of evaluating TRAVOS and TRAVOS-R in environments where the number of experience interactions is 20, the number of opinion experience interactions is 10, and the percentage of accurate (honest) opinion providers varies from 0% to 100%.

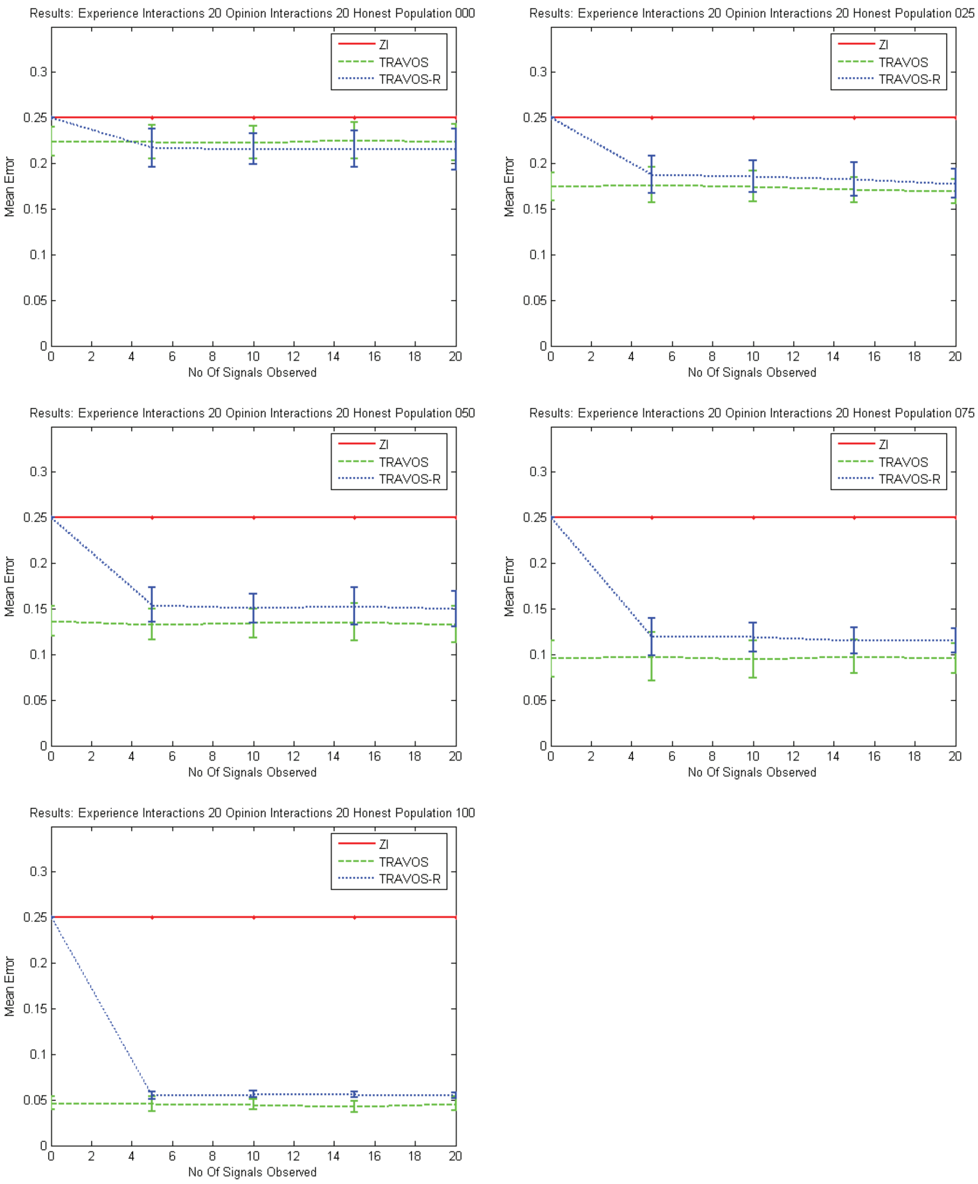

FIGURE B.9: Results of evaluating TRAVOS and TRAVOS-R in environments where the number of experience interactions is 20, the number of opinion experience interactions is 20, and the percentage of accurate (honest) opinion providers varies from 0% to 100%.

Bibliography

R. Ashri, M. Luck, and M. d'Inverno. On identifying and managing relationships in multi-agent systems. In *IJCAI*, pages 743–748. Morgan Kaufmann, 2003.

R. Ashri, S. D. Ramchurn, J. Sabater, M. Luck, and N. R. Jennings. Trust evaluation through relationship analysis. In *AAMAS '05: Proceedings of the fourth international joint conference on Autonomous agents and multiagent systems*, pages 1005–1011. ACM Press, 2005.

F. Baker. *Requirements for IP Version 4 Routers*. IETF, 1995.

B. Banerjee, A. Biswas, M. Mundhe, S. Debnath, and S. Sen. Using bayesian networks to model agent relationships. *Applied Artificial Intelligence*, 14(9):867–879, 2000.

F. Bellifemine, A. Poggi, and G. Rimassa. JADE: A FIPA2000 compliant agent development environment. In *AGENTS '01: Proceedings of the fifth international conference on Autonomous agents*, pages 216–217. ACM Press, 2001.

T. Berners-Lee, J. Hendler, and O. Lassila. The semantic web. *Scientific American*, 284(5): 34–43, 2001.

R. A. Brooks. Intelligence without representation. *Artificial Intelligence*, 47(1-3):139–159, 1991.

R. Butler, V. Welch, D. Engert, I. Foster, S. Tuecke, J. Volmer, and C. Kesselman. A national-scale authentication infrastructure. *Computer*, 33(12):60–66, 2000.

C. Castelfranchi and R. Falcone. Principles of trust for MAS: Cognitive anatomy, social importance, and quantification. In *ICMAS '98: Proceedings of the 3rd International Conference on Multi Agent Systems*, pages 72–79. IEEE Computer Society, 1998.

M. Chen and J. P. Singh. Computing and using reputations for internet ratings. In *EC '01: Proceedings of the 3rd ACM conference on Electronic Commerce*, pages 154–162. ACM Press, 2001.

P. R. Cohen. *Empirical Methods for Artificial Intelligence*. The MIT Press, 1995.

P. Dasgupta. Trust as a commodity. In D. Gambetta, editor, *Trust: Making and Breaking Cooperative Relations*, pages 49–72. Basil Blackwell, 1990.

M. DeGroot and M. Schervish. *Probability & Statistics*. Addison-Wesley, 2002.

C. Dellarocas. Mechanisms for coping with unfair ratings and discriminatory behavior in online reputation reporting systems. In *ICIS '00: Proceedings of the twenty first international conference on Information systems*, pages 520–525. Association for Information Systems, 2000.

D. DeRoure, M. Baker, N.R. Jennings, and N. Shadbolt. The evolution of the grid. In *Grid Computing: Making the Global Infrastructure a Reality*, pages 65–100. Wiley, 2003.

M. Esteva, J. A. Rodríguez-Aguilar, C. Sierra, P. Garcia, and J. Lluís Arcos. On the formal specifications of electronic institutions. In *Agent Mediated Electronic Commerce, The European AgentLink Perspective*, pages 126–147. Springer-Verlag, 2001.

R. Falcone and C. Castelfranchi. Social trust: A cognitive approach. pages 55–90, 2001.

FARNER. Farner: Factoring via network-enabled recursion. Published on World Wide Web: http://www.npac.syr.edu/factoring.html, 1995.

I. Foster, J. Geisler, B. Nickless, W. Smith, and S. Tuecke. Software infrastructure for the I-WAY high-performance distributed computing experiment. In *HPDC '96: Proceedings of the High Performance Distributed Computing*, pages 562–572. IEEE Computer Society, 1996.

I. Foster, N. R. Jennings, and C. Kesselman. Brain meets brawn: Why grid and agents need each other. In *AAMAS '04: Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems*, pages 8–15. IEEE Computer Society, 2004.

I. Foster and C. Kesselman. Globus: A metacomputing infrastructure toolkit. *The International Journal of Supercomputer Applications and High Performance Computing*, 11(2):115–128, 1997.

I. Foster and C. Kesselman, editors. *The Grid: Blueprint for a New Computing Infrastructure*. Morgan Kaufmann, 2nd edition, 2004.

I. Foster, C. Kesselman, G. Tsudik, and S. Tuecke. A security architecture for computational grids. In *CCS '98: Proceedings of the 5th ACM conference on Computer and communications security*, 1998.

I. Foster, C. Kesselman, and S. Tuecke. The anatomy of the Grid: Enabling scalable virtual organizations. *International Journal of Supercomputer Applications*, 15(3):200–222, 2001.

D. Gambetta. Can we trust trust? In D. Gambetta, editor, *Trust: Making and Breaking Cooperative Relations*, pages 213–237. Basil Blackwell, 1988.

D. K. Gode and S. Sunder. Allocative efficiency of markets with zero-intelligence traders: Market as a partial substitute for individual rationality. *Journal of Political Economy*, 101(1): 119–37, 1993.

N. Griffiths and M. Luck. Coalition formation through motivation and trust. In *AAMAS '03: Proceedings of the second international joint conference on Autonomous agents and multiagent systems*, pages 17–24. ACM Press, 2003.

D. Grossi, F. Dignum, L. M. M. Royakkers, and J. Ch. Meyer. Collective obligations and agents: Who gets the blame? In *DEON: Deontic Logic in Computer Science*, pages 129–145. Springer, 2004.

B. Horling and V. Lesser. A survey of multi-agent organizational paradigms. *Knowledge Engineering Review*, 19(4):281–316, 2004.

T. D. Huynh. *Trust and Reputation in Open Multi-Agent Systems*. PhD thesis, Electronics and Computer Science, University of Southampton, UK, 2006.

T. D. Huynh, N. R. Jennings, and N. R. Shadbolt. An integrated trust and reputation model for open multi-agent systems. *Autonomous Agents and Multi-Agent Systems*, 13(2):119–154, 2006.

R. Ismail and A. Jøsang. The beta reputation system. In *Proceedings of the 15th Bled Conference on Electronic Commerce*, Bled, Slovenia, 2002.

N. R. Jennings. Agent-Oriented Software Engineering. In F. J. Garijo and M. Boman, editors, *Proceedings of the 9th European Workshop on Modelling Autonomous Agents in a Multi-Agent World : Multi-Agent System Engineering (MAAMAW-99)*, pages 1–7. Springer-Verlag, 1999.

N. R. Jennings. An agent-based approach for building complex software systems. *Communications of the ACM*, 44(4):35–41, 2001.

N. R. Jennings and M. Wooldridge. Applications of intelligent agents. In *Agent technology: foundations, applications, and markets*, pages 3–28. Springer-Verlag New York, Inc., 1998.

A. Jøsang, R. Ismail, and C. Boyd. A survey of trust and reputation systems for online service provision. *Decision Support Systems*, to appear, 2006.

R. Jurca and B. Faltings. Towards incentive-compatiable reputation management. In R. Falcone, S. Barber, L. Korba, and M. Singh, editors, *Trust, Reputation and Security: Theories and Practice*, volume 2631 of *Lecture Notes in Artificial Intelligence*, pages 138–147. Springer, 2003.

R. Kalakota and M. Robison. *E-business: roadmap for success*. Addison-Wesley Longman Publishing Co., Inc., 1999.

Z. A. Karian and E. J. Dudewicz. *Fitting Statistical Distributions: The Generalized Lambda Distribution and Genreralized Bootstrap Methods*. CRC Press, 2000.

T. Li, H. Zhu, and K. Lam. A novel two-level trust model for grid. In S. Qing, D. Gollmann, and J. Zhou, editors, *ICICS 2003: Proceedings of the fifth international conference on Information and Communications Security*, volume 2836 of *Lecture Notes in Computer Science*, pages 214–225. Springer, 2003.

F. López López and M. Luck. A model of normative multi-agent systems and dynamic relationships. In *RASTA*, volume 2934 of *Lecture Notes in Computer Science*, pages 259–280. Springer, 2002.

M. Luck and M. d'Inverno. *Understanding Agent Systems*. Springer, 2nd edition, 2004.

M. Luck, P. McBurney, O. Shehory, and S. Willmott. *Agent Technology: Computing as Interaction A Roadmap for Agent-Based Computing*. Agentlink, 2005.

S. Marsh. *Formalising Trust as a Computational Concept*. PhD thesis, University of Stirling, UK, 1994.

D. McKnight and N. Chervany. The meanings of trust. Technical Report 94-04, Management Information Systems Research Center, University of Minnesota, 1996.

B. Misztal. *Trust in Modern Societies: The Search for the Bases of Social Order*. Polity Press, 1996.

A. Moukas, G. Zacharia, and P. Maes. Amalthaea and HISTOS: Multi-Agent Systems for WWW Sites and Reputation Recommendations. In M. Klusch, editor, *Intelligent Information Agents*, chapter 13. Springer-Verlag, 1999.

D. Nguyen, S. Thompson, J. Patel, W. T. L. Teacy, N. R. Jennings, M. Luck, S. Chalmers, N. Oren, T. Norman, A. Preece, P. M. D. Gray, G. Shercliff, P. J. Stockreisser, J. Shao, W. A. Gray, and N. J. Fiddian. Delivering services by building and running virtual organisations. *BT Technology Journal*, 24(1):141–152, 2006.

T. J. Norman, A. Preece, S. Chalmers, N. R. Jennings, M. Luck, V.D. Dang, T. D. Nguyen, V. Deora, , J. Shao, A. Gray, and N. J. Fiddian. CONOISE: Agent-based formation of virtual organisations. In *Proceedings of 23rd SGAI International Conference on Innovative Techniques and Applications of AI*, pages 353–366, 2003.

A. Oram and A. Oram. *Peer-to-Peer: Harnessing the Power of Disruptive Technologies*. O'Reilly & Associates, Inc., 2001.

J. Patel, W. T. L. Teacy, N. R. Jennings, and M. Luck. A probabilistic trust model for handling inaccurate reputation sources. In P. Herrmann, V. Issarny, and S.Shiu, editors, *iTrust'05: Proceedings of Third International Conference on Trust Management*, pages 193–209. Springer-Verlag, 2005a.

J. Patel, W. T. L. Teacy, N. R. Jennings, M. Luck, S. Chalmers, N. Oren, T. Norman, A. Preece, P. M. D. Gray, G. Shercliff, P. J. Stockreisser, J. Shao, W. A. Gray, N. J. Fiddian, and

S. Thompson. Agent-based virtual organisations for the grid. *International Journal of Multi-agent and Grid Systems*, 1(4):237–249, 2006.

J. Patel, W. T. L. Teacy, N. R. Jennings, M. Luck, S. Chalmers, N. Oren, T. Norman, A. Preece, P. M. D. Gray, G. Shercliff, P. J. Stockreisser, J. Shao, W. A. Gray, N. J. Fiddian, and S. Thompson. Agent-based virtual organisations for the grid. In *AAMAS '05: Proceedings of the fourth international joint conference on Autonomous agents and multiagent systems*, pages 1151–1152. ACM Press, 2005b.

J. Patel, W. T. L. Teacy, N. R. Jennings, M. Luck, S. Chalmers, N. Oren, T. Norman, A. Preece, P. M. D. Gray, G. Shercliff, P. J. Stockreisser, J. Shao, W. A. Gray, N. J. Fiddian, and S. Thompson. Monitoring, policing and trust for grid-based virtual organisations. In S. J. Cox and D. Walker, editors, *Proceedings of the UK e-Science All Hands Meeting 2005*, pages 891–898, 2005c.

I. Rahwan, S. D. Ramchurn, N. R. Jennings, P. Mcburney, S. Parsons, and L. Sonenberg. Argumentation-based negotiation. *Knowledge Engineering Review*, 18(4):343–375, 2003.

S. Ramchurn, D. Huynh, and N. R. Jennings. Trust in multiagent systems. *The Knowledge Engineering Review*, 19(1):1–25, 2004a.

S. D. Ramchurn. *Multi-Agent Negotiation using Trust and Persuasion*. PhD thesis, Electronics and Computer Science, University of Southampton, UK, 2004.

S. D. Ramchurn, N. R. Jennings, C. Sierra, and L. Godo. Devising a trust model for multi-agent interactions using confidence and reputation. *Applied Artificial Intelligence*, 18(9-10): 833–852, 2004b.

A. S. Rao and M. P. Georgeff. BDI-agents: From theory to practice. In V. Lesser, editor, *Proceedings of the First International Conference on Multiagent Systems*.

L. Rasmusson and S. Jansson. Simulated social control for secure internet commerce. In *NSPW '96: Proceedings of the 1996 workshop on New security paradigms*, pages 18–25. ACM Press, 1996.

M. R. Rodrigues and M. Luck. Analysing partner selection through exchange values. In J. Sichman and L. Antunes, editors, *Multi-Agent-Based Simulation VI*, volume 3891 of *Lecture Notes in Artificial Intelligence*, pages 24–40. Springer-Verlag, 2006.

S. Russell and P. Norvig. *Artificial Intelligence A Modern Approach*. Prentice Hall, 2nd edition, 2003.

J. Sabater and C. Sierra. REGRET: reputation in gregarious societies. In *AGENTS '01: Proceedings of the fifth international conference on Autonomous agents*, pages 194–195. ACM Press, 2001.

J. Sabater and C. Sierra. Social regret, a reputation model based on social relations. *SIGecom Exchanges*, *ACM*, pages 44–56, 2002. 3.1.

H. Schmeck, T. Ungerer, and L. Wolf, editors. *Trends in Network and Pervasive Computing - ARCS 2002*, volume 2299, 2002. Springer-Verlag: Heidelberg, Germany.

G. Shafer. *A Mathematical Theory of Evidence*. Princeton University Press, 1976.

J. Shao, W. A. Gray, N. J. Fiddian, V. Deora, G. Shercliff, P. J. Stockreisser, T. Norman, A. Preece, P. M. D. Gray, S. Chalmers, N. Oren, N. R. Jennings, M. Luck, V. D. Dang, T. D. Nguyen, J. Patel, and W. T. L. Teacy. Supporting formation and operation of virtual organisations in a grid environment. In S. J. Cox, editor, *Proceedings of the UK e-Science All Hands Meeting 2004*, pages 376–383, 2004.

M. P. Singh. Agent communication languages: Rethinking the principles. *Computer*, 31(12): 40–47, 1998.

D. C. Smith, A. Cypher, and J. Spohrer. Kidsim: programming agents without a programming language. *Communication of the ACM*, 37(7):54–67, 1994.

W. T. L. Teacy, J. Patel, N. R. Jennings, and M. Luck. TRAVOS: Trust and reputation in the context of inaccurate information sources. *International Journal of Autonomous Agents and Multi-Agent Systems*, 12(2):183–198, 2006.

W. T. L. Teacy, Jigar Patel, Nicholas R. Jennings, and Michael Luck. Coping with inaccurate reputation sources: experimental analysis of a probabilistic trust model. In *AAMAS '05: Proceedings of the fourth international joint conference on Autonomous agents and multiagent systems*, pages 997–1004. ACM Press, 2005.

I. Trencansky and R. Cervenka. Agent modeling language (AML): A comprehensive approach to modeling mas. 29(4):391–400, 2005.

G. Weiss. *Multi-Agent Systems*. The MIT Press, 1999.

A. Whitby, A. Jøsang, and J. Indulska. Filtering out unfair ratings in bayesian reputation systems. *The Icfain Journal of Management Research*, 4(2):48–64, 2005.

B. Williams, P. Bateson, D. Good, P. Dasgupta, J. Dunn, N. Luhmann, G. Hawthorn, A. Pagden, E. Gellner, K. Hart, E. Lorenz, and D. Gambetta. *Trust: Making and Breaking Cooperative Relations*. Basil Blackwell, 1988.

M. J. Wooldridge. *Introduction to Multiagent Systems*. John Wiley & Sons, Inc., 2001.

M. J. Wooldridge and N. R. Jennings. Intelligent Agents: Theory and Practice. *Knowledge Engineering Review*, 10(2):115–152, 1995.

B. Yu and M. P. Singh. An evidential model of distributed reputation management. In *AAMAS '02: Proceedings of the first international joint conference on Autonomous agents and multiagent systems*, pages 294–301. ACM Press, 2002.

B. Yu and M. P. Singh. Searching social networks. In *AAMAS '03: Proceedings of the second international joint conference on Autonomous agents and multiagent systems*, pages 65–72. ACM Press, 2003.