Monitoring, Policing and Trust for Grid-Based Virtual Organisations

Jigar Patel, W. T. Luke Teacy, Nicholas R. Jennings, Michael Luck
School of Electronics and Computer Science, University of Southampton, UK

Stuart Chalmers, Nir Oren, Timothy J. Norman, Alun Preece, Peter M. D. Gray
Department of Computing Science, University of Aberdeen, UK

Gareth Shercliff, Patrick J. Stockreisser, Jianhua Shao, W. Alex Gray, Nick J. Fiddian
School of Computer Science, Cardiff University, UK

Simon Thompson
BT R&D, Adastral Park, UK

Abstract

A key challenge in Grid Computing is the ability to create reliable and scalable virtual organisations (VOs) which operate in an open, dynamic and competitive environment. In response, in the CONOISE-G project, we are developing an infrastructure to support robust and resilient virtual organisation formation and operation. Specifically, CONOISE-G provides mechanisms to assure effective operation of agent-based VOs in the face of disruptive and potentially malicious entities in dynamic, open and competitive environments. In this paper, we describe the architecture of the CONOISE-G system, and provide details of its implementation.

1 Introduction

The engineering of systems that establish a fixed organisational structure is not sufficient to handle many of the issues inherent in open multi-agent systems (in particular, heterogeneity of agents, trust and accountability, failure handling and recovery, and societal change [8, 11]). Such issues are becoming increasingly important in the context of Grid computing, which aims to enable resource sharing and coordinated problem-solving in dynamic, multi-institutional virtual organisations (VOs) [8].

VOs are composed of a number of autonomous entities (representing different individuals, departments and organisations), each of which has a range of problem-solving capabilities and resources at its disposal. While such entities are typically self-interested, there are sometimes potential benefits to be obtained from pooling resources: either with a competitor (to form a coalition) or with an entity with complementary expertise (to offer a new type of service). The recognition of this potential can be the cue for the formation of a VO in which distinct, autonomous entities come together to exploit a perceived niche. When this is successful, the collection of independent entities acts as a single conceptual unit in the context of the proposed service, requiring that the participants cooperate and coordinate their activities in delivering the services of this newly formed organisation. Part of this demands that the participants have the ability to manage the VO effectively. In dynamic environments, however, the context may change at any time, so that the VO may no longer be viable. It must then either disband or rearrange itself to better fit the circumstances. This paper describes technologies developed to address both these phases.

VOs thus provide a way of abstracting the complexity of open systems to make them amenable to application development. The organisational structure, participant responsibilities, synchronisation concerns and economic mechanics of the VO are hidden from the VO user. This has two benefits: first, agents can be used to bridge between requester and providers to organise the VO and to provide a layer of flexibility between requesting applications and the underlying service infrastructure; second, the VO fulfils the role of information hiding in that the internal mechanics are abstracted away from the requesting application, and the VO formation and management system either supports a request or fails at well-defined points.

While the notion of VOs underpins the vision of Grid computing, the conditions under which a new VO should be formed, and the procedures for its formation, operation and dissolution, are not well-defined. This automated formation and ongoing management of VOs in open environments thus constitutes a major research challenge, a key objective of which is to ensure that they are both agile (can adapt to changing circumstances) and resilient (can achieve their aims in a dynamic and uncertain environment). In addition to traditional constraints that relate to issues such as resource management and bidding strategies, we must also consider softer constraints relating to contract management, trust between VO participants and policing of contracts.

The CONOISE-G project (Grid-enabled Constraint-Oriented Negotiation in an Open Information Services Environment, *http://www.conoise.org*) is directed at addressing just these issues. It seeks to support robust and resilient VO formation and operation, and aims to provide mechanisms to assure effective operation of agent-based VOs in the face of disruptive and potentially malicious entities in dynamic, open and competitive environments. In this paper, we describe the CONOISE-G system, in which VO formation is grounded on three key technologies [13]: the agent decision-making, auctions for allocation of contracts, and service discovery incorporating quality of service (QoS) assessment.

In addition, however, to operate an effective VO in open, dynamic and competitive environments, it is essential that we also consider how to encourage good interactions, and cope effectively with bad ones. In our view, this requires that QoS levels are monitored, that uncertainty in participant behaviour, possibly arising from participant self-interest and strategic lying and collusion, is minimised, and that mechanisms for recognising and addressing contract violations once they have occured are established. Addressing these concerns is integral to the wide-scale acceptance of the Grid and agent-based VOs.

The contribution of this paper lies in the construction of an implemented prototype for dynamic re-formation of VOs through the integration of several different techniques. The paper begins with a motivating example that introduces the need for VO formation and operation. It then describes the system architecture, elaborating the different aspects identified above in support of robust and resilient operation. The paper ends with a description of the implemented prototype that underlies the core of the current work in the CONOISE-G project to achieve effective VO formation and operation within a Grid environment.

2 A Motivating Scenario

As in [17] the motivating scenario is as follows. Lucy visits the 2012 Olympic Games and using her PDA, accesses multimedia services such as news, clips from the Games, and ticket purchase facilities. Many service providers offer such services, so Lucy must determine potential providers, select an optimal package, and then track the changing market for better deals. In such situations, creating a VO on demand can greatly simplify the problem, allowing users merely to specify their service requirements, with VOs providing the required services. However, forming and operating a VO is complex. By way of example, suppose there are five service providers (SP1, . . . , SP5), as in Table 1, each offering relevant multimedia services. These services form three groups: *video content* (Entertainment and Game Clips services), *HTML content* (News and Ticketing services) and *text messaging* (Text service); and they can be requested individually or taken as a package, with the constraint that the two services offered by SP2 must be taken together.

SP	Ent	News	Text	Games	Tkts
SP1	30	20			5
SP2		10	50		
SP3			100	30	5
SP4	30	10		60	
SP5			50	45	10

Table 1: Potential Service Providers

Service Required	Units Required
Entertainment	50 mins per month
News	10 updates per day
Text messages	100 per month
Game Clips	60 mins per day
Ticketing	10 alerts per day

Table 2: Example service package request

We assume that these providers may demand different prices for the same service, depending on the number of units requested. For example, SP1 may offer 20 news updates per day at £30 per month, and 10 updates at £25 per month. Also, the quality of services may not be stable: SP4 may offer Games clips with a frame rate of no less than 24 frames per second, but actually provide a rate that drops below that level. Finally, not all service providers are trustworthy, and what they claim may not be what a requester will get: SP5 may advertise sought-after tickets that it does not possess, and orders for tickets through SP5 may not always be honoured.

Now, suppose that Lucy wishes to purchase the service package of Table 2. It should be clear from Table 1 that many different solutions are possible. For example, for 50 minutes of entertainment, both SP1 and SP4 must be used, but different compositions of the two services are possible, with different price, quality and degree of trust. To find a good solution for a given service request, therefore, several issues must be addressed. During VO formation, multiple service providers may offer broadly similar services, each described by multiple attributes including, for example, price, quality, reputation and delivery time. We therefore need to determine how the relevant services for a given service request may be discovered, and how an optimal package may be selected, based on the above attributes. During VO operation, however, the services available may change over time: new services may become available, or providers may alter the way in which existing services are offered. Quality of service and provider reputation may also change over time. There is thus a need to monitor the performance of the members of a VO in terms of their trustworthiness, quality of service and conformance to contract, and to restructure the VO when necessary so that the integrity and usefulness of the VO are maintained. Thus, a poorly performing service may be replaced, a contract-breaking service may be dropped, and a new user requirement may be accommodated.

Creating and then effectively managing a VO in a dynamic environment thus poses significant research challenges. In seeking to address them, we have developed a system for dynamic formation and operation of VOs. In the following sections, we outline the system architecture and describe its key components.

3 Architecture

In essence, the CONOISE-G architecture comprises several different agents, including *system agents* and *service providers* (SPs), as shown in Figure 1. The former are those needed to achieve core system functionality for VO formation and operation, while the latter are those involved in the VO itself. For simplicity, we omit some specific components that perform basic functions, such as a Yellow Pages (YP) agent, since they add little to the issues to be discussed.

Assuming that service providers have already advertised their services to a YP, the VO formation process starts with a particular SP acting on behalf of a user, the Requester Agent (RA), which analyses the requester's service requirements, locates the relevant providers through the YP, and then invites the identified providers to bid for the requested services. The quality and trustworthiness of the received bids are assessed by the Quality Agent (QA) and the trust component, respectively, and the outcome is combined with the price structure by a Clearing Agent (CA) [13] to determine which combination of the services/providers will form an optimal VO (in terms of price, quality and trust) for the requester. At this point, the VO is formed and the RA takes on the role of VO Manager (VOM), responsible for ensuring that each member provides its service according to contract.

During the operational phase of the VO, the VOM may request the QoS Consultant (QoSC) to monitor any services provided by any members of the VO, and any member of the VO may invoke the Policing agent to investigate any potential dispute regarding service provision. Ultimately, our aim is for monitoring to take place to inform the user when the actual service level diverges from the agreed service level. At present, however, this is achieved by configuring the levels of QoS for each service that will cause the QoSC to alert the VOM, using predetermined service provision and quality level simulations. When the QoS provision of a service (say the *news* service in the scenario) in the VO falls below an acceptable level of service, or some breach of contract is observed, the QoSC alerts the VOM, which initiates a VO re-formation process; relevant information is fed into the trust component to ensure that the provider concerned is penalised to an appropriate level by updating its record of trust.

In this re-formation process, the VOM issues another message to the YP requesting a list of SPs that can provide the *news* service. As before, the YP identifies possible SPs, bids are received and evaluated, resulting in the CA determining the best SP to replace the failed provider. At this point, the VOM re-forms the VO with the new SP replacing the old one, and instructs the QoSC to stop monitoring the old SP and to monitor the new one instead. In the following sections, we discuss the core technical components of operation and re-formation processes in more detail. Further details of the formation process can be found in [17].

Trust and Reputation

It is now well established that computational *trust* is important in such open systems [16]. Specifically, trust provides a form of social control in environments in which agents are likely to interact with others whose intentions are not known. It allows agents within such systems to reason about the reliability of others. More specifically, trust can be utilised to account for uncertainty about the willingness and capability of other agents to perform actions as agreed, rather than defecting when it proves to be more profitable. In this work, we adapt Gambetta's definition [9], and define trust to be *a particular level of subjective probability with which an agent assesses that another agent will perform a particular action, both before the assessing agent can monitor such an action and in a context in which it affects the assessing agent's own action.*

In CONOISE-G, trust is often built over time by accumulating personal experience with others; we use this experience to judge how they will perform in an as yet unobserved situation. However, when assessing our trust in someone with whom we have no direct personal experience, we often ask others about their experiences with this individual. This collective opinion of

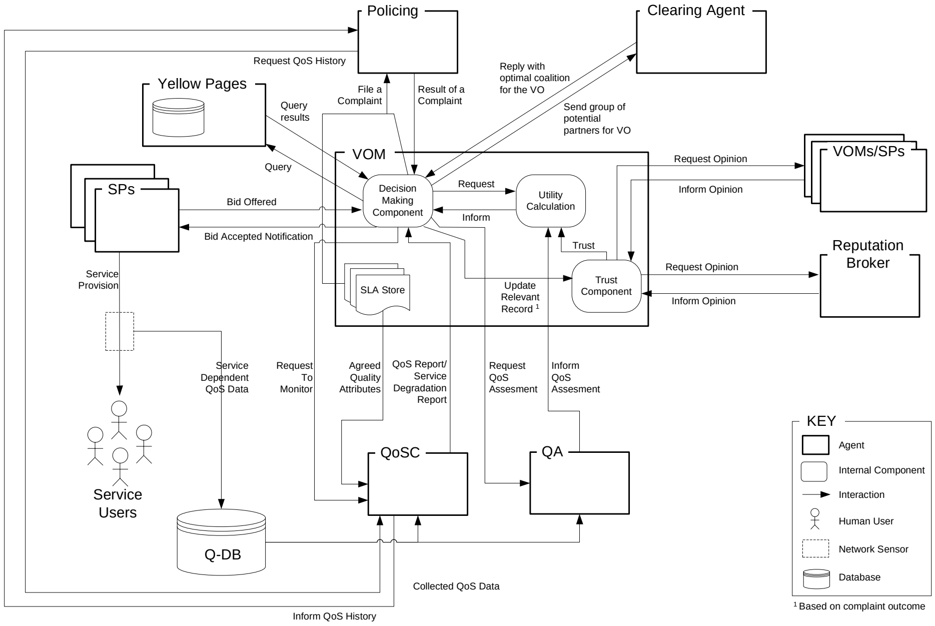

Figure 1: The CONOISE-G system architecture

others regarding an individual is known as the individual's *reputation*, which we use to assess its trustworthiness, if we have no personal experience.

Given the importance of trust and reputation in open systems and their use as a form of social control, several computational models of trust and reputation have been developed, each with requirements for the domain to which they apply (see [16] for a review of such models). In our case, the requirements can be summarised as follows. First, the model must provide a trust metric that represents a level of trust in an agent. Such a metric allows comparisons between agents so that one agent can be inferred as more trustworthy than another. The model must be able to provide a trust metric given the presence or absence of personal experience. Second, the model must reflect an individual's *confidence* in its level of trust for another agent. This is necessary so that an agent can determine the degree of influence the trust metric has on its decision about whether or not to interact with another individual. Generally speaking, higher confidence means a greater impact on the decision-making process, and lower confidence means less impact. Third, an agent must not assume that the opinions of others are accurate or based on actual experience. Thus, the model must be able to discount the opinions of others in the calculation of reputation, based on past reliability and consistency of the opinion providers. However, generally speaking, existing models do not allow an agent to effectively assess the reliability of an opinion source and use this assessment to discount the opinion provided by that source. To meet the above requirements, we have developed TRAVOS [18], a trust and reputation model for agent-based VOs.

The first and the second requirements have been dealt with in earlier work [17]. The novel contribution of TRAVOS is in the way it addresses the third. Inaccurate reputation reports can be due to opinion providers being malevolent or having incomplete information. In both cases, an agent must be able to assess the reliability of the reports passed to it. The general solution to coping with inaccurate reputation reports is to adjust or ignore opinions judged to be unreliable (in order to reduce their effect on the trustee's reputation). There are two basic approaches to achieving this that have been proposed in the literature; Jøsang et al. [19] refer to these as *endogenous* and *exogenous* methods. The former attempt to identify unreliable reputation information by considering the statistical properties of the reported opinions alone (e.g. [19, 6]). The latter rely on other information to make such judgements, such as the reputation of the source or the relationship with the trustee (e.g. [3, 20]).

Many proposals for endogenous techniques assume that inaccurate or unfair raters will generally be in a minority among reputation sources. Based on this assumption, they consider reputation providers whose opinions deviate in some way from mainstream opinion to be those most likely to be inaccurate. Our solution is exogenous, in that we judge a reputation provider on the perceived accuracy of its past opinions, rather than its deviation from mainstream opinion. Moreover, we define a two step-method: First, we calculate the probability that an agent will provide an accurate opinion given its past opinions and later observed interactions with the trustees for which opinions were given. Second, based on this value, we reduce the distance between a rater's opinion and the prior belief that all possible values for an agent's behaviour are equally probable. Once all the opinions collected about a trustee have been adjusted in this way, the opinions are aggregated. This is done such that, if all rater's opinions are considered accurate, the resulting trust value is the same as if the truster had *directly* experienced all the previous interactions of the raters with the trustee. In addition, however, any rater whose opinion is not considered completely reliable will have their affect on the overall evaluation of the trustee reduced. In the extreme case, an opinion which is judge to have a 0 probability of accuracy will have no affect on the overall evaluation.

The implementation of TRAVOS in CONOISE-G has three main parts. First, there is a trust component inside each agent that allows it to calculate trust and confidence values based on its own experiences. This component is also responsible for aggregating the opinions (by the process described above) provided by others when an agent does not have sufficient personal experience. Second, we have well defined trust ontologies that form the basis of the trust-related communication between agents in order to exchange trust and reputation information. Third, to address the problem of scalability we have implemented *reputation broker agents*. When an agent wishes to obtain reputation information in CONOISE-G it can make use of these brokering agents, of which several may exist in the system, serving as a distributed store of reputation information.

A reputation broker provides an aggregated store of trust information relating to specific service provider agents and each of their services. However, before any agent can query the broker, the broker must obtain the trust information that will form the query result. We achieve this using a *subscribe and publish* mechanism, by which the broker subscribes to agents in the community which then publish their internal information (the store of outcomes based on their individual direct experiences) to the broker. Agents in the community can obtain reputation information from these brokers by sending query messages, to which the brokers can reply with the relevant information or a failure message in the case where they do not have such information. When an agent does receive reputation information from a broker, it assesses the accuracy of this information, just as it would if the information was sourced from an individual reputation provider.

Policing within a VO

While trust and reputation ratings are able to reduce the likelihood of poorly performing (or malicious) agents becoming part of a VO, they do not offer any mechanism for minimising the impact of undesirable behaviour, such as an agent contracting to provide services it does not deliver. The policing system determines whether a party is in breach of a contract, determines if any corrective action (as stipulated in the contract) should be taken, and informs the trust mechanism of the result. Given the scalability concerns inherent in large, open distributed systems, the CONOISE-G system responds to reported exceptional circumstances, rather than monitoring operation.

The policing system initiates an investigation following the receipt of a complaint from a VO member. The process begins by obtaining the contract at the centre of the dispute, and gathering evidence to determine the actual state of affairs. This can take on a number of forms, including reports from agents in the system and other artifacts; it is recursive, in that one piece of evidence may have further evidence supporting or rebutting it. Furthermore, agents can submit evidence in support of or against a conclusion. The evidence gathered, therefore, constitutes a set of defeasible arguments in support of and in defence of the complaint. Our approach borrows from computational models of legal reasoning and argumentation [2].

We thus view the policing system as consisting of a number of distinct components, contained in both the environment infrastructure and the individual agents: a component able to describe ideal system behaviour (requiring a contracting language and a set of contract instances); an interface to allow agents to provide arguments and evidence to the system, as well as a method to allow the system to request further information; a reasoning mechanism to determine the evidence to be gathered; and a technique for weighing up evidence, without which policing agents cannot combine arguments to reach a verdict.

In CONOISE-G the representation of contracts is based on the emerging Web Services standard for agreements, WS-Agreement [4]. We extend this language to represent concepts such as prohibited activities, transferable responsibilities and group actions that do not appear in the existing standard. We are also investigating methods for grounding the semantics of such contracts to bring these pragmatic approaches closer to formal contract specification languages such as those developed by Dignum et al. [7] and Pacheco and Carmo [14]. The evidence gathering mechanism employed is tightly coupled with the reasoning machinery; both activities are driven by sets of defeasible arguments. Agents involved in the contract may submit evidence to the policing agent, which can ask questions, obtain logs, etc., according to the rules of a dialogue game developed for the purpose of evidence gathering. Strategies for determining what evidence should be submitted or sought, as well as reasoning about how arguments and evidence interact and combine are being used to facilitate reasoning about contract failure. Little related work on reasoning about contract failure exist, although the work of Daskalopulu et al. [5], in which Dempster-Schafer theory is employed, is one of the exceptions. However, a number of scientists have investigated methods for combining evidence in general, and the use of techniques from argumentation theory in particular [15, 12], and it is this work that complements the policing model being developed in CONOISE-G.

Monitoring QoS Levels

During the operation of a VO, it is important that the QoS provision is monitored. The QoS data collected from this monitoring process is vital in supporting the creation of a resilient VO. First, it serves as "evidence" in a range of critical assessment. For example, the QoS data is used: by the Trust component to establish the level of trust that can be placed in a service and service provider; by the Policing agent to deal with complaints; and by the QA to assess QoS for services during future VO formations. Second, the QoS data helps monitor and predict any QoS degradation within a VO. Any detection or prediction of such degradation can result in a possible replacement of a VO member, or trigger reformation of the VO, ensuring that the VO maintains an agreed level of QoS provision, limiting any damage to its reputation.

In the CONOISE-G system, the monitoring of QoS provision is carried out by the QoSC, which is designed to perform three main tasks. The first entails the recording and gathering of QoS data, a continuous activity that contributes to a QoS database. Here, data collection is performed through the use of network sensors and, for simplicity, we also assume that the QoS at any point on the link from the provider to the consumer is the same. The second task involves the monitoring of the QoS level. The current level of service provision is calculated from the data that is collected from the network sensors and compared to the QoS level stated in the service level agreement. Any service whose QoS has dropped below the level required is then reported to the VOM concerned. Since QoS data can be generated continuously at a very fast speed, and needs to be processed with respect to dynamic, ad-hoc monitoring requests from individual VOMs, we adopt a data stream [1] approach to QoS monitoring in constructing the QoSC agent. The third task to be performed by the QoSC is that of alerting the VOM to any anticipated drop in QoS. Taken together, these tasks provide a versatile, accurate and robust QoS monitoring mechanism.

4 Implementation

The CONOISE-G environment is FIPA[1] compliant and the implementation uses the JADE[2] agent platform. Agents communicate by exchanging FIPA ACL (agent communication language) messages, the content of which is defined using lightweight ontologies expressed in Semantic Web (SW) representations, following experience from previous work [10]. We chose these representations in preference to the more conventional use of FIPA-SL in the content of FIPA messages for a number of reasons. First, the SW representations are more widely used than FIPA-SL, so CONOISE-G is lent greater interoperability by aligning with W3C recommendations. Second, we can reuse existing schemas and ontologies; for example, we borrowed heavily from the DAML-S service ontology. Thus, we would be in a position to exploit any existing schemas or ontologies in a particular application domain. Third, particularly at the lower (RDF) layers of the SW formalism stack, the semantics of the data model are much simpler than FIPA-SL (while still adequate for operational use), so there is less of a learning curve for designers and implementors of CONOISE-G agents

In the current system, we have created a set of interrelated ontologies expressed in a relatively lightweight manner as RDF schemas. For now, RDFS is sufficiently expressive to capture usable structures, and has allowed us to rapidly develop the necessary message formats for inter-agent communication in our scenario. We envisage the definitions in the ontologies being refined with the addition of OWL (Web Ontology Language) statements once the formats have stabilised through further testing and refinement. Two sample RDF messages expressed using a number of the ontologies are shown in Figures 2 and 3. The first shows a sample call for bids, as issued to SPs. This consists of an instance of a user `Requirement` structure, stating a number of services that the user's requirement `consistsOf`, and also a `qualityPreference` property, indicating that the most important thing for this user is lowest cost. The descriptions of each required service are adorned with service-specific properties; for example, the `MovieContent` requirement specifies a number of movies (per month), a subscription preference, and a genre type. This illustrates the use of terms from three CONOISE-G ontologies:

- the `package` ontology describes service packages, defining terms such as the class

[1] `http://www.fipa.org`

[2] `http://sharon.cselt.it/projects/jade`

```
<package:Requirement rdf:about=‘‘http://conoise.org/samples/request’’>
  <quality:qualityPreference rdf:resource=
        ‘‘http://conoise.org/ontologies/quality#minCost’’/>
  <package:consistsOf
    rdf:type=‘‘http://conoise.org/ontologies/media#PhoneCalls’’
    media:numberOfMinutes=‘‘25’’/>
  <package:consistsOf>
    <media:MovieContent media:numberOfMovies=‘‘72’’>
      <media:subscriptionType rdf:resource=
          ‘‘http://conoise.org/ontologies/media#monthly’’/>
      <media:mediaStyle rdf:resource=
          ‘‘http://conoise.org/ontologies/media#scienceFiction’’/>
    </media:MovieContent>
  </package:consistsOf>
  <package:consistsOf>
    <media:HtmlContent media:updateFrequency=‘‘24’’>
      <media:mediaStyle rdf:resource=‘‘http://conoise.org/ontologies/media#news’’/>
    </media:HtmlContent>
  </package:consistsOf>
  <package:consistsOf
    rdf:type=‘‘http://conoise.org/ontologies/media#TextMessaging’’
    media:numberOfMessages=‘‘100’’/>
</package:Requirement>
```

Figure 2: RDF call for bids sent to SPs

`Requirement` and the property `consistsOf`;

- the `quality` ontology describes domain-independent quality-of-service terms such as the `qualityPreference` property, and its various settings such as “minCost”;

- the `media` ontology defines all application domain-specific terms for the Olympics scenario, including the service classes `MovieContent`, `HtmlContent`, `PhoneCalls`, and `TextMessaging`, all of which the ontology defines to be (indirect) sub-classes of the generic CONOISE `ServiceProfile` class (closely based on DAML-S).

The second sample message, in Figure 3, shows a bid issued by one of the SPs in response to the call shown in Figure 2. The bid is for just one of the required services (the `HtmlContent` part); the `Bid` structure is similar to the `Requirement` structure in that it also employs the `consistsOf` property, but here there is also an identified instance of a `Provider`, whose properties are defined using terms from the `profile` ontology (that also defines the `ServiceProfile` class mentioned above). This information allows the user to access the service if the bid is ultimately accepted as part of the winning package. Note also that the services offered in bids have `Price` structures attached, which are rich enough to identify different price *bands* depending on the volume the user might wish to consume.

These examples illustrate how the capability to create modular, interlocking ontologies using the SW formalisms allow us to build up quite elaborate information representations, all of which are easily serialisable in a portable, open XML syntax, and easily parsed and processed using tools such as Jena2[3].

5 Conclusions

The work described in this paper takes an approach in which issues relating to the formation and operation of robust VOs in the dynamic environments with unreliable agents are considered. In contrast to the “brawn” of the Grid, we have concentrated on the “brains” [8] — on the development of techniques for autonomous problem-solving in VOs. Thus, we have described an agent architecture for re-forming VOs in the face of unreliable information, through the use of a range of techniques that support robust and resilient VO formation and operation for application to realistic electronic commerce scenarios. We described our implemented prototype of the system, and elaborated the work being done on extending the system to incorporate more sophisticated application scenarios.

Acknowledgements

CONOISE-G is funded by the DTI and EPSRC through the Welsh e-Science Centre, in collaboration with the

[3] `http://www.hpl.hp.com/semweb/jena2.htm`

```
<package:Bid rdf:about=‘‘http://conoise.org/samples/pa2bid#bid2’’>
  <package:providedBy>
    <package:Provider
      profile:fipaAddress=‘‘pa2@conoise.org:15551/JADE’’>
      <profile:name>Provider Agent 2</profile:name>
    </package:Provider>
  </package:providedBy>
  <package:consistsOf
    <media:HtmlContent rdf:about=‘‘http://conoise.org/samples/pa2bid#pa2news’’
      media:updateFrequency=‘‘72’’>
      <media:mediaStyle rdf:resource=‘‘http://conoise.org/ontologies/media#news’’/>
      <package:hasPriceStructure
        rdf:type=‘‘http://conoise.org/ontologies/package#Price’’
        package:min=‘‘0’’ package:max=‘‘10’’ package:unitPrice=‘‘3’’/>
      <package:hasPriceStructure
        rdf:type=‘‘http://conoise.org/ontologies/package#Price’’
        package:min=‘‘10’’ package:max=‘‘50’’ package:unitPrice=‘‘2’’/>
      <package:hasPriceStructure
        rdf:type=‘‘http://conoise.org/ontologies/package#Price’’
        package:min=‘‘50’’ package:max=‘‘1000’’ package:unitPrice=‘‘1’’/>
    </media:HtmlContent>
  </package:consistsOf>
</package:Bid>
```

Figure 3: RDF bid issued by SP

Office of the Chief Technologist of BT.

References

[1] B. Babcock, S. Babu, M. Datar, R. Motwani, and J. Widom. Models and issues in data stream systems. In *Proc. 21st ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, pages 1–16. ACM Press, 2002.

[2] T. Bench-Capon, J. B. Freeman, H. Hohmann, and H. Prakken. Computational models, argumentation theories and legal practice. In *Argumentation Machines: New Frontiers in Argument and Computation*, pages 85–120. Kluwer, 2003.

[3] S. Buchegger and J. Y. Le Boudec. A robust reputation system for mobile ad-hoc networks ic/2003/50. Technical report, EPFL-IC-LCA, 2003.

[4] K. Czajkowski, A. Dan, J. Rofrano, S. Tuecke, and M. Xu. WS-Agreement: Agreement–based grid service management. In *Global Grid Forum*, 2003.

[5] A. Daskalopulu, T. Dimitrakos, and T. Maibaum. Evidence-based electronic contract performance monitoring. *Group Decision and Negotiation*, 11(6):469–485, 2002.

[6] Chrysanthos Dellarocas. Mechanisms for coping with unfair ratings and discriminatory behavior in online reputation reporting systems. In *Proc. 21st International Conference on Information Systems (ICIS)*, pages 520–525, Brisbane, Australia, December 2000.

[7] V. Dignum, J.-J. Meyer, F. Dignum, and H. Weigand. Formal specification of interaction in agent societies. In *Proc. 2nd Goddard workshop on formal approaches to agent based systems*, 2002.

[8] I. Foster, N. R. Jennings, and C. Kesselman. Brain meets brawn: Why Grid and agents need each other. In *Proc. 3rd International Joint Conference on Autonomous Agents and Multi-Agent Systems*, pages 8–15, 2004.

[9] D. Gambetta. Can we trust trust? In *Trust: Making and Breaking Cooperative Relations*, chapter 13, pages 213–237. Basil Blackwell, 1988.

[10] G Grimnes, S. Chalmers, P. Edwards, and A. Preece. Granitenights – a multi-agent visit scheduler utilising semantic web technology. In *Proc. 7th International Workshop on Cooperative Information Agents*, pages 137–151, 2003.

[11] M. Luck, P. McBurney, and C. Preist. A manifesto for agent technology: Towards next generation computing. *Autonomous Agents and Multi-Agent Systems*, 9(3), 2004.

[12] P. McBurney and S. Parsons. Chance discovery using dialectical argumentation. In *Proc. JSAI 2001 Workshop on New Frontiers in Artificial Intelligence*, pages 414–424. Springer, 2001.

[13] T. J. Norman, A. Preece, S. Chalmers, N. R. Jennings, M. Luck, V. D. Dang, T. D. Nguyen, V. Deora, J. Shao, W. A. Gray, and N. J. Fiddian. Agent-based formation of virtual organisations. *Knowledge-Based Systems*, 17:103–111, 2004.

[14] O. Pachheco and J. Carmo. A role based model for the normative specification of organized collective agency and agents interaction. *AAMAS*, 6:145–184, 2003.

[15] H. Prakken. Analysing reasoning about evidence with formal models of argumentation. *Law, Probability & Risk*, 3(1):33–50, 2004.

[16] S. Ramchurn, D. Huynh, and N. R. Jennings. Trust in multiagent systems. In *Knowledge Engineering Review*, volume 19, pages 1–25, 2004.

[17] J. Shao, W. A. Gray, N. J. Fiddian, V. Deora, G. Shercliff, P. J. Stockreisser, T. J. 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 *Proc. UK OST e-Science 2nd All Hands Meeting (AHM'04)*, 2004.

[18] W. T. L. Teacy, J. Patel, N. R. Jennings, and M. Luck. Coping with inaccurate reputation sources: Experimental analysis of a probabilistic trust model. In *Proc. AAMAS05* , 2005.

[19] A. Whitby, A. Jøsang, and J. Indulska. Filtering out unfair ratings in bayesian reputation systems. In *Proc. of the Workshop on Trust in Agent Societies, at the Third International Joint Conference on Autonomous Agents & Multi Agent Systems*, 2004.

[20] Bin Yu and Munindar P. Singh. Detecting deception in reputation management. In *Proc. of the 2nd International Joint Conference on Autonomous Agents and MultiAgent Systems (AAMAS)*, pages 73–80, Melbourne, Australia, July 2003. ACM Press.