Poslad S. Using Multi-agent Systems to Specify Safe and Secure Services for Virtual Organisations. M. Barley et al. (Eds.): SASEMAS 2004, Lecture Notes in Computer Science, 2009, Vol. 4324, pp. 258-273.

Using Multi-Agent Systems to Specify Safe and Secure Services for Virtual Organisations

Stefan Poslad

Department of Electronic Engineering, Queen Mary, University of London,
Mile End Road, London E1 4NS
stefan.poslad@elec.qmul.ac.uk

Abstract. Organisations increasingly interoperate with others to participate in Virtual Organisations or VO. This leads to models that need to deal with open service interaction with multiple levels of openness. Other VO security and safety challenges include decentralised operation and management, interoperability, balancing privacy versus accountability, managing error events with multiple semantics and managing anomaly events versus normal variations. The interrelationships, interactions and behaviours of agents in MAS and multi-MAS (MMAS) are analogous to those in VO. MAS provide a useful method of modelling VO, to manage their and to make them secure and safer.

1 Introduction

Distributed system security is primarily concerned with using cryptographic protocols to protect organisational assets against external threats, treating the organisation as a trusted system internally without security checks – a fortress model of security. Security safeguards protect organisational assets such as the network infrastructure, processing, services and data against threats such as information disclosure, corruption, masquerade, unauthorised access, denial of services and repudiation. In addition, more finely grained access control often used internally according to the organisational role, level of access e.g., Clark-Wilson [1] and Bell-LaPadula [2] and the type and value of the asset – a compartmentalised fortress model. These in turn use sets of authorisation rules or policies to define what type of access a user has to a type of asset.

Security requires additional security management protocols often at an inter-organisational level. For example, external Trusted Third-Parties (TTP) are used by organizations to create the authorisation and authentication credentials they distribute, configure and revoke. TTP are trusted not to make copies of credential for others, not to provide back door access and to produce easily forgeable credentials.

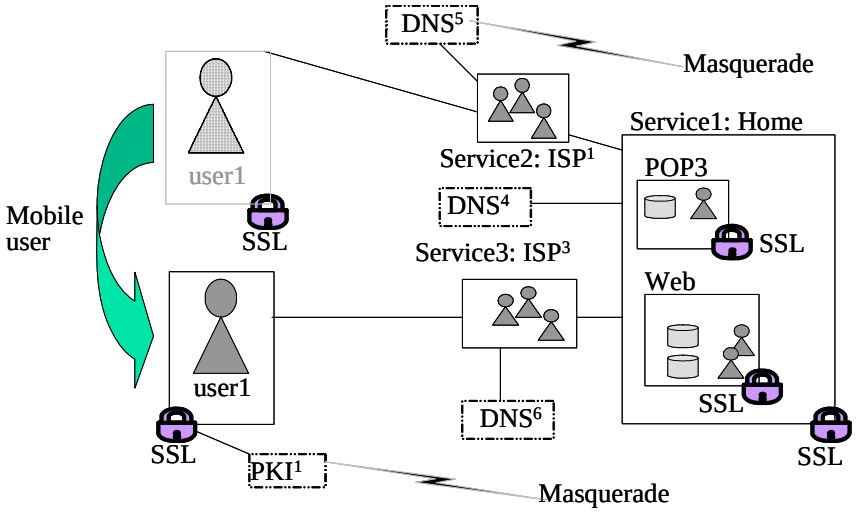

Fig. 1. A scenario to illustrate the use of a fortress security model to give secure access to POP3 and Web services to a mobile worker on his home network who must then switch to use another ISP to access them. However, DNS and PKI services are threatened by a masquerade.

The human users of ICT or Information, Communication Technology, including their organisational inter-relationships and human-ICT relationships, act as additional vital organisational assets that need to be protected. Obviously, humans such as hackers can act malevolently, e.g., stealing unauthorised information, and threaten security. However, people can also frequently inadvertently weaken security. For example, people can non-optimally configure system security, e.g., switching it off or to a minimum level of security because it is too complicated for them to configure to a higher level. When a natural or man-made environment disaster occurs, local mobile wireless communications can get overloaded and cause a Denial of Service or DoS as each participant or observer reports the event to others, perhaps preventing local emergency and help messages getting though. Humans can also weaken security systems because they often add an additional level of indirection that can be compromised. For example, humans prefer to use domain names rather than numerical IP addresses in order to address Internet assets such as Web servers. A mapping service such as the Domain Name Service or DNS must be used to map human readable names into numerical IDs and vice versa. DNS can be attacked by altering DNS messages and DNS data so that messages get redirected to a different destination server than the one intended by the sender. Humans thus often represent the biggest security challenge of the whole system [3].

Whereas security commonly refers to the ability to protect services against malicious threats using cryptographic protocols, safety refers to managing the system when failures occur. Failures may occur for several reasons because of system design errors, because of task errors that occur when they interact with another system and because of environment conditions or operators that cause a system to operate outside its normal, safe, operating conditions. Another definition of safety is that it is the probability that a system does not fail in a manner that causes catastrophic damage

[4]. Safety management is akin to failure management and has four main strategies. Fail-operational systems continue to operate when they fail, sometimes unsafely. Fail-safe systems become safe when they cannot operate. Fail-secure systems maintain maximum security when they can not operate. Fault-tolerant systems continue to operate correctly when parts operate incorrectly because parts are replicated and these can be accessed, ideally transparently, instead.

Security and safety management is complex because it deals with heterogeneous distributed organisational assets. It must not only deal with security and safety of the physical resources and services but it must also deal with the human stake-holders that interact with systems and it must deal with dynamic, virtual, inter-organisational boundaries.

1.1 VO Security and Safety Management

An organisation, e.g., a business firm, is an arrangement of relationships between individuals that produces a system endowed with qualities not present at the level of individuals. It ensures a relatively high degree of interdependence and reliability, providing there are procedures in place to replace individuals and their interactions within the organisation, thus providing the organisation with the ability to persist in the face of disruptions. The relationships between individuals within the organisation are determined by the interactions and their interactions are in turn constrained by the organisation. These interrelationships exist only within organisations, but reciprocally organisations require these relationships in order to exist. Behaviours or actions are normally performed internally within organisations such as firms on economic grounds, i.e., only if they cannot be performed more cheaply in the market or by another organisation. It was partly for this finding in 1937 [5] that Ronald Coase was awarded the Nobel Prize for economics over 50 years later.

There are three generic types of Virtual Organisation or VO [6]. Firstly, there are organizations that extend some of their organisational activities externally, thus forming virtual alliances to achieve (shared) organisational objectives. E-commerce organisations that participate in supply-chains and coalitions of military and humanitarian forces are examples of this type. Secondly, a virtual organization can be related to a perceptual organization that is "abstract, unseeing and existing within the minds of those who form a particular organization". The third type of virtual organization is established when corporations are distributed and intensively use ICT to support this such as the use of Virtual Private Networks or VPN, e.g., see Figure 1. Primarily, the focus of this article is on modelling security for the first type of VO.

1.2 Paper Outline

The remainder of this article is organised as follows. The next section, 2, describes the requirements for VO security and safety. Section 3 discusses the design and implementation using MAS as a design model and technology for implementing VO. MAS are also considered as primary entities to manage VO security and safety. Section 4 presents conclusions.

2 VO Security and Safety Challenges and Requirements

Challenges	Characteristics
Open Access vs. Regulated Access	Pure open service access, any time, any place, anyhow, anybody access is unregulated & difficult to secure. May need to span across infrastructures, owned by others that can only be indirectly managed, e.g., to delegate authority,
Decentralised Operation and Management of services	No single authority; multi autonomous stakeholders and autonomous groups. Emergent behaviour can arise and violate existing security One system can compromise the security and safety of another.
Security Interoperability	Services and users may reside in multiple administrative domains. Multiple security mechanisms in use across domains.
Balancing privacy vs. accountability	Masquerades are easier Degrees of privacy from anonymity to public identification
Managing error events with multiple semantics	Higher-level policies may not be directly implementable at a lower-level Low level faults often not propagated to, are not understandable, at a higher level
Managing Anomalies versus Normal variations	Deterministic vs. semi-deterministic vs. random Normal perturbations vs. Anomalies. Intended attacks vs. unintended failures.

Table 1. Virtual Organisation Security & Safety Challenges and Requirements.

The security and safety challenges for VO are summarised in Table 1. Each type of challenge is discussed in more detail below.

2.1 Open Access vs. Regulated Access

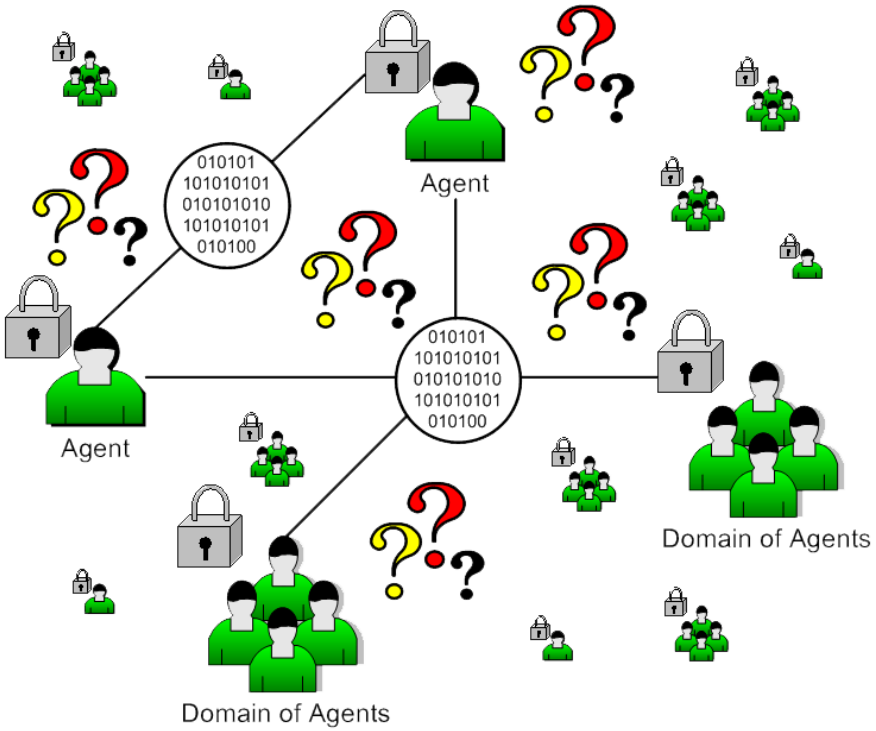

Fig. 2. Heterogeneous service access and interoperability in multiple domains

Virtual organisations tend to use open service environments in order to interoperate. A pure open service environment supports ubiquitous access: any time, any place, anyhow by anybody. This openness has several dimensions. It refers to service interfaces being public available. Services are accessible over a public network. It enables heterogeneous implementations of the services to be built that inherently interoperate. Heterogeneous interfaces can be aligned to each other, possibly with some information loss. Securing heterogeneous services is dealt with further in section 2.3. Services are extensible so that new services can be synthesised as composites of existing services and that old service interfaces can be extended. Users can dynamically bind to and unbind from service invocations, possibly interacting via third-party mediators. Services are scalable, supporting a range of concurrent users.

The operational management of open service environments requires that it both supports any desired dynamic configurations and service operations yet restricts any undesired configurations and operations. In practice, open doesn't necessarily mean that access costs nothing, that anyone can have access; pure openness is often replaced by regulated openness. VO tend to be secured using cryptographic protocols, see Fig. 2. It means that anyone can apply and register providing they meet the access policy constraints. For example, there is an age limit to purchase alcohol and lottery

tickets. Hence an important technique for dynamic system management is policy-based management coupled to cryptographic security protocols. Policy-based management defines the management rules or policies that are activated when service events occur such as unauthorised resource access.

2.2 Decentralised Operation and Management of Services

Within a virtual organisation, the individual organisations act with a degree of autonomy. They interact more on a peer to peer level, without a central manager, in part, because they are autonomous and because of openness. They may draw up and adhere to contracts and agreements that formalize commitments between them and that have a legal force and legal consequences. Both the users and provider entities within a transaction can simply switch to another similar entity if a problem occurs, providing that this is not prohibitively expensive to do so when breaking their contractual agreement, e.g., they may judge the financial penalty is small or that it requires too much time and financial resources for others to bother with.

An important issue is how service users and providers select each other. The dominant model in distributed computing is that service providers advertise their capabilities in service directories and that the choice is made on the best match service capabilities by the user, but this often not user centric, i.e., it may not be linked to a rating of the actual services delivered. Before choosing to delegate tasks to others to act on their behalf, some sort of local evaluation of others, perhaps using recommendation, ratings, reputation and referrals from others or past experiences of oneself, can be carried out. One party may also decide to trust another party, that is, the trustor believes that the trustee is benevolent and competent and voluntary delegates tasks to the trustee with no real commitment from the trustee [7]. Within an open service environment and in particular on ones based upon trust, it is far easer for masquerades to occur. For example, convicted criminal hacker Kevin Mitnick testified before U.S. congress that he often gained security credentials by simply masquerading as someone else. Ironically, another hacker gained a reduced criminal sentence by having encrypted the evidence of his actions.

There is also the issue of how to audit, classify and manage interdependent interactions when they are orchestrated across multiple organisations and domains. It is common in cases where some ambiguity about the cause of events exists for the organisations involved to compete to get others to fix a problem in order to reduce their own costs. It may also be unclear how to identify the cause and to handle a fault when it is arises because of interference between two autonomous systems, so called emergent behaviour. It may also not be possible to determine the exact or even the probable cause of events. [8] has outlined the following security challenges for managing emergent properties, in ad-hoc networks. Emergent properties represent new beneficial security features of ad-hoc networks, e.g., to help establish or revoke trust relation. Some emergent properties are undesirable and may lead to security violations e.g., emergent properties may bypass traditional intrusion-detection techniques and mask that a certain network node has been captured by an adversary. Third, inadequate assessment of emergent properties such as false detection may also lead to

security and robustness violations. For example, the false detection of node capture may lead to network partitioning and denial of service in an ad-hoc network by the unnecessary revocation of node membership. Fourth, the understanding of the characteristics of emergent properties helps determine scalability and resilience.

2.3 Security Interoperability

ICT environments are naturally open; providers and users often discover and evaluate each other's capabilities and preferences dynamically, enabling users to invoke providers' services on-the-fly. Interoperability in a heterogeneous open service environment can be greatly aided by using public service specifications and by specifications being in a computation form that is able to be analysed and to be aligned to related heterogeneous specifications. For example, different parties may use different kinds of authentication certified by different TTF – these will require the use of some sort of authentication certificate gateway to allow each other to interoperate. Interoperability is aided when they can share semantic type metadata to support a common understanding between them.

2.4 Balancing privacy versus accountability

Privacy can be regarded as both a basic human right to be left alone and as a user freedom to not be observed by others. In practice, a portion of privacy is often traded in order to have greater convenience in order to interact with others such as other human users and service providers, e.g., to avoid entering the same personal details for repeat transactions. [9] has identified intermediate levels of identity between true identity and true anonymity. He distinguished four levels of ''nymity:'' anonymity, non-reversible pseudonymity (not tied to a true identity), reversible pseudonymity (tied to a true identity), and identity. Maximum disclosure and invasion of user privacy occurs when a user's personal details are revealed and linked to additional contexts such as the current location or to payment credentials.

There are also concerns with identity management and accountability within open systems. It may be difficult to assign an accountable identity or address to the perpetrator of a malicious action within an open service environment, because the identity and address can be easily masked or spoofed. This represents a significant weakness in authentication services as unauditable IDs and addresses may be bound to authentication tokens thereby restricting the usefulness of the tokens. Therefore, this brings forth a fresh type of problem whereby malicious intent can no longer be bound to an accountable identity or address and where varying contextual meanings can represent hidden malevolent agendas.

2.5 Managing events and errors with multiple semantics

VO potentially have a greater variety of failures because of the larger number and possible dynamic compositions of service components. They also use a more complex

operating environment, leading to transient and intermittent failures and because status messages and error messages may trigger failures that are not understood by the application processes. A further complication in an open distributed system is that Byzantine type faults may be more likely. Byzantine faults occur when processes continue to operate issuing incorrect results or possibly work together with other faulty processes to give the impressions that they are working normally.

Either errors can be trapped and handled internally within the application or they can be handle using some sort of world model that can be shared across applications [10]. The application-level approach often involves instrumenting the application code to generate events and to catch the fatal ones. This has the important disadvantage that the event can only have a context within an application and can't easily have a more global context across applications. A second disadvantage is the semantics of the error is designed and annotated at the level of the software infrastructure in which the application execution. However, users may lack an understanding of the low-level infrastructure semantics.

Conventionally, failures can be handled in several ways. A very common technique is to handle them locally in the application using static exception handlers. Failures can be masked, by switching to a replicated copy if fault-tolerance is supported or by catching it and simply propagating infrastructure errors up to a user process. However, application level processes and users often cannot understand the meaning of errors because they are at a lower level of abstraction or because they use a particular perspective of understanding that users are unfamiliar with. For example, one well known personal firewall for computers expects users to be able to understand which types of network protocol ports an application should be allowed access to in order to get through the firewall. Users may not understand why a service stops behaving as expected if no information is given and so may not be clear which type of remedial action, such as a restart, should be used.

2.6 Managing normal variability versus anomalies

Simple open service environments can be designed to support Event-Condition- Action (ECA) type interaction. Conditions constrain both how events cause actions and the order in which actions normally flow, e.g., a metadata directory is queried first to identify which data resources hold certain types of data, thus avoiding wasting time querying data resources that do not hold the data. It is important to model both normal and invariant pattern of events. Specific normal variant event and action patterns are specified that need to be handled so that they do not disrupt the system, e.g., a resource manager may receive a flood of repeat event requests because a user receives no acknowledgement that his request has been received.

In more complex distributed systems, users, providers and mediators have more autonomy; sub-systems can refuse to carry out requests even although they have the capability to do so. Systems can have the flexibility for multiple and concurrent service providers and users to dynamically bind and unbind to each other and for actions to be orchestrated into work-flows at run-time rather than at design-time. More complex behaviours can also emerge from the interplay between simple ones and a com-

plex environment. Anomalous events or actions, not defined in the standard or expected set of planned normal behaviours can arise.

3 MAS Designs to Support the Safety and Security Management of VO

Challenges	MAS Solution
Open Access vs. Regulated Access	Agent can *manage policies* dynamically to regulate access. Agents can *reason* about which are the important parts of interactions to secure and which to leave public. Agents can *share tasks and goals* with other agents across organisational boundaries and 'delegate' these to others to act on their behalf.
Decentralised Operation and Management of services	Different types of agents act as multi *autonomous* stakeholders and autonomous groups. Agents can share information about security threats that anyone detects, acting as a *neighbourhood watch*. Simple interactions between agents can lead to the emergence of global behaviour – *swarm intelligence*.
Security Interoperability	Agents can be used support *semantic mediation* between different security protocols
Balancing. Privacy vs. accountability	Agent mediators can be used as TTP and can use *policy-based management* between customers and providers so that identity and details of customers are *revealed on a need to know basis*. TTP agent mediators that reveal private information unnecessarily can suffer a *loss of reputation*, gain a low merit mark for their quality of service. Agents *can share knowledge of events and expertise* to allow services to be managed across organisational boundaries.
Managing events: mismatch of high-level management vs. lower-level operational, application, ones	Agents can be used to support *semantic mediation* between different security views, protocols, concepts and policies. Agents can *model events and reason about them externally to the application* processes in which they occur
Managing Anomalies versus Normal variations	Normal goal-directed behaviours can be modelled using explicit plans by agents. Agents can also *re-plan* when events are detected that cause disruptions to the plan. Agents can *explicitly model environments' expected events and infer unusual events.*

Table 2. MAS use to support VO safety and security

Multi Agent Systems (MAS) are a distributed artificial intelligence problem model that can be solved and reified using a range of technological solutions from procedural and object-oriented to declarative and to a wide spectrum of logic programming. MAS are organisations of individual agents. The properties of agents are that they are goal directed and can behave with a degree of autonomy. They can react instinctively to interactions and proactively initiate interactions, or they can deliberate about their

interactions. They can interact cooperatively, to share common goals, tasks, information and tasks or they can act in a self-interested competitive manner.

Software type of agents, not only interact with other agents but they are also situated in a non-agent environment, an ICT infrastructure. They can sense events, process them and affect the environment. The link between the sensed events, the behaviours or actions that agents undertake and any subsequent effects in the environment depends on the type of agent such as being deliberative, reactive, goal-directed, utility-based or a combination of these. The interrelationships, interactions and behaviours of agents in MAS and multi-MAS (MMAS) are analogous in many ways with those of members of social and economic organisations and virtual organisations.

The first major design issue that needs to be considered in using a MAS model of VO security is whether or not security should be managed by agents within an organisation or delegated or trusted to others external to an organisation such as to the ICT infrastructure environment. The degree of security control under the management of the agent needs to be considered. This can range as follows:

- Agents (application) have no control of security: they delegate security management to the infrastructure or to third parties;
- Agents manage security at a coarse level: they can monitor security changes directly or indirectly; they can switch on and off their own security; they can switch between various levels of security;
- Agent manage security at a fine level: They can deliberate and control security using semantic and social collaborative models.

There are a number of advantages to using MAS to manage security that are expounded in the next sub-sections and summarised in Table 2. These are related to the VO security challenges and requirements given in Table 1.

3.1 MAS support for Open Access vs. Regulated Access

The organisation or application domain can specify policies to say who has what access to which organisational assets. Agents can reason about the instances of policies and policy flows that are triggered when agents access. It is not sufficient for agents to just reason about triggered policies, agents should also reason about when to trigger security protocols in situated actions of the work-flow. To do this, agents will need to track work-flows for tasks and assess the policies for when security should be used, and at what level. Polices may need to change dynamically, e.g., if threat levels are perceived to have changed. When individual organisations interact within a VO, policy mediation may be needed to check the compatibility and priority of policies when multiple ones apply and to decide which ones should be applied.

In the past, no single framework existed to express domain specific Ontologies, rules and reasoning about the ontology concepts and rules. With the advent of the W3C Web Ontology Language, OWL, there is a single framework to express rich concept structures and relationships, rules in the form of constraints on concept classes and properties and a type of description logic to reason about concepts.

3.2 MAS support for Decentralised Operation and Management of services

Safety and security can be increased using MAS models because autonomous agents can build different independent viewpoints or models of the nature of these failures. Agents can be designed to use multiple heterogeneous plans to achieve goals in the face of limited environmental failures and malicious attacks. In addition, they can socialize, acting together as a neighbourhood watch and use triangulation, i.e. using a combination of multiple view-points, to understand a given problem or situation improving the limitation of relying on any single viewpoint of the failure, see Fig. 3.

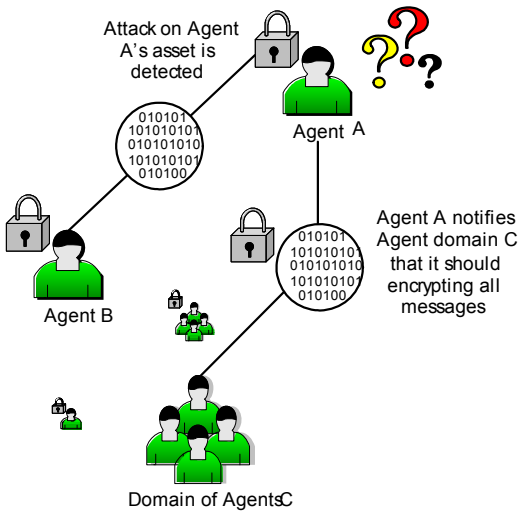

Fig. 3. One agent can socialise with others about an intrusion it detects

Computer systems will need to manage themselves according to high-level objectives specified by humans, otherwise the increasingly complex, heterogeneous distributed systems, such as ubiquitous computing systems will become impossible to administer [11]. IBM first introduced this vision of self-managing system in 2001 when it launched the autonomic computing initiative. Autonomic Systems support four basic properties: self-configuration, self-healing, self-optimization and self-protection.

VO often rely on norms or group prescriptions to constrain its members to do or not to do actions. They rely on the obedience of trustees to adhere to the norm. Swarm behaviour is influenced by instinct, by local interactions, past experiences. Herd behaviour is influenced by market-leaders. If there are no norms, chaotic behaviour can arise. There is often a local convergence to norms but this may cause global polarisation within a VO because local norms are orthogonal [12].

3.3 MAS support for Security Interoperability

Open service environments for VO such as those based upon public Web service specifications from the W3C and public MAS specifications from FIPA, the Foundation for Intelligent Physical Agents [13] provide the means for different actors within a VO to interoperate, more easily, and at a lower cost. New entrants to the market can more readily publish and offer services, users can more easily interact to select services, and mediators can more easily provide various value-added channels between providers and users. Not only should the service interactions use public specifications but the service management should also use public security protocols in order to operate. Earlier applications of MAS security tended to use proprietary security mechanisms, for example using non-standard certificates for authentication [14], [15] this tends to lead to closed systems that cannot interoperate with others as part of a VO.

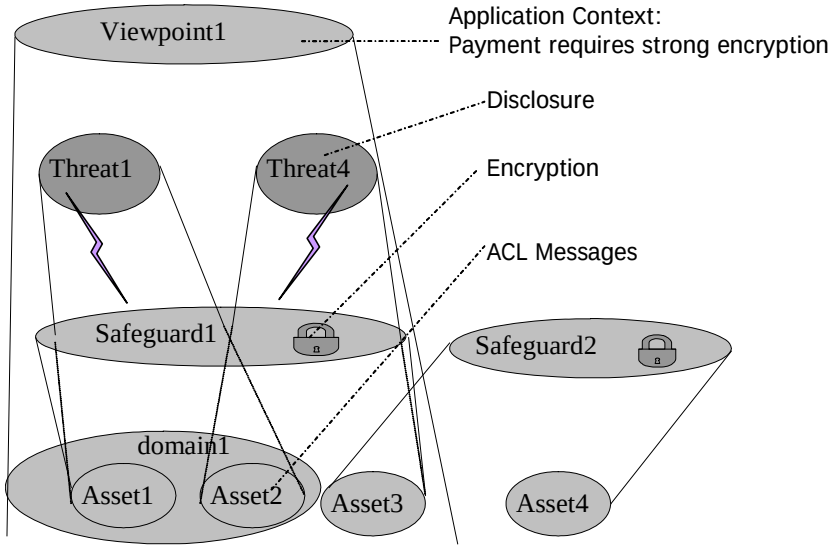

Fig. 4. The V-SAT model, Viewpoints of Safeguards that protect Assets against Threats

Although, security protocols based upon public specifications are readily available, there are great many to choose from and the same security protocol can be configured differently to adhere to different organisational security models. A semantic mediation approach based upon a core model, the V-SAT model, Viewpoints of Safeguards that protect Assets against Threats, see Fig. 4, has been proposed in [16] to support security interoperability using multiple security protocols. This has been represented first in DAML+OIL and then in OWL.

```
<!-- Policy Instances -->
<secprofile:AgentCondition rdf:ID="PlatformCredentialPolicy">
        <policy rdf:range="xsd:String">
        (exists (?c (Security_Ontolo:SignatureMethod ?c))
```

```
            (or (Security_Ontolo:Algorithm ?c  http://www.w3.org/2000/09/xmldsig#rsa-sha1)
            (Security_Ontolo:Algorithm ?c http://www.w3.org/2000/09/xmldsig#dsa-sha1) )
            </policy>
    </secprofile:AgentCondition>
    <secprofile:AgentCondition rdf:ID="PlatformProtocolPolicy">
            <policy rdf:range="xsd:String">
            (exists (?p (Security_Ontolo:Protocol ?p)) (and (Security_Ontolo:FIPASecurityProtocol ?p
urn:fipa:security:SecurityObject1) (Security_Ontolo:FIPASecurityProtocol ?p urn:fipa:security:SecurityObject2) )
            </policy>
    </secprofile:AgentCondition>
    ...
```

Fig. 5. An example policy constraining the algorithm used for digital signatures

Because (external) security configurations can be described at an abstract level, different mechanism choices can be specified in V-SAT model terms in instances of the viewpoint called application profiles. For example, support for encryption using DES or AES and message hashing using MD5 or SHA-1 can be specified. Constraint or policy based reasoning can be used to see if a common security configuration can be agreed between different communicating parties, see Fig. 5.

3.4 MAS support for balancing privacy versus accountability

Protecting the privacy of the user context, e.g., user ID, service preferences, past interaction history, personal details and payment credentials, requires more than supporting confidentiality and authorisation using encryption mechanisms – a multilateral approach is needed [17]. This approach offers four different levels of privacy: owner-centred privacy management, use of different degrees of anonymity, restricted disclosure of the user context on a need to know basis and to detect the misuse of privacy and to penalize those who do so. The privacy model is supported in a system called USHER, U-commerce Services Here for Roamers. There are several elements to this system. It uses a MAS organisation to model and associate roles to participants in the organisation and to control access to the user context according to specified context sharing policies. These roles are: an Owner of a user context; a Holder e.g., GPS device, authorised to collect user context information,; a Requester, e.g., route-planner; authorised to get an owner's context; Nymisers that act as mediators, that are authorised to selectively reveals owners' contexts to others and an Authoriser that issues credentials for rights to access user-contexts, see Fig. 6.

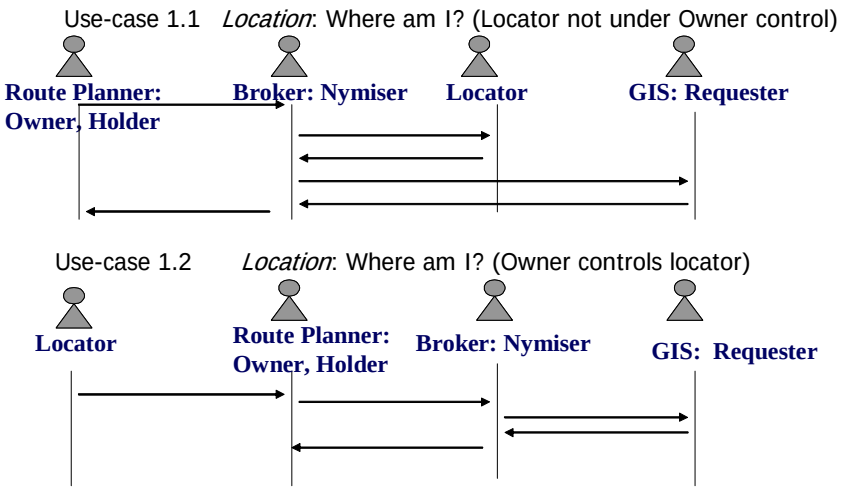

Fig. 6. Organisation interaction and roles for controlled sharing of a user's location context that has been created using two different location determination systems: client-side, e.g., GPS, and server-side, e.g., wireless network trilateration.

Because of the richness of the interaction between the different actors involved in exploiting and safeguarding the owner contexts, problems due to lack of a shared semantic model for privacy can arise [18]. These include: the inability to interlink service processes that access the user-context with the security processes that safeguard them; the lack of a common terminology to allow owner contexts to interlink to management concepts that are understood by the different actors and the lack of rich meta-data to manage the operation of the privacy safeguards. Hence, not only a formal policy-model is needed to support the management of owner preference access but also for a semantic model is needed to support it. The privacy model in [18] is based on the V-SAT model given above where the main asset that needs to be protected against threats is the user context. There is a W3C standard that could be used to express privacy policy called P3P but this is provider- not user-centred. P3P needs to be enhanced to support user contexts and to link the policy evaluation to be under user control. This system has been applied to two applications – a location-aware religious service that locates the nearest Mosque for Muslims and a restaurant recommender service [18].

3.5 MAS support for managing events and errors with multiple semantics

Interoperability amongst the various open system actors can be greatly aided by sharing semantic type metadata. A Semantic-based failure communication model based on an integrated failure and security ontological model has been developed as an extension to the V-SAT model [19]. This is combined with a semantic profile model to interlink policy models and process models of normal and failed behaviour can support improved, higher level, application oriented fault management. The semantic

model can also support a reasoning model to support adaptive fault and security management and the choice of an appropriate fault handler. Detected errors can be propagated and transformed when necessary into semantically tagged error events. In some cases errors are best handled and more efficiently handled locally whereas in other cases errors may best propagated up to the agent level because errors may have global significance.

In a VO, there are often multiple (sub-)domains, e.g. Restaurants and Hotels. Some parties trust one another and some do not, but each believes that honest parties are in a majority and that a domain administrator manages the admission and consensus of agents operating in its respective domain (Fig. 7) on its behalf. Traditionally, the verification of authenticated peers possessing private keys corresponding to the public keys determines if a party is honest and capable. This concept of identity is very general and insufficient especially in rich semantic environments which may span individuals and organizations.

The operational capability including security of systems and applications in open service environments is reduced when malicious and services are present. Such behaviour can be particularly hard to detect because of the dynamics of open systems and because it may only become apparent in certain service states and when a certain percentage of disruptive stake-holders or occurrences become noticeable. Whilst the Semantic Web has the potential to promote a richer interaction and interoperability between different parties, Byzantine behaviour may also arise because of the added complexity in unambiguously interpreting and processing semantic metadata and meta-processes in a consistent way between heterogeneous parties. The Byzantine Agreement Protocol (BAP) [20] can be used to tackle Byzantine behaviours. It uses a majority based rating mechanism for identifying and pruning Byzantine entities in a domain. It has received considerable attention for the design of fault tolerant systems. It seeks to establish a fault-tolerant agreement when one or more of the nodes in a system have been compromised or failed.

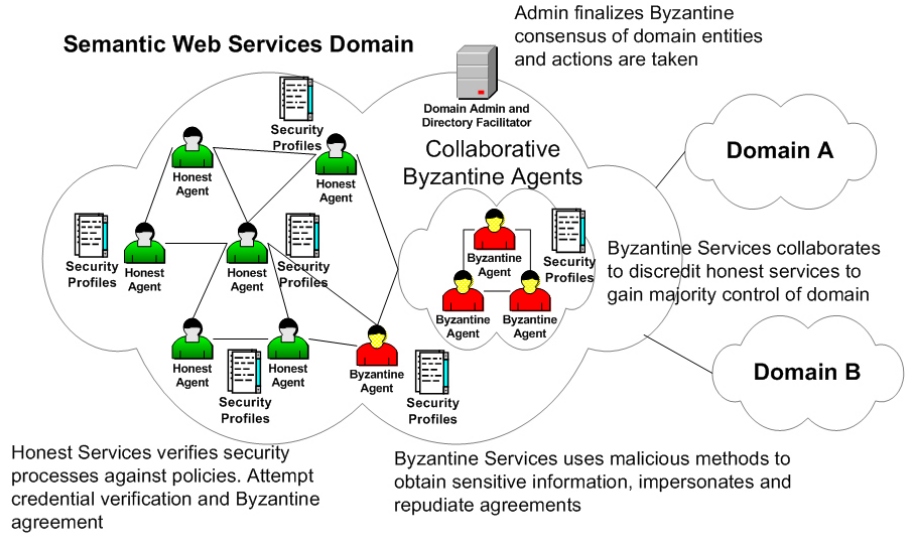

Fig. 7. Byzantine Services Scenario

3.6 MAS support for Managing Anomalies versus Normal variations

A basic assumption underlying anomaly detection is that anomaly patterns differ from normal behaviour patterns. However if may not always be not be possible to differentiate the two with certainty, e.g., a flood of repeat requests by a frustrated user and a flood of events used by a malicious user designed to cause a DoS are similar. In this case, the patterns of events need to be characterised along a number of features with respect to some normal distribution of these features and only a difference probability can be calculated between normal and anomaly events. There are two main strategies for detecting and handling anomalies. If predefined normal patterns of events are available, then either anomaly events can be matched against anomaly pattern sets or against normal pattern sets. Secondly, if normal patterns are not available, they can perhaps be learnt from observation of operational events as normal events tend to cluster whereas anomalies tend not to [21].

4. Conclusions

Although cryptographic security protocols represent the core protocols to safeguard distributed systems a multi-lateral approach is needed, cryptographic protocols need to be complemented with additional approaches based upon policy-based management, semantic-based mediation, agent-based task and information sharing and policy assessment. Some behaviours of VO such as Byzantine faults, anomalies, emergent, human and social behaviours are hard to model and will require more research to find ways to manage the security and safety aspects of these more effectively.

Acknowledgements
The research described was aided by the endeavours of the following research assistants: Juan (Jim) Tan and Leonid Titkov.

References

[1] Clark, D.D. Wilson, D.R.A.: Comparison of Commercial and Military Computer Security Policies. IEEE Symposium of Security and Privacy (1987) pp 184-194

[2] Bell, D.E. LaPadula, L. J.: Secure Computer Systems: Mathematical Foundations. MITRE Technical Report 2547, Volume I (1973)

[3] Schneider, B.: Secrets and Lies: digital security in a networked world. John Wiley & Sons. ISBN 0-471-25311-1 (2000)

[4] Nicol, D., Sanders, W., Trivedi, K.: Model-Based Evaluation: From Dependability to Security. IEEE Transactions on Dependable and Secure Computing, vol. 1:1 (2004) 48-65

[5] Coase, R.H.: The Nature of the Firm. Economica 4 (1937) 386–405

[6] Shao, Y.P., Lee, M.K.O. Liao, S.Y.: Virtual organizations: the key dimensions. Proc. Academia/Industry Working Conference on Research Challenges (2000) 3 – 8

[7] Coleman, J.S.: Foundations of social theory. Harvard University Press. ISBN 0-674-31226-0 (1990)

[8] Gligor, V.D.: Security of Emergent Properties in Ad-Hoc Networks. Proc. of the Security Protocols Workshop, Cambridge, UK, (2004)

[9] Goldberg, I. A Pseudonymous Communications Infrastructure for the Internet, PhD thesis, Univ. of California at Berkerley (2000)

[10] Garlan, D and Schmerl, B.: Model-based adaptation for self-healing systems. Proc. 1st workshop on Self-healing systems, Charleston, South Carolina, (2002) 27 - 32

[11] Kephart, J.0. Chess, D.M.: The vision of autonomic computing. Computer, 36:1 (2003) 41-52

[12] Axelrod, R.: The Dissemination of Culture: A Model with Local Convergence and Global Polarization. Journal of Conflict Resolution, 41:2, (1997) 203-226.

[13] Poslad, S, Charlton, P.: Standardizing agent interoperability: the FIPA approach. In: Michael Luck, Vladimír Marík, Olga Stepánková, Robert Trappl (Eds.): Multi-Agent Systems and Applications, Lecture Notes CS, Vol. 2086 (2001) 98-117

[14] Zhang, M, Karmouch, A, Impey R.: Towards a Secure Agent Platform based on FIPA. Proc. MATA 2001. Springer-Verlag. LCNS, Vol. 2164, (2001) 277-289

[15] Ghanea-Hancock, R, Gifford, I.: Top secret multi-agent systems. 1st Int. Workshop on security of mobile multi-agent systems (SEMAS-2001), 5th Int. Conf. Autonomous Agents, Montreal, Canada (2001)

[16] Tan, J.J., Poslad, S.: Dynamic Security Reconfiguration in Semantic Open Services Environment. Eng. Apps. of AI, Vol. 17 (2004) 783-797

[17] Titkov, L., Poslad, S., Tan, J.J.: An Integrated Approach to User-Centered Privacy for Mobile Information Services. Applied Artificial Intelligence J., 20: 2-4 (2006) 159-178

[18] Tan, J.J., Poslad, S., Titkov, L.: A Semantic Approach to Harmonising Security Models for Open Services. Applied Artificial Intelligence J., 20:2-4 (2006) 353-379

[19] Poslad, S., Tan, J.J., Huang, X., Zuo, L.: Middleware for semantic-based security and safety management of open services, Int. J. Web and Grid Services, 1: 3-4 (2005) 305 – 327

[20] Lamport, L., Shostak, R., Pease, M.: The Byzantine Generals Problem. ACM Transactions on Programming Languages and Systems, 4:3 (1982) 382-401

[21] Steinwart I., Hush, D., Scovel, C.: A Classification Framework for Anomaly Detection. J. Machine Learning Research 6 (2005) 211–232