A SEMANTIC APPROACH TO HARMONIZING SECURITY MODELS FOR OPEN SERVICES

Juan Jim Tan, Stefan Poslad and Leonid Titkov

Query Sheet

Q1 Au: ok?
Q2 Au: (year)?
Q3 Au: city?
Q4 Au: need full cit.
Q5 Au: (year), author if not, just give a URL
Q6 Au: Please cite figure 2 in text.

3b2 Version Number : 7.51c/W (Jun 11 2001)
File path : p:/Santype/journals/Taylor&Francis/Uaai/v20n2-3/uaai148412/uaai148412.3d
Date and Time : 20/12/05 and 11:47

Applied Artificial Intelligence, 20:1–27
Copyright © 2006 Taylor & Francis Group, LLC
ISSN: 0883-9514 print/1087-6545 online
DOI: 10.1080/08839510500484298

Taylor & Francis
Taylor & Francis Group

A SEMANTIC APPROACH TO HARMONIZING SECURITY MODELS FOR OPEN SERVICES

Juan Jim Tan, Stefan Poslad, and Leonid Titkov □ *Department of Electronic Engineering, Queen Mary, University of London, London, United Kingdom*

□ *There is a plethora of different security standards proposed by a range of standards consortia, including the IETF, W3C, and OASIS. There are also sometimes multiple configuration settings for a given security specification. In a heterogeneous open service environment, the variety of security standards and possible settings used can hinder security interoperability, because a common security configuration may not be able to be agreed upon in advance. In this paper, we have developed a generic security model expressed in an XML extension (DAML) and have investigated how to ground this in order to reuse the security specifications from various standards consortia. We have applied this model to support security discovery and dynamic security reconfiguration for use within open service infrastructures.*

INTRODUCTION

This paper describes a holistic security ontology model for the development of distributed open security systems. The main objective is to describe how distributed security management involving interoperability is approached in open dynamic heterogeneous service environments such as multi-agent systems multi-domains (MAMD). Although numerous initiatives have developed models and specifications for the interoperability of distributed security, the lack of a holistic solution that harmonizes the various models is a major obstacle in the development of open systems for use by business critical applications. The term *open* refers to services whose interfaces are based on publicly defined interfaces and implementations that adhere to consensual standards where available and that can be dynamically offered and accessed over public infrastructures. More research is needed to support service process models for distributed heterogeneous

We acknowledge the support of the EU Agentcities.RTD (IST-2000-28385) for this work and the EU iTrust (IST-2001-34910) projects for disseminating this work.

Address correspondence to Stefan Poslad, Department of Electronic Engineering, Queen Mary, University of London, Mile End Road, London E1 4NS, United Kingdom. E-mail: stefan.poslad@elec.gmul.ac.uk

services in which the management of security configurations can be discovered, orchestrated, and enforced using policies.

In this paper, an ontological model has been developed to link a variety of security specifications into a general yet epistemologically rich meta-data representation. As security is a process and middleware, it is not just a set of mechanisms; the binding of the ontology model to policy and service models is also considered. Agent or Web services descriptions commonly have three main parts: the service *profile* for advertising and discovering services; the *process model*, which gives a detailed description of a service's operation; and the *grounding*, which provides details on how to interoperate with a service (DAML-S). An essential component of the profile is the specification of what functionality the service provides and the specification of the conditions that must be satisfied for a successful result. The condition of a service in this paper is defined as a policy for specifying constraints of instances associated with security functions.

To make these ideas more comprehensible and appraisable, a scenario is given in Figure 1. The example describes an open environment setting where different systems publish their services along with their externally

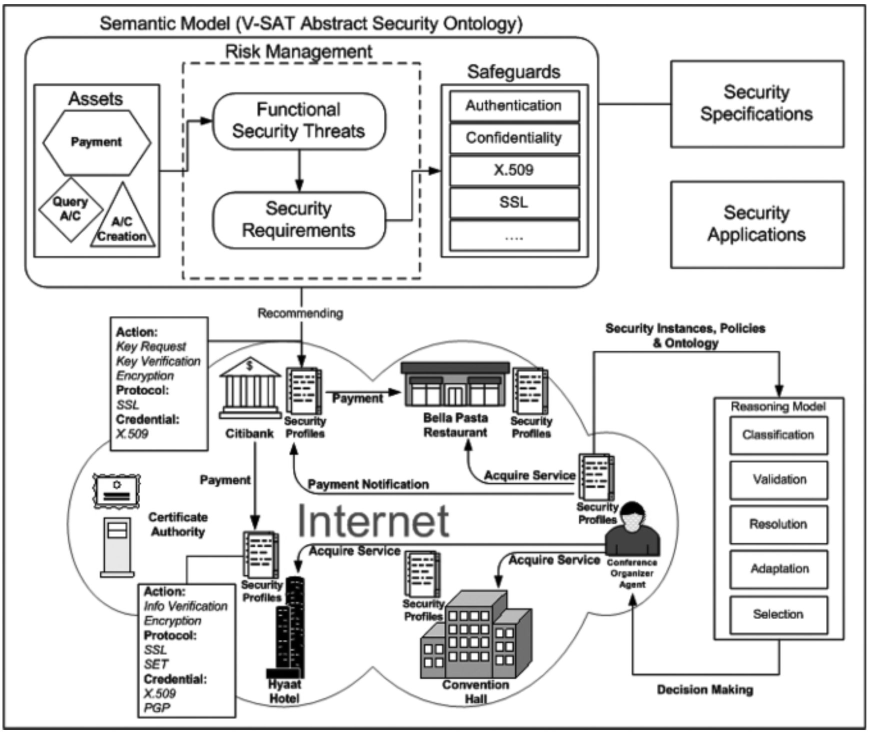


FIGURE 1 Profile driven security scenario.

public security configuration and requirements. These service descriptions include a detailed service process description of their security choreography or workflow. The security processes are represented as profiles enforcing their respective security configurations (protocols, credentials, and actions) and policies. Some of these configurations include combinations such as SET or SSL, X.509 certificates, and "confidentiality" or "authentication" actions. In the scenario, a conference organizer agent (COA) wishes to organize an event and interacts with services such as a convention hall, a restaurant, and a hotel. Initially, the agent discovers these services through directories and discerns the security choreography and profile. Through the profile description and policies governing these security instances, an agent is able to reason about the profiles with the aid of the holistic security ontology, hence capturing and understanding the concepts and processes needed to support interoperability between disparate services. Eventually payment is made through the banking service and a conference event is organized. The security profiles represented by each service are created by the security ontology and risk application models. Agents interacting in this environment utilize security meta-data and reasoning models for configuring services to meet specific security requirements.

Drivers for Holistic Security Model

When operating in a closed homogeneous domain, the use of static security configurations must be specified and agreed upon in advance. For example, Web-clients can use HTTPS to support confidential information exchange with a Web service provider. However, in order to operate within a heterogeneous domain consisting of autonomous heterogeneous stakeholders, a more dynamic approach to configure and manage security is needed.

A semantic-based holistic model captures multiple security stakeholders using abstract and explicit formalisms based on existing standards. The domain knowledge (ontology) is separated from the control knowledge (profiles): Systems can manage and reconfigure themselves without affecting their underlying implementation. By applying such a separation, reasoning becomes particularly useful within open service infrastructures as it enables us to detect, analyze, and resolve multiple policy conflicts; to decide if a change in the environment necessitates a security reconfiguration; and to decide if a suitable level of security interoperability between heterogeneous systems is achievable.

The separation also partitions the environment into two abstractions: one at the domain level where its generality supports interoperability, and another at the control level where specific descriptions ground practical applications. Subsequently, this interlinks services from one

TABLE 1 Summary of Existing Solutions

Functionalities Specification	Confidentiality	Authentication	Integrity	Authorisation	Non-repudiation	Credential Management	Policies	Trust Delegation	Semantic Service Adaptable
XML Encryption	✓		✓						✓
XML Signature		✓	✓		✓				✓
XACML		✓	✓	✓	✓		✓		✓
SAML		✓	✓	✓					✓
XKMS			✓			✓			✓
SESAME	✓	✓	✓	✓					
PICS			✓	✓	✓		✓		
P3P							✓		✓
KeyNote			✓	✓		✓	✓	✓	
S/MIME / PGP	✓	✓	✓						
S-HTTP	✓		✓		✓				
SSL/TLS	✓	✓	✓						
IPSec	✓	✓	✓						

domain with another and represents some of the major contributions of this paper. The main drivers for a holistic security model are:

- There are many security specifications. None of these is able to provide a total security solution; they need to be combined (see Table 1).
- There is no common method for configuring inter-related external security configurations.
- Particular security specifications of external security configurations of systems at the application level may be proprietary and may not be available in a form to support automatic and dynamic security reconfiguration.
- Security specifications may not be expressed in a form that allows them to be used as part of a security process that can be interlinked into service processes.
- Public security specifications do not always explicitly represent a rich enough set of meta-data in order to define the semantics to correctly use a security mechanism.
- There are multiple settings for a specific specification. As a result it may be difficult to agree on a configuration.
- There are many stakeholders and many of them do not understand security operational requirements. There is no common terminology for security between stakeholders and across the different application domains. As a result, stakeholders may misconfigure security.
- Current offerings are incomplete, driven by a technology push such as the use of HTTPS and PKI. These architectures use hardwired mechanisms and are likely to be brittle.

Hence, there is a need for a semantic-based holistic security model that promotes the exchanging and reasoning of information to satisfy these general requirements.

SECURITY ONTOLOGIES: RELATED WORK

Security for open systems faces many new challenges when operating either in closed homogeneous domains or in heterogeneous domains. Whilst static security configurations can be specified, these lack flexibility in a dynamic heterogeneous domain, which has multiple security requirements. The current plethora of different active security standards indicates that no single security framework is suitable for use with heterogeneous distributed systems. This has lead to the use of mediating models that are able to bridge between disparate security standards (Denker et al. 2003). Here, security mechanisms and objects are represented using DAML + OIL ontologies in order to allow agents to specify security requirements and capabilities. Service descriptions are published in DAML-S in Denker et al. (2003), but this does not yet address other emerging service models such as *BPEL4WS* + *WSDL* (BPEL4WS) and *WSCI* (WSCI). User and service requirements are paired and negotiated if they do not match. A bridging method can be used to mediate between different security standards but there is no explicit conceptual model to support open heterogeneous security. Overly sophisticated negotiation of security requirements in open environments can result in complex systems that are too brittle and impractical for large-scale interoperable deployments.

The concept of an ontology is used in its broadest sense: ranging from flat sequential syntactical models, as advocated by the IETF and W3C byte stream protocols, to hierarchical syntactical models (XML) and dictionaries of security terms, and also more expressive frameworks based on RDFS and DAML. There are several inputs into a more general upper ontology for security.

First *taxonomy dictionaries* (TD) describe security concepts as terms that can be analyzed by a particular application. TDs are often used by intrusion detection and trust-based systems. In the area of standardization, trust first became an issue almost twenty years ago (DoD 1985). A few years later, formal methods for the analysis of cryptographic protocols were developed. Trust played an important role. For example, a most successful formalism in the field, BAN logic (Burrows et al. 1990), was developed. Other specialized trust management solutions appeared such as the W3C PICS (PICS) used to define formats and to distribute meta-data labels for the description of Web documents. AT&T has developed PolicyMaker (Blaze et al. 1996) that binds access rights to an owner of a public key using certificates. IBM recognizes that trust is at the center of e-business so it has developed a Java-based trust establishment module and a complementing control language (Herzberg et al. 2000). An extensive survey on trust has been published by Grandison and Sloman (2000). This survey defines trust informally. The main advantage of taxonomy dictionaries is that security concepts are explicitly defined. There exists a direct mapping between a particular action and an entry in a dictionary. Conversely, the major drawbacks of this approach are the difficulty of representing every single concept and action within the dictionary and the omission of defined relationships between concepts.

Second, *XML-based approaches*, where the semantics of an XML document is explicitly encoded within the document using tags as identifier, has the potential to support a finely grained security architecture. Until recently, most of the security systems created around XML standards have focused on protecting the transmission of documents. For example, SOAP (Simple Object Access Protocol) uses XML to encode messages to send across the network. As such, XML messages can be protected using HTTPS. These support confidentiality, integrity, and authentication. Newer XML-based security standards include SAML, XML Signature, and XKMS.

Third, *policy-based and access control solutions* support the management of large multi-domain distributed systems. Management of domains and entities are partitioned into groups based upon membership details. In distributed security, trust must be decentralized to support verification from multiple domain servers such as the privilege attribute service (PAS) in the SESAME architecture (Kaijser et al. 1994). Access control policies are specified in terms of domain membership rather than individual identities. Hence, the performance of the verification of domain membership can be critical for open systems. Similarly, other related work employing security agents on a per-node basis (Yialelis and Sloman 1996) has been used to support secure communication, rights delegation, and authentication.

Finally, we look at various *security ontologies* that can represent security information in an intelligible manner. Using semantic models such as RDFS, entities can interpret security information correctly.

Table 1 contrasts selected security standards. Public key infrastructure (or PKI) is currently the most widely available authentication and certification framework used by the industry. However, there exists numerous vendor-specific PKIs, such as the SET protocol PKI, Verisign PKI, etc. They are not compatible with each other because they use different types of certificates, use signatures in different ways, and they manage keys differently. Users and suppliers may also need to hold multiple certificates because related certificates themselves vary by certificate type, cipher type, version within a type, and the use of extension fields within a type.

In Table 1, semantic service adaptable specifications refer to XML-based representations that are easily convertible into richer semantic notations such as RDF/RDF-S. Security communication technologies such as S-HTTP 1999; SSL/TLS (1999); and IPSec (1998) are standards that have limited expressivity for representing semantically rich information. This hampers the propagation of lower-level security annotations and faults into

Q1 higher-level descriptions that support analysis. For example, low-level information can be captured, reasoned about, and shared amongst multiple systems in order to discern new threat patterns, share security configurations to promote interoperability, integrate standards from different consortia, and develop self-manageable and more adaptable autonomic systems.

In addition, an ontology supports multi-level abstractions and extensions to add new concepts using sub-classes and instances. The holistic security ontology does not aim to replace existing security standards; it represents a common framework for exchanging and representing security annotations of popular technologies such as SSL/TLS (1999) and IPSec (1998) using security profiles. Protocols belonging to these standards can be explicitly represented as workflow processes in service description languages.

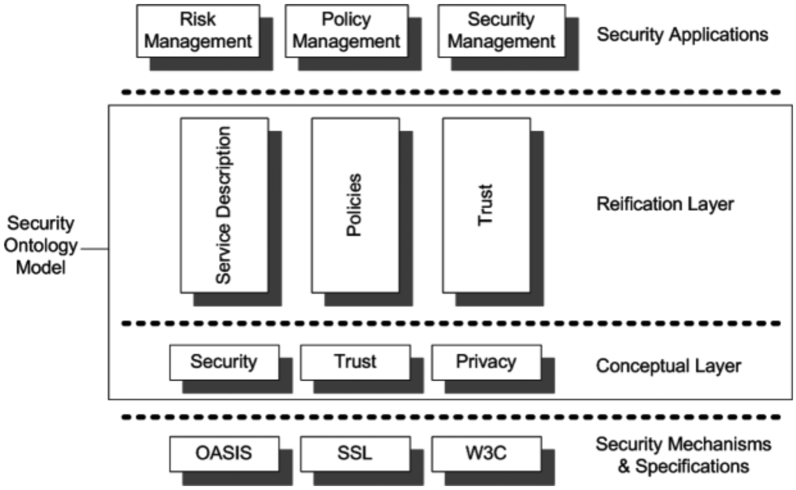
HOLISTIC SECURITY SEMANTIC MODEL

The development of a holistic ontology is not an end itself: It provides the means by which security services and software such as agents can advertise and exchange security-related information between them and it acts as a model for the management of security processes and services. In developing an ontological model, we wanted to specify an abstract model that wasn't directly dependent on particular security mechanisms or specifications, making the model more maintainable. The ontological model consists of:

- *Conceptual Layer*: Defines the properties and relations between security, trust, and privacy related concepts
- *Reification Layer*: Comprises the following sub-layers:
  - *Service Description Layer*: Provides the means for security processes to be hooked into service processes.
  - *Policy Layer*: Provides the means for defining security rules and constraints.
  - *Trust Layer*: Provides the means for defining trust implementations within systems to enable soft security interoperability between disparate applications. This is, however, not the focus of this paper and is not discussed further.

These are separated from:

- *Security Mechanisms*: Specific instances of security concepts, policies, and service entities as defined in existing security standards
- *Security Applications*: Commitments to use the security ontology within specific application domains



Q6 **FIGURE 2** Layered ontological security model.

This separation enables the security conceptual model to be made independent of the application requirements and the use of specific security mechanisms. In the following sections the ontological model is defined in more detail.

Conceptual Layer

At a very abstract level, security is modeled in terms of three main concepts: *safeguards* that protect the *assets* of value in a system against *threats* (SAT). Viewpoints are expressed using profiles that represent particular configurations of sets of instances of safeguards, assets, and threats. Hence, this model has been termed V-SAT (Poslad et al. 2003). The conceptual model represents two main relationships: safeguards protecting assets and threats attacking assets (Figure 3).

In more detail, an *asset* entity is referenced as an agent, service, or data resource that is offered within an open environment. A *safeguard* is an entity that serves to increase the protection of assets in the systems. Safeguards are modeled in terms of *credentials* such as a public key and a certificate, and *protocols* that define the set of safeguard *actions* such as exchange public keys, verify keys, and revoke keys. *Threats* trigger particular safeguards being statically or dynamically configured or reconfigured. Threats can also be expressed as security requirements contained in policies.

A viewpoint or *profile* defines reifications of sets of relationships between specific assets, safeguards, and threats. Profiles bind policies, applications, or enterprise operational constraints and also link to risk models. Profiles

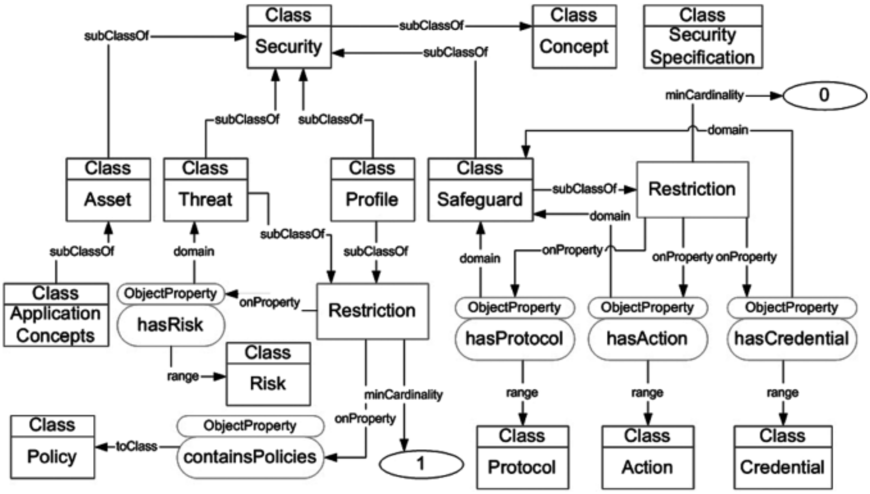


FIGURE 3 A fragment of the abstract security ontology.

are published using service description languages to provide service or platform security information across various domains hence publishing external security configuration information to support interoperability. Publishing external security configurations is considered to be one type of generic security service in our model. Dynamic security management is supported through retrieval and reasoning about profiles based upon the security ontology. Profiles support the management and selection of active versus inactive policies to ensure controllable secure environments within MAMD systems.

Concrete security concepts, such as RSAKey Value and signatures defined in the W3C, Oasis, and IETF public specifications, are linked to the abstract concepts by means of associating and instantiating instances of an action, protocol, and credential. In Figure 4, a signature is instantiated as part of a credential concept from Figure 3. Therefore, an abstract conceptual model is used as an upper ontology to mediate between concepts in different concrete specifications. A more complete description of the security ontology is also available in Poslad et al. (2003) and a normative version is available at Agentcities-QM (■).
Q2

Service Description Layer

Security is not just a set of security mechanisms to support a preventive approach against threats. It is also a dynamic process, whose use is triggered by changing threats and by the use of particular service operations. The assets in the security conceptual model are part of an actual application

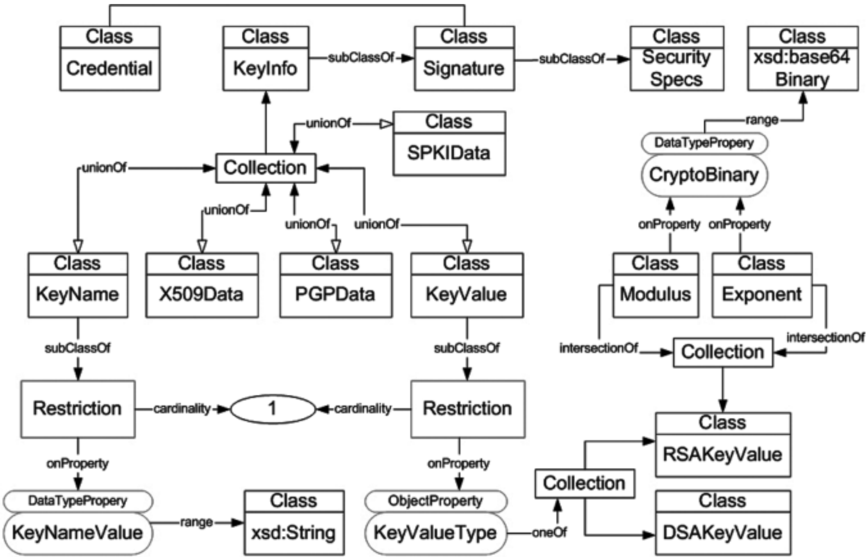


FIGURE 4 A fragment of the abstract security ontology and the link to specific security mechanisms.

and service process. For example, an asset such as a hotel room booking record is part of a hotel room reservation process. The reservation action in the process may trigger a different level of security being used to protect the exchange of customer payment credentials in order to reserve the room. Hence, assets must be modeled as part of service processes. In addition, the use and management of safeguards are also parts of security processes (modeled as *operations* in the conceptual model). Security and service processes need to be interlinked. Therefore, we need to interlink the semantics of the service concepts to the semantics of the security concepts. This interlinking is achieved using the service description layer.

As part of the complete methodology, the service description layer provides the means for modeling, managing, and advertising external security profiles. When an advertised service is specified, interaction with the service is represented using logic-based languages, such as FIPA-SL (2000) or KIF (1995), to formulate expressions for request or query agent communication protocol primitives. In order to formulate precise expressions, attributes pertaining to the slots of the concepts need occasionally to adhere to certain conditions or rules, for example, an instantiated data signing action concept with a rule defining the algorithm type (RSA-SHA1 or DSA-SHA1). Hence, the use of DAML-S pre-conditions is used to offer a more precise formulation of these rules. However, the current DAML-S pre-condition descriptions for specifying these input rules are vague and do not provide examples for integrating agent-based applications (D3.3 2003).

As a result, other constructs are used to express these conditions. These constructs are based upon introducing a string-based property in relation to a DAML-S condition (Poslad et al. 2003). Using a string-based property has its advantages; it provides the freedom for specifying the required condition of the process, and its proprietary representation. However, the use of structured semantic representation methods is encouraged to provide a more normative way of representing policies. Some grounding examples of utilizing the security ontology and an exemplary application ontology with DAML-S service descriptions are presented next. A grounding example demonstrates the integration and application of specific technologies in order to publish the security requirements (safeguards) for services (assets) within heterogeneous service environments. To begin the grounding, Figure 5 describes a service description defining the processes of a SET (Secure Electronic Transaction) dual signature (SET 2003) in an e-business secure transaction. The description defines the conference organizing agent (COA) maintaining data privacy when completing a restaurant booking without exposing payment information to the merchant and purchase information to the bank. Part of the restaurant booking service with a number of security processes expressed as functions of the booking service processes is presented in Figure 5.

In the example in Figure 5, three security processes are part of the restaurant booking service: certificates are exchanged for identity verification, session keys are used for communication, and PODS (Encrypted payment and order dual signature information) is requested for completing order transaction with a clearance house (e-bank).

However, the model aims at addressing these issues by providing a flexible, configurable, and open approach to security interoperability amongst multiple services. The model also provides a descriptive account of security instances and policies and components for managing interoperability and security. This allows systems to transparently adapt to changes in the environment, customize service execution according to requirements, and cope with threatening situations. In addition, the model is configurable so that systems can be adjusted to meet differing user requirements at the policy level, thus avoiding changes at the implementation level.

Figure 6 defines how the V-SAT model is utilized to support the scenario in Figure 1. An expression of a viewpoint constituting functional threats, safeguards, and assets is declared as a security profile. The secure payment safeguard expresses the action, credential, and protocol needed to complete a SET-based transaction between the COA, restaurant, and bank. In relation to this viewpoint, policies can be defined to govern the security conditions of the services. The policies are deliberated with the reasoning model in Tan and Poslad (2004) to achieve an interoperability consensus.

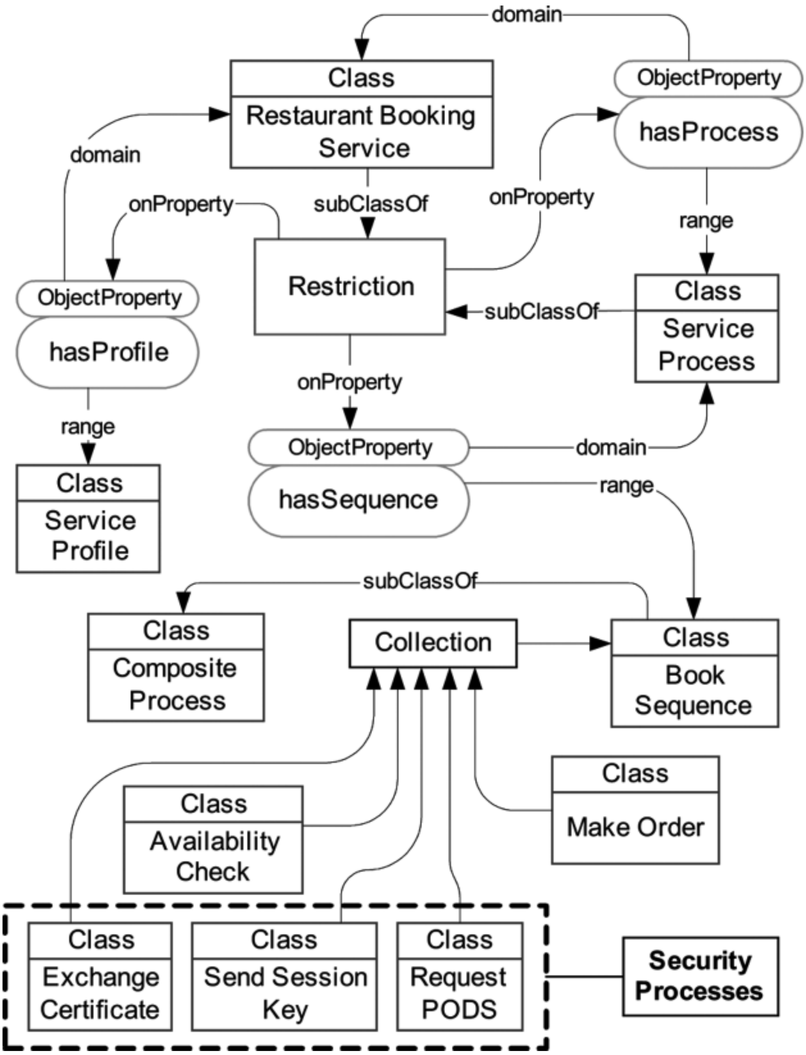


FIGURE 5 Restaurant booking service and security processes.

The service is published using directory services and can be discovered through agent queries and directory lookups. The security profile includes a description of the service to aid service selection and matching. The service workflow has a number of inputs required for invoking services; each process is managed and executed by the process logic application that enables the invocation of appropriate interaction messages. The process has several input parameters, defined by workflow variables in relation to the service and security ontologies. The services published in the system are presented by service profiles describing

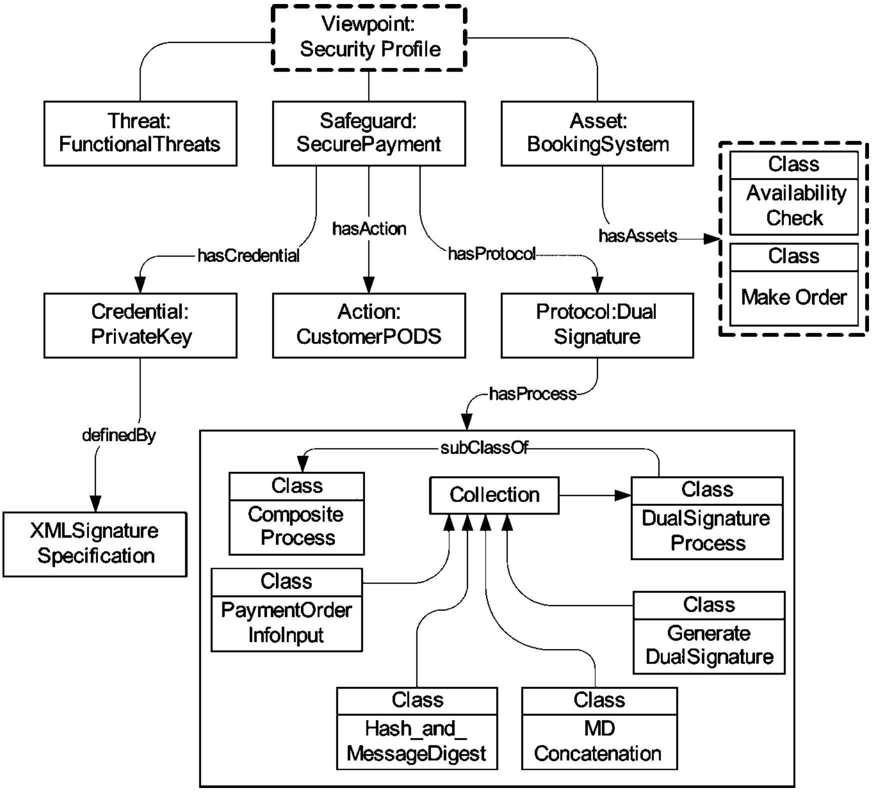


FIGURE 6 V-SAT and scenario.

processes, which also include relevant links to service ontologies. The workflow description involves one or more processing sequences, which specifies the detailed procedures and parameters when the service is invoked. The process description not only represents the service process model, but also interlinks security mechanisms to form secure processes. In reference to the scenario, the restaurant service security processes are built using DAML-S descriptions. The following sections give a more detailed account.

Exchange Certificate Process

The Exchange_Certificate security process is defined as a subclass of an atomic process that cannot be decomposed any further. It includes two communication streams: one is the certificate_in, which is the subproperty of the input stream, and the certificate_out, which is a subproperty of the output stream. Both these properties have a parameter of certificate type. It also includes a property of a policy concept, which

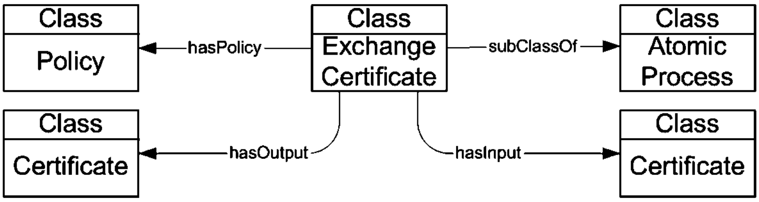


FIGURE 7 Exchange certificate process.

describes the constraints related to this process. Figure 7 defines the process.

Send Encrypted Session Key Process

The second security sub-process is the Send_En_SessionKey that is also a subclass of an atomic process. This process has an output property sessionKey_Out that is encrypted using the restaurant service's private key, and a policy property for specifying security constraints. Figure 8 illustrates the process.

Request Payment and Order Information Dual Signature Process

This process is a subClassOf atomic process; it has three input properties and a policy property:

1. *enpayment_In:* This property has an input parameter, Encrypted_Payment, which is the cipher text of the payment information.
2. *enorder_In:* This property has an input parameter, Encrypted_Order, which is the cipher text of the order information
3. *DS_In:* This property has an input parameter, Book_DualSignature, which can be generated by using payment information and order information based on the logic relationship defined in Figure 9.
4. *Policy:* This property specifies the security constrains of the request PODS process.

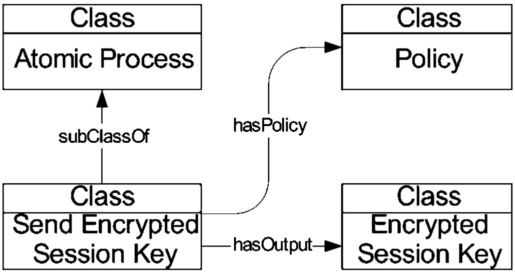


FIGURE 8 Send encrypted session key process.

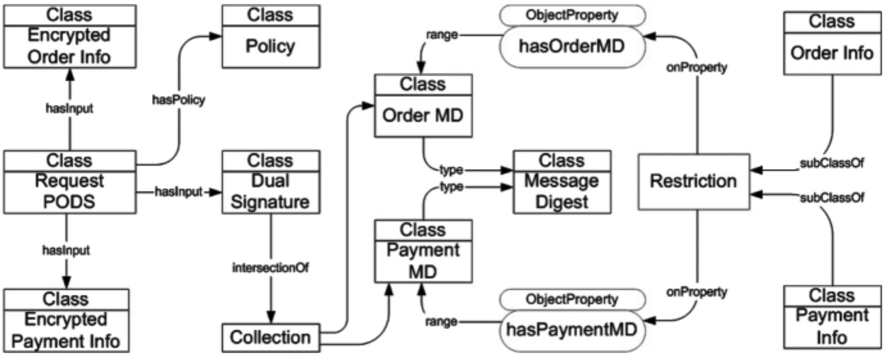
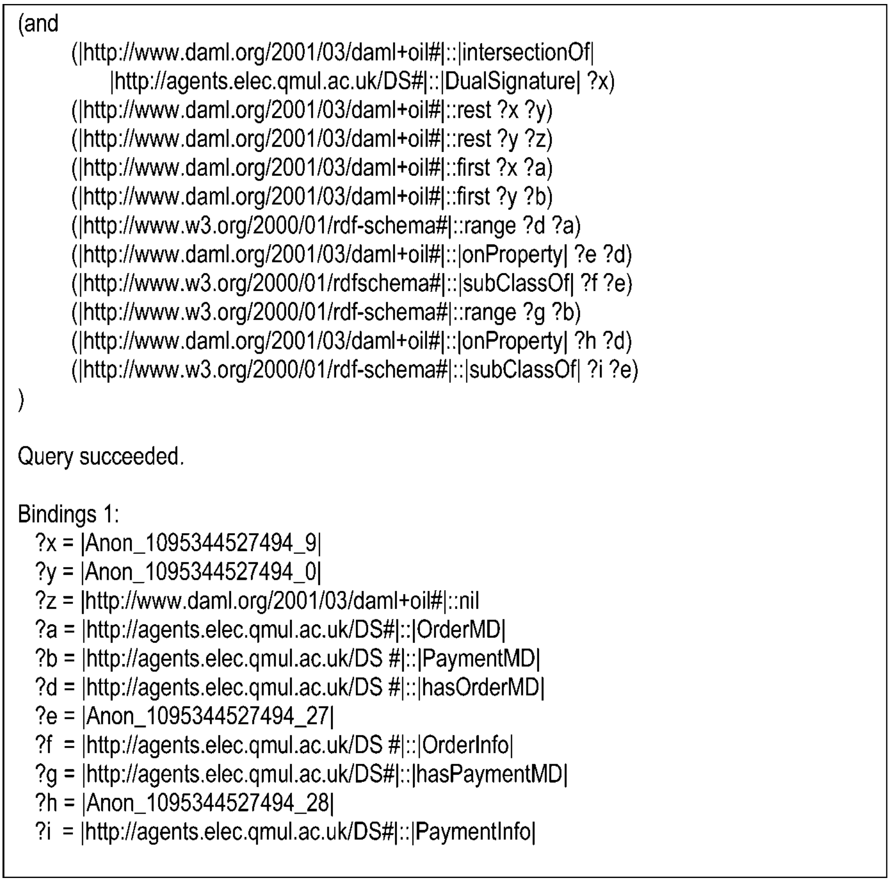


FIGURE 9 Request PODS process.

Ontology Reasoning for Interpreting Semantics

In service-driven environments, security processes have inputs and instances associated with policies that govern the operational security constraints. The policies better promote fine grain specification of rules that can support multiple combinations of detailed system requirements using logical operators. As a result, policies not only provide a flexible security process requirement specification, but also support adaptable and reconfigurable security rule selection using reasoning systems (Tan and Poslad 2004). Although the reasoning of policies can better promote decision making in support of security interoperability, it is still impeded by a lack of policy enforceability that gives no guarantee that users or services behave appropriately. Enforcement of policies is managed by two components: the business process logic (BPL) tool provides semantic verification of concepts necessary to the security process input, and an API that provides the invocation of suitable security mechanisms such as authentication and confidentiality in the operational model.

The inseparable unit of a process is an atomic process. Therefore, the BPL tool initially determines the resource type of the process and breaks it down into sequential processes, if necessary. If the sequence is a subclass of a composite process, the tool decomposes the process into smaller units and repeats this operation until the unit is an atomic process. Subsequently, the tool enumerates all properties supported by the atomic process. Finally, it checks the property type and deals with the parameters individually by meeting the property restrictions. Security policies associated with processes are dealt with in combination with the BPL tool; it is matched with policies governing the process requirements by triggering executions with the security API to determine if enumerated properties match. The API applies security mechanisms that adhere to the process input parameters and to the security policies to check the information correctness. In this



```
(and
        (|http://www.daml.org/2001/03/daml+oil#|::|intersectionOf|
              |http://agents.elec.qmul.ac.uk/DS#|::|DualSignature| ?x)
        (|http://www.daml.org/2001/03/daml+oil#|::rest ?x ?y)
        (|http://www.daml.org/2001/03/daml+oil#|::rest ?y ?z)
        (|http://www.daml.org/2001/03/daml+oil#|::first ?x ?a)
        (|http://www.daml.org/2001/03/daml+oil#|::first ?y ?b)
        (|http://www.w3.org/2000/01/rdf-schema#|::range ?d ?a)
        (|http://www.daml.org/2001/03/daml+oil#|::|onProperty| ?e ?d)
        (|http://www.w3.org/2000/01/rdfschema#|::|subClassOf| ?f ?e)
        (|http://www.w3.org/2000/01/rdf-schema#|::range ?g ?b)
        (|http://www.daml.org/2001/03/daml+oil#|::|onProperty| ?h ?d)
        (|http://www.w3.org/2000/01/rdf-schema#|::|subClassOf| ?i ?e)
)

Query succeeded.

Bindings 1:
  ?x = |Anon_1095344527494_9|
  ?y = |Anon_1095344527494_0|
  ?z = |http://www.daml.org/2001/03/daml+oil#|::nil
  ?a = |http://agents.elec.qmul.ac.uk/DS#|::|OrderMD|
  ?b = |http://agents.elec.qmul.ac.uk/DS #|::|PaymentMD|
  ?d = |http://agents.elec.qmul.ac.uk/DS #|::|hasOrderMD|
  ?e = |Anon_1095344527494_27|
  ?f  = |http://agents.elec.qmul.ac.uk/DS #|::|OrderInfo|
  ?g = |http://agents.elec.qmul.ac.uk/DS#|::|hasPaymentMD|
  ?h = |Anon_1095344527494_28|
  ?i  = |http://agents.elec.qmul.ac.uk/DS#|::|PaymentInfo|
```

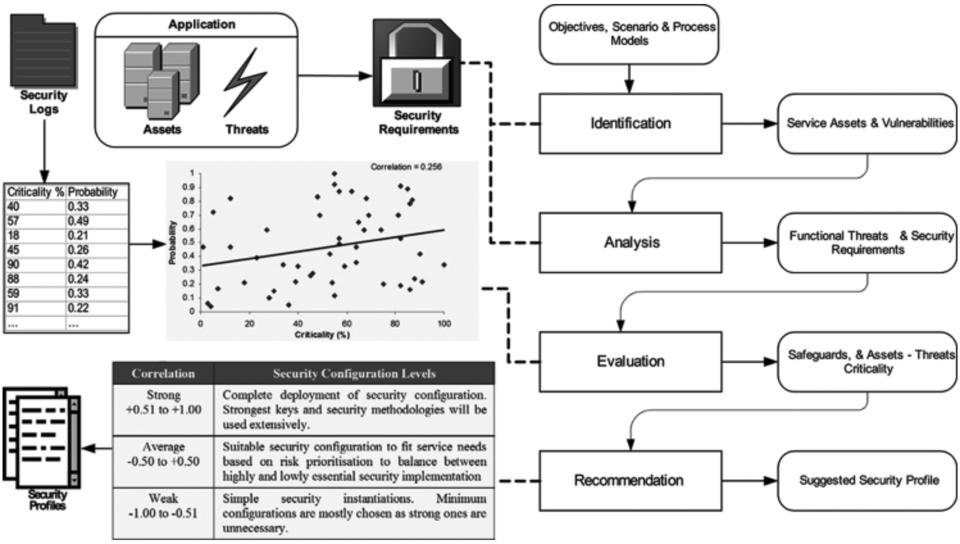
FIGURE 10 Ontology logic reasoning.

manner, the system is able to enforce policies by validating the inputs received against its service description and security policies. Figure 10 gives an example of the BPT tool identifying the property types needed to complete a dual signature input parameter. The dual signature concept is reasoned about using the Java theorem prover (JTP) for understanding its ontological bindings.

Consequently, these service descriptions (together with the security profile) are published using directory services. This description layer is supported by the policy layer, using substantiated facts and logical assertions.

Modeling Security Profiles Using a Risk-Based Approach

The security processes integrated with existing services utilize risk management (security applications) to support rational recommendations of security instances. As part of the security framework, a risk management model develops a qualitative and quantitative approach for advocating



Criticality %	Probability
40	0.33
57	0.49
18	0.21
45	0.26
90	0.42
88	0.24
59	0.33
91	0.22
...	...

Correlation	Security Configuration Levels
Strong +0.51 to +1.00	Complete deployment of security configuration. Strongest keys and security methodologies will be used extensively.
Average -0.50 to +0.50	Suitable security configuration to fit service needs based on risk prioritisation to balance between highly and lowly essential security implementation
Weak -1.00 to -0.51	Simple security instantiations. Minimum configurations are mostly chosen as strong ones are unnecessary.

FIGURE 11 Risk-based security profile recommendation.

the identification, analysis, and evaluation of the V-SAT model, supporting a rational recommendation of representing security profiles for a given service. The model encompasses a set of steps defining inputs and outputs related to the complete process.

In Figure 11, the service is analyzed by identifying functional threats affecting assets from use case scenarios. Based on the *criticality* and *probability*, a risk evaluation produces correlation data for assessing the security recommendation of a system. A scatter plot determines the correlation between impact *criticality* of threats onto processes and its occurrence *probability*. Plots are based on a number of vulnerable service processes within a system. Each service process defines *criticality* and *probability* values. The occurrence frequency *(probability)* is derived from a number of sources, such as security logs from existing systems and a Byzantine assessment of the environments. By utilizing these data, a graph plotting these denotations with a linear regression trend line defines the correlation. The value from –1.00 to + 1.00 describes if these two values are lowly or highly correlated. A highly correlated value suggests a strong security configuration and vice versa. For example, it is highly correlated if the impact *criticality* of a threat is *significant* and its occurrence *probability* is *nearly definite* (Table 2).

Policy Layer

Safeguards such as authentication, confidentiality, privacy, integrity, and Authorization are expressed using security profile(s), and are linked to a

TABLE 2 Criticality and Probability of Risk (Reference Definition)

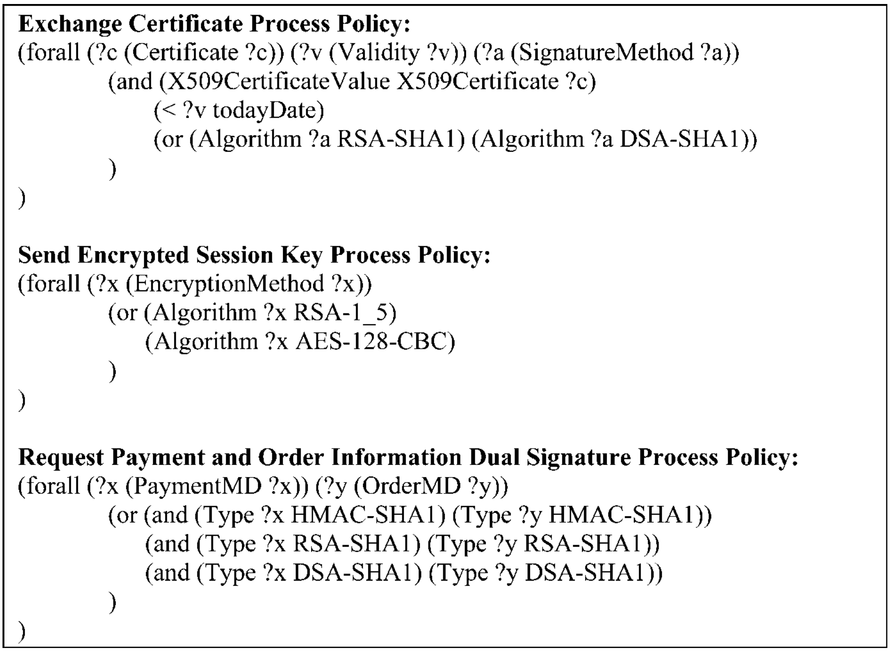
Criticality levels	Criticality definitions	Probability
Significant 76–100%	Significant costly loss of system assets or resources. The occurrence frequency is very high and threats are highly motivated and capable.	Nearly definite 0.81–1.00
Moderate 26–75%	Exercise of this threat may result in moderate to costly loss of assets or resources. Possible occurrence of threats; may need necessary safeguards as a prevention.	Possible 0.21–0.80
Trivial 0–25%	Exercise of this threat may result in minor losses. Occurrence levels can be low.	Doubtful 0.00–0.20

set of policies. The policy is defined as a security preference, requirement, or capability constraint related to the safeguards of application services. The profile specifies the relevant security information such as credentials and protocols (ontology) along with a logic-based description for expressing policies. In this policy layer, a reasoning engine is used to reason about the policies. The conceptual layer supports the policy layer by means of providing security facts. As a result, policies are applied to these facts using a reasoning engine to enable a rational decision to be made.

Based on the scenario, the restaurant service has a number of security processes defined previously. The invocation of these processes has service condition rules or policies defined in the V-SAT policy specification language (Agentcities-QM). This logic-based description language is based on knowledge interchange format (KIF) (KIF 1995) and can be easily assimilated into other knowledge representation connotations to support diversity. Security processes have input and output parameters based on the security instances defined in the security profile.

Input parameters have security policies that enable the customization of rule execution, according to the specific security process instance data. The mapping between security policies, processes, and instances is based upon the security and service ontologies. The input/output data policies are derived from service description "conditions," and the condition is specified as an "agent condition," a string representing the rules and preferences of the security instance. When an agent interacts with the service, it traverses the service description to deliberate about the security instances and policies. A reasoner is executed against security profiles stored in the knowledgebase repository (Tan and Poslad 2004). Figure 12 shows a sample specification of policies associated with input parameters related to the processes described previously and Figure 5.

The matching of policies within the reasoner is a daunting task. The idea of supporting policy negotiation for conflicting policies is potentially non-scalable and can introduce high overheads and complexity. Therefore,



```
Exchange Certificate Process Policy:
(forall (?c (Certificate ?c)) (?v (Validity ?v)) (?a (SignatureMethod ?a))
        (and (X509CertificateValue X509Certificate ?c)
            (< ?v todayDate)
            (or (Algorithm ?a RSA-SHA1) (Algorithm ?a DSA-SHA1))
        )
)

Send Encrypted Session Key Process Policy:
(forall (?x (EncryptionMethod ?x))
        (or (Algorithm ?x RSA-1_5)
           (Algorithm ?x AES-128-CBC)
        )
)

Request Payment and Order Information Dual Signature Process Policy:
(forall (?x (PaymentMD ?x)) (?y (OrderMD ?y))
        (or (and (Type ?x HMAC-SHA1) (Type ?y HMAC-SHA1))
           (and (Type ?x RSA-SHA1) (Type ?y RSA-SHA1))
           (and (Type ?x DSA-SHA1) (Type ?y DSA-SHA1))
        )
)
```

FIGURE 12 Example of *n-arity* security policies of the restaurant service.

to enable alternative solutions for policy conflict resolution, the introduction of alternative policies (*n-arity*) is defined in Figure 12, where optional algorithms are supported (Tan and Poslad 2004).

Consequently, security profiles are executed by the security application layer in which, policy, privacy, and cryptographic computation management are enforced.

Security Applications Layer

The security application layer is partitioned into multiple common management functionalities such as policy, security, and risk management applications. Each application independently provides management capabilities and may have interactions with one or more application entities. The management applications can be either used individually or collectively to provide greater management support. The policy management service provides policy editing, creation, and deletion functions. The security management supports credentials, cryptographic, and authentication services for establishing secure communication between systems. The risk management supports a qualitative and quantitative assessment of security threats within the system to offer rational security recommendations.

EVALUATION AND DISCUSSION

The profile-based security model has been specified and implemented in demonstrations of services, such as market places, event organizers, and e-banking agent systems, as part of the EU-funded Agentcities RTD project Q2 (Agentcities ■). An explicit ontology defining abstract concepts is specifically grounded with normative security specifications, such as SAML, XKMS, XML signature, and XML encryption, and implemented. The ontology provides viewpoints of associations between safeguards, assets, and threats (V-SAT) defined using profiles (Poslad et al. 2003). The Q5 ontology is available at Agentcities-QM (■). We present, in Figure 13 part of a directory entry describing an advertised service with references to the core and functional ontologies in an FIPA-based service application.

A reasoning application uses JTP and a security API to advocate both the reasoning of security policies and cryptographic computation between communicating entities. The API can be triggered to support authentication, confidentiality, integrity, key management, and key distribution services after high-level processing such as discovery, reasoning, and orchestration has been established. These security API services are available Q2 at Agentcities-QM (■).

Multiple service profiles can be registered. For example, a "secure tunneling" service can have the following sub-services (actions) such as "authentication" and "key exchange," where each action can be represented in a separate security profile to aid management. The use of multiple profiles is supported using ontology extensibility and this helps to maintain a loosely coupled relationship between profiles. As a result, classifying actions into profiles allows us to manage them collectively.

The methodology uses an abstract security semantic model, where conceptual representations of security entities are mapped onto explicit security specifications. This combines *generality* and *practicality* for expressing high-level security relationships of application scenarios that can also be explicitly defined. A set of core functional security requirements such as authentication, integrity, and confidentiality is defined using core

```
(df-agent-description
....
 :services (set (service-description
    :name Authentication :type Security
        :ontologies (set http://agents.elec.qmul.ac.uk/agentcities/security/AbstractSecurityOntology)
    :properties (set (property
       :name ExchangeCredentials
       :value http://agents.elec.qmul.ac.uk/agentcities/security/VenueCredentialProfile))))
.....)
```

FIGURE 13 A service advertisement showing the interlinked use of the security ontology.

```
<policy rdf:range="xsd:String">
   (exists (?a (Transform ?a)) (or (Algorithm ?a  http://www.w3.org/2000/09/xmldsig#rsa-sha1)
 (Algorithm ?a http://www.w3.org/2000/09/xmldsig#dsa-sha1) )
</policy>
```

FIGURE 14 An example policy for signature verification.

safeguards of the model. Further configurability is specified within actions, protocols, and credentials. The model also advocates extensibility, where specific safeguards are extended or incorporated into the ontology.

Security profiles representing the process, configuration, and instances of the system are constrained using specific policies. An example policy constraining the algorithm of the retrieval method in a signature is given in Figure 14.

The V-SAT policy specification language provides the normative descriptions for specifying policies within this framework. The language is modeled at an instance value level as opposed to a conceptual level. As a result, we are able to produce more specific and flexible representations for specifying policies but with the disadvantage of introducing a sizeable number of syntaxes.

An evaluation of the performance relating to the reasoning model developed using JTP and its ontologies and profiles written using DAML + OIL has been performed. The system is loosely coupled and is easily instantiated with existing semantic-based technologies such as Web or agent services. The model is bootstrapped either by a GUI or by instantiating Java method calls to the reasoning engine into existing applications, where ontology and profiles are loaded through URN(s). Some key issues for practical reasoning models are their performance and scalability for MAMD environments. The performance of loading facts and rules into the JTP is given in Figure 15.

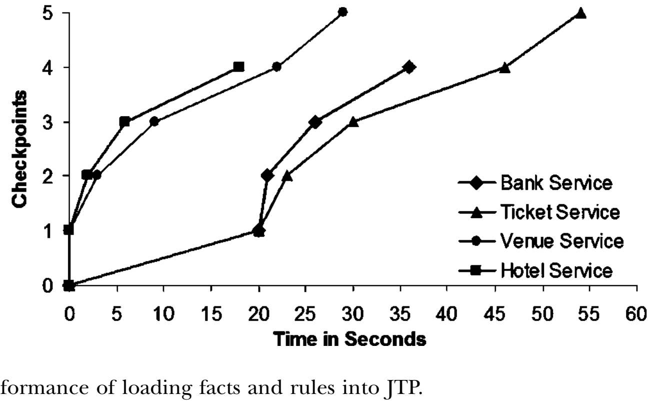


FIGURE 15 Performance of loading facts and rules into JTP.

The performance of the reasoning system was tested and takes approximately 36 seconds to completely establish its required knowledge base (KB) on a 1.5 GHz Pentium 4 notebook computer with 256MB of memory. The reasoning time is almost negligible but the computation complexity of loading facts and policies can be significant. The core ontology can be loaded in 20 seconds for around 70 facts and 60 seconds for around 250 facts. This is largely dependent on the size of the ontology. Therefore, the reasoner is partitioned into various checkpoints where knowledge is loaded at different intervals. Hence, it increases the system's performance where it can load or unload knowledge dynamically. In Figure 15, the four types of services are grouped into two composite services: hotel and venue services and bank and ticket services. N.B.: the hotel and venue service do not need to load the knowledge for checkpoint 1 if the bank and ticket service has already loaded their KB. JTP developers are currently working on performance improvements, which may affect these results. Consequently, the computation time for handling multi-system profiles is feasible.

Modeling Choice

The policy model supports reconfiguration of security requirements and conflict resolution using *n-arity* policies, and can be compared against other models such as Bradshaw et al. (1997), Corradi et al. (2001), and Kagal et al. (2003). These models are similar in the nature to policy management infrastructures that support access control mechanisms for domain-based environments. Multi-domain interoperability between disparate open architectures can be incorporated (Bradshaw et al. 1997), but the notion of open is constrained to membership domains where entities must be registered in advance. Policy models (Kagal et al. 2003) that define rules to support access control for domain-based security in RDF for the semantic Web are complementary to our model. The open MAMD model can be extended, in which segments or collections of domains can be implemented using policy management technologies expressed in Bradshaw et al. (1997) and Damianou et al. (2001). But, our system provides a holistic model for dynamically specifying security configurations between open systems to promote interoperability.

The abstract security ontology coupled with security specifications provides the model with the means for sharing knowledge between disparate services using a knowledge base. The argument that security configurations should not be revealed because advertising how the system is protected enables attackers to gain useful information to gain unauthorized access, the so-called security by obscurity, gives systems a false sense of security. Security by obscurity may make the initial attacks harder for the adversary but this may hide weaknesses that an open peer review might have revealed.

There is a tradeoff in analyzing MAMD system security in such an abstract way. The advantages of this kind of abstract reference model includes being insulated from popular particular technological security models that may become disused or frequently superseded and able to support heterogeneous application security requirements. The disadvantage is that an abstract model may appear to be too abstract, complex, and flexible to be used to specify concrete MAMD security systems for particular application requirements in practice. In order to minimize the disadvantages, a profile-based approach is used to map an abstract common view of security to particular application-oriented reifications of the model.

Having highly configurable systems can easily lead to bad configurations, thus the support for heterogeneity would need to follow an agreed semantics. Policies defined based on semantics of the security ontology, where the ontology provides a limited set of nouns to associate its facts as rules in policies, can provide additional support to cluster policies and to optimize management.

Viewpoints, Profiles, Contexts, and its Semantics

The security ontology model and its semantics define the sense behind the knowledge. Unfortunately, knowledge can be interpreted in different ways to make different sense from single or many disjoint situations. Subsequently, these situations can either be conflicting, contradictory, or not make any sense at all. Therefore, the concept of viewpoints (composite profiles), profiles (scenarios or situations), and contexts (meaning of the profile) provides an account for distinguishing the semantics of representing various similar instances of the common knowledge at different granularities. Figure 16 and its attached formula explain a normative example on the concept of viewpoints, profiles, and contexts to resolve its semantic anomalies.

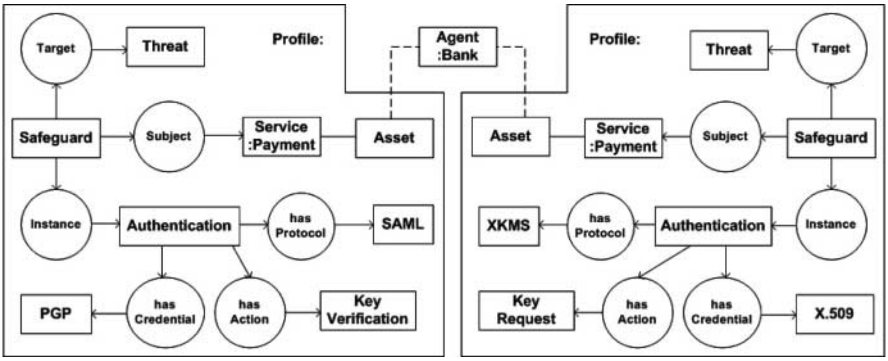


FIGURE 16 Conceptual graph of profiles constituting contexts.

In our definition, the security ontology represents a possible world model describing a wide range of stakeholders and entities. This representation expressed in ontology languages such as DAML + OIL is supported by model theoretic semantics, describing a formal account of the interpretations of legitimate expressions of the language. Therefore, the profile represented in Figure 16 defines the entities that are contained in the security world model, and that each collection of entities retains its identity over a period of time. These collections of entities could in this model be termed a context, where it brings together a collective view of a situation that is an instance of the world model to resolve a security requirement of a service. In the conceptual graph in Figure 16, safeguards are the protection relationships between subject assets and threat targets. A graph also constitutes collections (profiles) of asset, safeguard, and threat relationships, representing an application context for an agent actor.

The profile provides a rational bond between each disjointed entity within the security ontology. In this case, an agent could result in having many profiles that may become contradictory to one another and may result in conflicts (Sowa 2000). To avoid this contradiction, each profile can be represented as distinct contexts. To do this, profiles are considered meta but explicitly distinguished to define the contexts of each agent. These contexts can also be time-stamped to separate certain timelines to which they need to adhere. In Figure 16, we define an agent represented using multi profiles derived from the security ontology as contexts. The entities describing the context are distinct from other contexts and could be used conjunctively or disjunctively.

The two profiles described by the propositions in Figure 17 are nested inside the *dscr* predicates, and effectively define the explicit contexts. The descriptions inside those contexts refer to the agent *x*, which is quantified outside, but neither of the nested contexts can refer to or contradict any information in other contexts.

(∃*x*: Agent) (identity(*x*, Bank) ∧
(∃*p*: Profile)
(dscr(*p*, (∃*a*: Asset) (∃*w*: Service) (∃*s*: Safeguard) (∃*t*: Threat) (∃*y*: Authentication) (∃*f*: Protocol)
(∃*c*: Credential) (∃*d*: Action) ∧ subClassOf(*w*,*a*) ∧ subject(*s*, *w*) ∧ target(*s*, *t*) ∧ instance(*s*, *y*) ∧
hasProtocol(*y*, *f*) ∧ has Action(*y*, *d*) ∧ hasCredential(*y*, *c*) ∧ asset(*x*) ∧ service(Payment) ∧
credential(PGP) ∧ action(KeyVerification) ∧ protocol(SAML))) ∧
(∃*z*: Profile)
(dscr(*z*, (∃*a*: Asset) (∃*w*: Service) (∃*s*: Safeguard) (∃*t*: Threat) (∃*y*: Authentication) (∃*f*: Protocol)
(∃*c*: Credential) (∃*d*: Action) ∧ subClassOf(*w*,*a*) ∧ subject(*s*, *w*) ∧ target(*s*, *t*) ∧ instance(*s*, *y*) ∧
hasProtocol(*y*, *f*) ∧ has Action(*y*, *d*) ∧ hasCredential(*y*, *c*) ∧ asset(*x*) ∧ service(Payment) ∧
credential(X.509) ∧ action(KeyRequest) ∧ protocol(XKMS))))

FIGURE 17 Security profile formula of an agent.

Consequently, various contexts represent the different semantic meanings defining distinct security situations. These situations weave together a contemporaneous set of events to create collateral security viewpoints of the system. Viewpoints are a collection of many disjointed profiles and contexts that share a consistent rational binding between one another. Hence, viewpoints provide the meta-representation of composite profiles that describes the complete security operations of multi-systems in an MAMD environment.

Other Instantiations of the V-SAT Model

In Titkov et al. (this volume), user privacy for mobile information services is derived from the V-SAT model. The mappings between the concepts of the privacy framework and the V-SAT model are separated into three taxonomies: *user, service provider,* and *broker.* A *user* specifies personal information as assets and policies, whilst a *service provider* specifies tokens as credentials and the information request operation as a protocol, and finally, a *broker* acts as a mediator between the *user* and *service provider.* This is expressed in the V-SAT model as privacy safeguards constituting various actions (e.g., banking info request, user personal info request, etc.), credentials (e.g., X509, reputation, etc.), and protocols (P3P, APPEL). The users can specify preference and security instance policies using security profiles. The reasoning model supports the mapping of security instances to determine if the service provider and user security settings match. The reasoning of preference policies determines the result of the request by informing an *accept, reject,* or *notify* to the service provider.

In Ricci et al. (this volume), the agent coordination context (ACC) defines a first class abstraction for modeling agent environment and interaction in theory. In practice, the ACC is useful for modeling certain security aspects of the V-SAT model to support authorization in multi domains and services contract formation. For authorization, the ACC is explicit yet an abstraction for modeling roles in heterogeneous environments where access control policies can be flexibly administered and dynamically activated for organizing complex separation of duty amongst multiple agents. In secure contract formation, ACC distinguishes the agent role associated with the organization along with a representation of contract states, notion of time, and protocol information. This is particularly useful when monitoring complex contracts to identify failures at different states for promoting safer fault tolerant environments.

In Sonntag (this volume), a trusted gateway agent is used by the internal system for managing interaction with external environments. This can also serve as a type of security provider by instantiating the system in synchrony with the V-SAT model. The use of security profiles enables system developers to specify explicit service requirements governing the business process logic such as inputs, outputs, and policies. By means of reasoning and process validation, internal agents can be protected by the gateway agent controlling the interaction and supporting interoperability between multiple domains.

CONCLUSION

A common security model has been developed to address some requirements and challenges to aid automatic and dynamic configuration of security and the interlinking of security to service process for use in open heterogeneous environments. The challenges include: the lack of a holistic model to allow existing specific specifications to be combined and upgraded; the lack of an explicit rich enough set of meta-data in order to define the semantics to correctly use security mechanisms; and the lack of a semantic meta-data model in order to dynamically interlink security and service processes. An abstract semantic security model expressed in DAML + OIL ontology has been created. It has been demonstrated that it can link to existing standard security mechanisms; a service layer is used to interlink service and security processes and a profile-based model is combined with a policy layer and a reasoning model to support the dynamic reconfiguration of security and to support security interoperability.

As part of ongoing work, we plan to improve this model to support other industry standard semantic representation languages such as *WSCI* and *ebXML*. The aim is to provide a holistic approach for the discovery of security profiles for supporting interoperability between heterogeneous systems. In addition, there are plans to include trust description models and concepts to support interoperability between heterogeneous secure and trusted domains. As a result, open services can benefit from policy-based environments that are dynamic and manageable through consistent viewpoints or profiles.

REFERENCES

Agentcities D3.3: Dynamic Value Creation in Agent based Environments, http://www.agentcities.org/EURTD/

Agentcities. RTD: Global Agent Testbed, http://www.agentcities.org/

Agentcities. RTD: Queen Mary Agentcity, http://agents.elec.qmul.ac.uk/agentcities/security/

Blaze, M., J. Feigenbaum, and J. Lacy. 1996. Role of trust management in distributed systems security. In *Proceedings of the IEEE Conference on Security and Privacy*, pages 164–173. Oakland, California, USA.

BPEL4WS + WSDL, http://www-106.ibm. com/developerworks/webservices/library/ws-bpel/

Bradshaw, J., S. Dutfield, P. Benoit, and J. D. Woolley. 1997. KAoS: Toward an industrial-strength generic agent architecture. In: *Software Agents*, ed. J. M. Bradshaw, 375–418. Cambridge, MA:AAAI/MIT Press.

Burrows, M., M. Abadi, and R. Needham. 1990. A logic of authentication. *ACM Transactions on Computer Systems* 8(1):18–36.

Corradi, A., N. Dulay, R. Montanari, and C. Stefanelli. 2001. Policy-driven management of agent systems. In: *Policies for Distributed Systems and Networks*, eds. M. Sloman, J. Lobo, and E. Lupu, 214–229. Berlin: Springer-Velag.

Damianou, N., N. Dulay, E. Lupu, and M. Sloman. 2001. The ponder policy specification language. In: *Policies for Distributed Systems and Networks*, eds. M. Sloman, J. Lobo, and E. Lupu, 18–38. Berlin: Springer-Verlag.

DAML Services, http://www.daml.org/services/owl-s/

Denker, G., L. Kagal, T. Finin, M. Paolucci, and K. Sycara. 2003. Security for DAML web services: Annotation and matchmaking. In: *The Semantic Web – ISWC 2003*, eds. D. Fensel, K. Sycara, and J. Mylopoulos, LNCS 2870, 335–350. Berlin: Springer-Verlag.

Department of Defense. 1985. *Trusted Computer System Evaluation Criteria*, DOD 5200.28-STD.

Dierks, T. and C. Allen. 1999. *The TLS Protocol Version 1.0*, http://rfc.net/rfc2246.html

FIPA-SL, http://www.fipa.org/specs/fipa00008/SC00008I.html

Grandison, T. and M. Sloman. 2000. A survey of trust in Internet applications. *IEEE Communications Surveys & Tutorials* 3(4):2–16.

Herzberg, A., Y. Mass, J. Michaeli, and Y. Ravid. 2000. Access control meets public key infrastructure, or Assigning roles to strangers. In *Proceedings of the IEEE Symposium on Security and Privacy*, pages 2–14. Q3 ■, California, USA.

Kagal, L., Y. Finin, and A. Joshi. 2003. A policy based approach to security for the semantic web. In: *The Semantic Web – ISWC 2003*, eds. D. Fensel, K. Sycara, and J. Mylopoulos, LNCS 2870, 402–418. Berlin: Springer-Verlag.

Kaijser, P., T. Parker, and D. Pinkas. 1994. SESAME: The solution to security for open distributed systems. *Computer Communications* 17(7):501–518.

Kawatsura, Y. 2003. *Secure Electronic Transaction (SET) Supplement for the v1.0 Internet Open Trading Protocol (IOTP)*, http://rfc.net/rfc3538.html

Knowledge Interchange Format, http://logic.stanford.edu/kif/specification.html

Q2 Miller, J., P. Resnick, and D. Singer. *PICS Rating Services and Systems*, http://www.w3c.org/TR/REC-PICS-services.

Poslad, S., J. J. Tan, and L. Titkov. 2003. *Agentcities D3.4: Harmonising Heterogeneous Security Models*, http://www.agentcities.org/

Rescorla, E. and A. Schiffman. 1999. *The Secure Hypertext Transfer Protocol*, http://rfc.net/rfc2660.html.

Ricci, A., M. Viroli, and A. Omicini. 2006. Agent coordination context: From theory to practice. *Journal* Q4 *of Applied Artificial Intelligence* 20(2–3): ■

Sonntag, M. 2006. Multi agent systems as Web service providers. *Journal of Applied Artificial Intelligence* Q4 20(2–3): ■

Sowa, J. F. 2000. *Knowledge Representation: Logical, Computational, and Philosophical Foundations*, Thomson Q3 Learning.

Tan, J. J. and S. Poslad. 2004. Dynamic security reconfiguration for the semantic web. *Journal on Engineering Applications of Artificial Intelligence*: Special Issue on Autonomic Computing and Automation 17(7):783–797.

Thayer, R., N. Doraswamy, and R. Glenn. 1998. *IP Security Document Roadmap*, http://rfc.net/rfc2411.html

Titkov, L., S. Poslad, and J. J. Tan. 2006. An integrated approach to user-centered privacy for mobile Q4 information services. *Journal of Applied Artificial Intelligence* 20(2–3): ■

WSCI, http://www.w3.org/TR/wsci/

Yialelis, N. and M. Sloman. 1996. A security framework supporting domain based access control in distributed systems. In *Proceedings of the Symposium on Network and Distributed System Security*, pages 26–39. San Diego, California, USA.