

# Trust and Reputation Policy-Based Mechanisms for Self-protection in Autonomic Communications

Martin Serrano<sup>1</sup>, Sven van der Meer<sup>1</sup>, John Strassner<sup>1</sup>, Stefano De Paoli<sup>2</sup>,  
Aphra Kerr<sup>2</sup>, and Cristiano Storni<sup>3</sup>

<sup>1</sup> Waterford Institute of Technology, Telecommunications Software and Systems Group,  
ArcLabs Ireland, West Campus, Waterford Co., Ireland  
{jmserrano, vdmeer, jstrassner}@tssg.org

<sup>2</sup> National University of Ireland, Maynooth, Sociology Department  
Maynooth Campus, Maynooth, Ireland  
{stefano.depaoli, aphra.kerr}@nuim.ie

<sup>3</sup> University of Limerick, Department of Computer Science and Information Systems  
Interaction Design Centre, Limerick, Ireland  
cristiano.storni@ul.ie

**Abstract.** Currently, there is an increasing tendency to migrate the management of communications and information systems onto the Web. This is making many traditional service support models obsolete. In addition, current security mechanisms are not sufficiently robust to protect each management system and/or subsystem from web-based intrusions, malware, and hacking attacks. This paper presents research challenges in autonomic management to provide self-protection mechanisms and tools by using trust and reputation concepts based on policy-based management to decentralize management decisions. This work also uses user-based reputation mechanisms to help enforce trust management in pervasive and communications services. The scope of this research is founded in social models, where the application of trust and reputation applied in communication systems helps detect potential users as well as hackers attempting to corrupt management operations and services. These so-called “cheating services” act as “attacks”, altering the performance and the security in communication systems by consumption of computing or network resources unnecessarily.

**Keywords:** Trust Management, Pervasive Services, Policy-Based Management, Autonomic Communications, Pervasive Computing, Reputation Mechanisms, Systems Management, Social Networks, Information Systems.

## 1 Introduction

Social relationships are built based on the trust between people. Computing and communications systems are now aiming to take advantage of such models and then use the concepts of reputation and trust to, for example, generate systems offering trustworthy and secure information services and networking applications. Such systems, as trust generators, can also be used to support diverse applications in other

systems or sub-systems requiring certain security levels. In computing, trust management arise from the necessity to remotely execute operations, and has been adopted as a way to enable security for distributed systems in situations where risk taking management decisions exists. Hence, trust management systems must offer certain guarantees to securing information, as well as processes that create, manage distribute, and govern information and services, in a reliable and efficient manner.

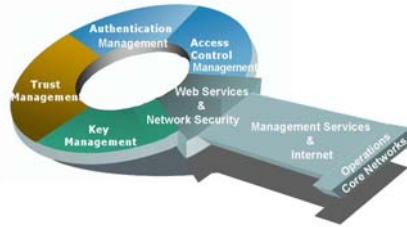
Trust management [1] is based on a philosophy of decentralizing security decisions, and as consequence of this, the creation of open and decentralized systems and stable and secure services [2] are promoted. In current service management systems and Future Internet solutions is crucial to protect the system and its sub-systems. Actually, there are several initiatives focused on specifying how to build open, distributed and secure management systems. The NGOSS, or New Generation Operations Systems and Software from the Tele-Management Forum (TMF) [3], attempts to standardize the processes and data used by Business and Operations Support Systems (BSSs and OSSs) for example. However, even ambitious initiatives such as this have failed to produce information models that are able to provide trust management and reputation services. Without a standard definition of such concepts, vendors will build their own device- and application-specific data models that will redefine common concepts. As information and management of communications systems migrating onto the web, the adoption of trust management practices is crucial to protect information and processes that have an inherent risk associated with it.

The use of a services-oriented philosophy helps this problem, and enables service support models to evolve and meet the needs of new applications that incorporate new technologies. The DEN-ng information model [4][5][6] was built using many different abstractions following this philosophy, and forms the basis for the work presented in this paper. However, the development of a robust information-centric view is only one part of the solution. The evolution of current security mechanisms in systems is not sufficiently effective to protect each management system and/or subsystem from intrusions or hacking attacks, especially if web-based operation is desired. This requires dedicated trust management protocols, formats, applications, and tools. In addition, trust management plays, more than ever, an important role in the design of any system and the interfaces with the user(s).

Communications and Internet systems have not yet sufficiently addressed the importance of the *social* needs that users and the adaptation of their social models have in computing systems, however some initiatives following this translation between domains exist. A clear example of such translation of models can be found in bio-inspired systems, where biological reactions from the human body or animals are studied and implemented as computing mechanisms emulating such behavior [7]. Another example is the operation of social networks, where features and human behaviors are implemented in communications networks and systems. So, the awareness of such social models and the necessity of using them to provide an immersive environment generating trust in computing systems is required.

The capture of such models and its adequate translation and implementation into computing environments has acquired more attention in the trust management community. Trust management is broadly accepted as required for modeling, analyzing, and managing decisions within certain trust levels [8][9]. This paper presents an approach, rooted in the management of pervasive systems, where autonomic

management is shown to be a promising approach to implementing self-protection. Figure 1 depicts our vision about typical web service security in the left-hand side of the picture supporting management communications. Our approach specifically examines how to support services and network security in pervasive services and the Future Internet. This trust management approach concentrates on management services and applications using an autonomic orientation.



**Fig. 1.** Management of Services to Enable Trust in Future Internet using Autonomic Technologies

This paper discusses a methodology that can be used in the framework of trust management to create solutions using reputation mechanisms based on policies. This approach can then apply this knowledge to support dynamic management of pervasive services. The reputation mechanism proposed follows social networks and other user-based reputation management systems principles [10][11]. The shortcomings of such systems, in terms of multi-criteria analysis and evaluation as well as implementation and realization experiences, are addressed in this paper, with the objective to illustrate how this research activity can develop new solutions that satisfy the important real world requirements of using multi-criteria for computing appropriate levels of reputation and trust using policy-based management mechanisms.

The rest of the paper is organized as follows. Section 2 describes related work for offering efficient and secure service deployment using trust management operations. Section 3 briefly describes the interaction between users and systems, and then introduces trust management based on reputation models that describe its conceptual relationship with the policy-based management paradigm. Section 4 introduces our policy-based trust model in services support as well as in pervasive management operations. Section 5 introduces a scenario in which policies and the trust and reputation model proposed are used for validation purposes. Finally, section 6 summarizes the contributions and conclusions of this paper.

## 2 Related Work

Participative user design [12] has been strongly influenced by ubiquitous computing, which is in turn motivated by developing systems with the ability to incorporate surrounding information about users and the environment, and to use such information to perform operations as described in [13][14][15]. However, these efforts do not usually include the use of trust management concepts.

There are a significant number of approaches that use social models for defining secure interactions between users and computational systems [16][17]. Recently, in the field of management services, user behavior has been described and introduced for taking control of specific management operations in networks and the systems [18]. Other approaches explore the translation of human behavior using social models, which enables systems to control services in a more secure and transparent manner [19]. An example is the NetTrust project [20] that uses a value-sensitive design mechanism to validate trust levels, particularly for e-commerce applications.

This approach can be applied to many other systems, such as [21] and [22]. Our research extends these approaches and concentrates on the task of supporting trust and reliable management service operations. Trust management is crucial in the deployment of pervasive secure services and their dynamic management nature allows delegating decision-making. Trust management helps to generate reliable management systems supported by self-protection and autonomic mechanisms.

Approaches for managing trust can be categorized in two major fields, as classified in [23]. However, when related to the field of autonomic communications, both major fields are in some way complementary each other. Today, with most of the services tendency towards a service-based design, the development of trust management follows a more integrated perspective for managing trust, and focuses on providing security and reliability about and for an entity. The use of policies to address this challenge in trust management is relevant; however, policies traditionally manage the decisions of a system for controlling specific set of operations that are pre-defined or pre-programmed. Policies can assist in making decisions when a certain level of ambiguity in the decision-making mechanism is present by utilizing the results of trust management systems.

Policies, as a tool for managing networks and services, have promoted a number of approaches for controlling such operations [24]. The main policy models used in network management are: 1) the IETF policy model [25][26], 2) the DMTF CIM [27], 3) the TMF SID [28], and 4) the DEN-ng in ACF [6]. We use the DEN-ng model because of the reasons documented in [6]. Conceptually, the semantics of a DEN-ng policy rule are: WHEN a set of events *triggers* the evaluation of a set of conditions, IF those conditions evaluate to TRUE, THEN execute a set of actions. Optionally, a set of alternative actions can be executed if the evaluation of the condition is FALSE. Our research uses policies as the mechanism to produce dynamic control and changes in management, orchestration of services, and performance of systems [5][6][29].

### 3 Trust and Reputation Model

Our current research is based on extending our policy representation to enable it to be used in trust management scenarios. This novel research task relates two different application fields – context awareness and trust management - by using contextual information from social models to determine reputation and trust values that can then guide the deployment of service offerings.

In pseudo-code, these mapping relationships are as follow:

**The Social Model** ... (1)

**WHEN** an *User* is requesting a service,  
**IF** an *Evaluator* evaluates *User* can be **TRUSTED**,  
**THEN** allows to *Execute* activities,  
**ELSE** apply *Restrictive* actions.

**The Trust Policy-Based Model** ... (2)

**WHEN** an *event\_clause* from a *User* is received,  
 (which is able to trigger a *condition\_clause* evaluation)  
**IF** a *condition\_clause* evaluates to **TRUE**,  
 (subject to the evaluation strategy)  
**THEN** execute one or more *actions*,  
 (subject to the rule execution strategy)  
**ELSE** execute alternative one or more *actions*,  
 (subject to the rule execution strategy)

An *event\_clause* specifies the event or set of events that trigger the evaluation of the *condition\_clause* of the policy rule. A *condition\_clause* evaluates the condition or set of conditions in order to determine which, if any, of the set of actions should be executed in response to the triggering event(s). An *action\_clause* specifies the set of actions to be executed if the result of the *condition\_clause* evaluates to **TRUE** (optionally, a second *action\_clause* can be defined to specify the set of actions to be executed if the *condition\_clause* evaluates to **FALSE**). In our policy model, the concept of restrictions arising from a lack of trust is also represented as actions.

Previous work has modeled an enhanced version of role-based access control using the DEN-ng policy model [30]. This work enables us to extend the DEN-ng policy model to include trust management concepts. Another important feature of our previous work is the concept of the Policy Continuum [4][31]. This is an abstraction that enables different concepts and terminology to be used to define policies for different constituencies (e.g., business users, architects, and programmers), and relates these policies through a set of transformations. This is an important tool to represent multi-criteria decisions and different levels of trust, in which different criteria and trust concepts for different constituencies can be related to each other.

The users of future communication systems should be able to interact with systems more freely, and should be able to configure their own services according to personal preferences and needs. This has the unfortunate side effect of encouraging a larger number of malicious users to try to cheat or possibly disable the system. For these and other reasons, it is important to create mechanisms that help detect such attacks and differentiate between trusted and malicious users. To do so, we propose to use policies that incorporate trust and reputation mechanisms.

Reputation mechanisms can be considered to be a subset of trust management systems that assign a computable measure of trust to any entity on the basis of the past history of that entity. The existing approaches in reputation-based trust management systems are not very different from that of social scientists. For example, [32] observed that participants in a trust relationship thought of trust as follows: "We wish to know the sort of person we are dealing with before we deal with him. But we will

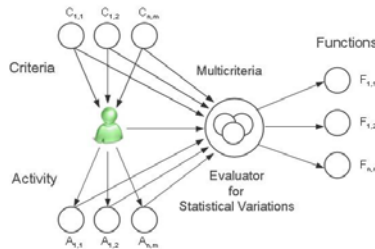
know it only imperfectly”. For this reason, [33] concluded: “Prior to the Internet, such questions were answered, in part, through personal and corporate reputations.

Vendors provided references, Better Business Bureaus tallied complaints, and past personal experience and person-to-person gossip told you on whom you could rely and on whom you could not.” The point made by [34] is that questions and concerns related to the trustworthiness and reliability of users and other entities over the Internet seems to be more challenging than in off-line relationships. The issue is therefore how to know the past behavior of an entity, how to compute and incorporate that behavior into calculations to define current trust levels, and then decide which rules to use to determine whether to trust or not such an entity.

The trust model is based on certain levels of reputation and, as it happens in real life, people trust in other people. However, there are questions that arise, including (1) what kind of reputation levels are necessary to evaluate the trustworthiness of a user, (2) what criteria must be considered when a trust value of the reputation of the users is assigned, and (3) what happens if trust depends on a combination of several criteria.

The mechanisms based on reputation and trust, for example, have been broadly used in many and diverse on-line sales and auction models, such as eBay, Amazon and Expansys. In this context, when a buyer wants to purchase a product, the buyer must make a decision based on the description and reputation of the seller. Most of the time, this is defined as a percentage, being the trust level average assigned by other buyers about the sellers. However, what happens if the seller is new to the market and does not have an established reputation? How can a buyer determine if sellers are cheating buyers by evaluating themselves positively? Hence, the final decision in trust is delegated to the buyers, mainly because there are not efficient mechanisms implemented that, based on trust or reputation can help the users and/or the systems to make such decisions. We use policies to control service operations and activities based on statistical variations that indicate untrusted operations in the system and then apply actions as restrictions. We evaluate the individual reputation values and the combined values (reputation values with statistical variations) to provide a more accurate trust value; this is used by policy-based trust management mechanisms to offer more reliable decisions to the systems based on the revised reputation values.

Multi-criteria is an important issue in trust management. We address this issue by using policy-based management. Figure 2 represents multi-criteria being used to



**Fig. 2.** Policy-Based Approach for Solving Multi-Criteria Problems

define the trust level being assigned to users. An evaluator uses reputation and statistical variations to evaluate and control if the user is trusted or not. If the user is determined to be trusted, then that user is allowed to execute authorized function(s).

The functional component acting as an evaluator in Figure 2 has the capability to analyze and provide actions as a result of this analysis. For example, when management applications must perform network changes, policy rules can take into account static as well as dynamic end user criteria and activities. Examples of criteria are user identity and electronic keys, while examples of activity are the statistical results such as visiting specific sites (e.g., eBay or Amazon) or contracts with services (e.g., phone and broadband services). Thus, users will be able to modify and execute system's actions according reputation performance, which is enforced using policies.

### 3.1 Premises in Trust Management

Typical approaches for security in the systems are based on passwords or key codes, and these techniques work well on closed systems. However, when decentralized security systems are used, public key infrastructure (PKI) mechanisms [35] emerge as more suitable solutions that provide decentralized and more secure models. Cryptography and digital certificates [36] are now used in many security solutions. Digital certificates ensure with an increased level of security, the transfer of information. The challenge is to decide who can access what type of information.

Sociological research [37] views trust as a relationship between individuals (for example between persons) or collective (for example Nation States) social actors. Trust relationships enable social actors to take decisions that have some amount of risk, in a situation in which there is a lack of knowledge and the possibility to make an informed choice is precluded. This means that if we look at the pseudo code describing the social model in Section 2 above, in real social situations, the roles of *User* and *Evaluator* are filled by appropriate social actors (i.e., by persons). In sociological literature, these roles are called the *Trustor* (the entity who places trust, the *Evaluator*) and the *Trustee* (the actor which receives trust, the *User*). When trust is placed in the *Trustee*, the *Trustor* is able to solve uncertain situations by choosing one of several alternatives, each based on trust and reputation. An example is when a user (*Trustor*) wants to use an on-line banking system (*Trustee*). How can the user be sure that the online banking system, which represents a bank, is a reliable banking system? The user cannot know this beforehand, because the user cannot prove that what he or she thinks is a banking system is not instead a forgery. Here is when a trust relationship must be established.

In this case, the recommendation of other people, based on previous experiences, and/or other evidence, can be used to establish a trust relationship. This can then be tested by, for example, conducting a small banking transaction and using the online system and then verifying the correctness of that transaction by physically visiting the bank. In other words, a trust relationship involves risk as well as uncertainty.

The same model applied to Trust Management in autonomic computing solutions needs to take in account that the *Evaluator* (or *Trustor*) in a trust relationship is a Trusted System (i.e., a machine making decisions of behalf of human beings and controlling their actions). Therefore, in Trust Management, the decisions of users to

trust systems are delegated to the Trusted System [38]. Along this line of reasoning, it is important to recognize that the enacted evaluation strategy satisfying the IF condition is never a neutral one.

Security policies and the mechanisms enforcing them might, for example, determine that there are unnecessary divisions of labor or unwanted social discriminations (e.g., gender, racial or age discriminations) in relation to how information is accessed. For these and other reasons, systems must be assisted in making decisions for determining the trustworthiness of users as well as in detecting cheating users. In addition, these systems must also be evaluated in their actions. The systems must take appropriate actions, such as restricting operations and/or blocking some or all user activities that could be performed. The service management tasks can then offer more dynamic performance and more efficient operation.

### 3.2 Methodology and Formal Approach

In this section, we describe the general concepts of the trust and reputation mechanism that uses policies to evaluate and define user capabilities to create, use and deploy web-based and Internet services. The reason to use policies is founded in the benefits that policies provide when they are used to control pervasive services [4][39]. We implement secure policies using an autonomic solution approach.

We start from the premise that a user can create, configure and personalize services according to personal requirements and/or needs. For example, Joe wants a broadband service for downloading video on demand on weekends, but a simpler and more cost-effective data service for checking email on weekdays. Key identifiers are created when the service is deployed and are associated (one to him and one to the service). The lowest rate of reputation is assigned to Joe, since he is the creator of the service. However Joe can be a user of other services; this is an activity that can be monitored and studied by systems to adjust Joe's reputation level. Thus, Joe's reputation increases when he uses other services in a reliable manner. This is an automatic way to adjust the reputation value of Joe. However, it does not adjust the trust level of Joe's service – this requires other people to be able to use that service reliably. In fact, it may be that to compute a trust level for Joe's service, multiple criteria must be considered, studied and evaluated.

To increase the trust level, multiple criteria must be considered, studied and evaluated. For example, if Joe provides personal information, or if Joe uses a key identifier already assigned from a trusted source, then those criteria can be used to provide statistics for defining a trust level for Joe. Hence:

$$\text{Criteria } C_{n,m} + \text{Activity } A_{n,m} \rightarrow \text{Functions } F_{n,m} \quad \dots(3)$$

A formal way to capture and study the criteria and activity is to relate them in a matrix, with elements as rows and columns containing first order logic values to categorize and classify the user(s). In other words, if the element in the matrix exists, the value assigned will be 1; if the element in the matrix is not accessible, then the value assigned will be 0. In this way, the matrix is composed as Figure 3 shows.



		Criteria Columns			
		$a_{i,j}$	Name	Surname	e-Key ...
Activity Rows	Ebay User	$a_{1,1}$	$a_{1,2}$	$a_{1,3}$	...
	Paypal User	$a_{2,1}$	$a_{2,2}$	$a_{2,3}$	...
	Amazon User	$a_{3,1}$	$a_{3,2}$	$a_{3,3}$	...
	...	...	...	...	...

**Fig. 3.** Trust Matrix for Criteria and Activity Allocation

The user extensibility is represented by a single row matrix. Figure 4 shows this formalism, in which this single row matrix is made up of single elements that operate as a scalar product with the user matrix. This provides a scalable representation of our approach.

	User 1	User 2	User 3	...
User Row	$u_1$	$u_2$	$u_3$	...

**Fig. 4.** User Matrix in a Single Row

The values of the criteria correspond to the number of users. Figure 5 shows the matrix representation.

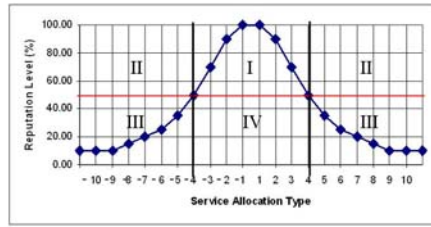
$a_{i,j}$					
$a_{1,1}$	$a_{1,2}$	$a_{1,3}$	...	,	$T$
$a_{2,1}$	$a_{2,2}$	$a_{2,3}$	...		
$a_{3,1}$	$a_{3,2}$	$a_{3,3}$	...		
...	...	...	...		

**Fig. 5.** User Matrix for Multiple Users

We assume that multiplication of two matrices is well-defined only if the number of columns of the left matrix is the same as the number of rows of the right matrix. The number of criteria must correspond to the number of users; this restriction is strictly followed to get the identity matrix. Thus, when the diagonal contains the value “1” in all of its elements, a user is evaluated as a trusted user to operate services and applications.

Figure 6 shows the trust function as a matrix representation for multiple user criteria. In other words, the matrix representation helps to identify possible cases when it is necessary to evaluate multiple-criteria cases. Furthermore, this evaluation is made in combination with the user’s activity represented as statistical values.





**Fig. 8.** Regions Related to Policy Service Allocation and User's Reputation

Region I is the most secure region, where the users are trusted all the time. Region IV has users that lack high reputation values, but these users can be considered reliable as they do not attempt to create services and their reputation value is medium. Region II is where the users have created many services, but do not themselves use those services; consequently, these users are trusted but do not have high reputation levels. Region III is where hackers and other malicious users are located; these users have many services they created and therefore, the reputation of each user is low.

The methodology used to create the trust and reputation policy-based model, which can be applied to trust the management of service applications, is to define the trust model representation and the concepts involved according to standard sociological models. Sociological criteria are used to define relationships between the information in the service model described as policies, and the information contained in service management policies.

The formal representation of social models are expressed in policy-based form and integrated as classes into the object-oriented policy-based management system. The policy model function defining the number of pervasive management operations in reference with the number of policies is shown in (4), details are described in [5], we associate the management operations with the activity of the user, thus in this way are generated the values used for statistic operations.

$$\sum_{X_s=1}^{ps+pn} F[\{(Ct_n)_m\}\{(Xs_n)_m\}]I^{(ps+pn)} \rightarrow \text{Service Operations} \quad \dots(4)$$

where  $ps$  = number for initial service policies,

$pn$  = number of total service policies, and

$X_s$  = service function;  $Ct$  = content function for values of  $n \geq 1$  and  $m \geq 1$

The advantage of combining policy-based management and reputation approaches is the possibility to define values based on the statistical variations in the reputation level and use those values in policies that define the trust level of the user. Figure 9 depicts an example where the number of sub-regions is more specific as a consequence of many users generating services. The policies here assist in evaluating regions and sub-regions according to the same criteria already defined for each of the four main regions above. The number of policies follows the function used in (4) to calculate service operations and define the user activity according to the set of services that the user is using.

The probability is calculated based on specific sub-regions of the four main regions already defined, according to the policy model function in [5], and includes the policy model to construct the trust model for services support.

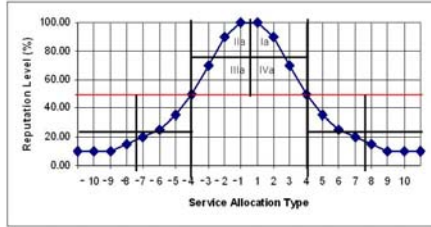


Fig. 9. Sub-regions Related to Policy Service Allocation Number

The policies also assist in supporting decisions based on multiple criteria. Put another way, policies are used to evaluate a diversity of opinions from different users about the same service. The multiple criteria evaluation is done according to personal experiences when using the service provides a level of reputation.

### 4 Policy-Based Trust Management Model

We support the idea that using social trust models for systems in which users communicate and make decisions having an inherent risk will enhance the security of such systems. A social model of trust needs to be used in order to provide a robust information model that is able to represent user information in a formal manner. Once this is done, this model can then be used in autonomic systems.

Figure 10 illustrates the challenge and the scope of applying trust management multi criteria results. From sociology, we have a social model of trust based on reputation, whereas from technology, we have the ability to create new services. When these two studies are combined a more powerful services composition to trustworthy users can be assigned. The definition of the trust model based on reputation to be used by policy-based systems can make use of the information to generate new services. Therefore, applying such a trust model requires different assignments of metadata to describe trusted users as well as malicious users.

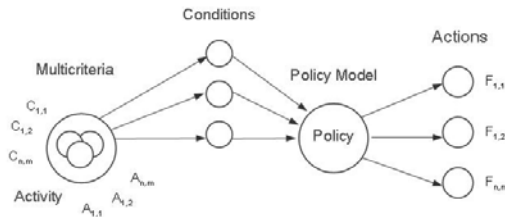


Fig. 10. Policy-Based Model and Trust Concepts Relationships

## 4.1 Policy-Based Descriptions

The user's information should be used to generate personalized services that reside in the service management system but are enabled by the service operators. Both service creators as well as operators should be able to determine if the services are performing properly according to both the service model definition as well as with the proposed trust model. In this study the use of the DEN-ng policy information model [29] facilitates the inclusion of business goal policies, and utility functions. The DEN-ng *PolicyRule* class represents an intelligent container that gathers metadata and at least one (or more) *PolicyEvent*, *PolicyCondition*, and *PolicyAction*.

Figure 11 shows the definition of Management Policies used in this approach; note that (1) Management Policies may use *any* type of structural representation of a Policy Rule, since the *ManagementPolicy* is an intelligent container that aggregates *PolicyRuleStructures*; (2) the concepts of *PolicySubject* (a set of *ManagedEntities* that requests and/or invokes policies in a holistic manner from and/or on a *PolicyTarget*) and *PolicyTarget* (a set of *ManagedEntities* that a set of policies will be applied to).

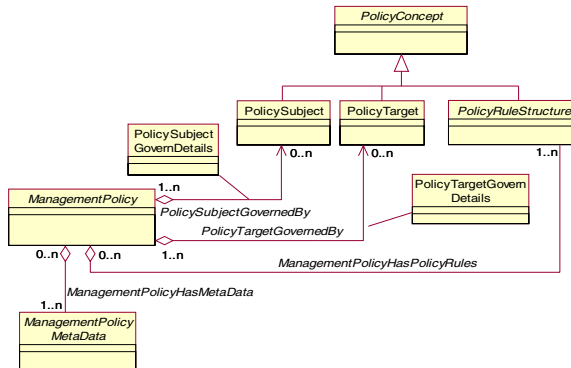


Fig. 11. Simplified DEN-ng ManagementPolicy Model

In DEN-ng, a *ManagedEntity* is something of interest that can be managed. Any *ManagedEntity* can have contextual information associated with it. We model context [40] as an overall concept (the Context class) that may be made up of a set of independently manageable aspects. The associations relating *ManagedEntity* to Context and *ContextData* are both optional. *ContextDataFacts* and *ContextDataInferences* are intelligent containers that house facts and inferences, respectively, that are computed by external applications. Both the Context and *ContextData* classes use the composite pattern for flexibility and extensibility. The *ContextAtomic* and *ContextDataAtomic* classes represent context that can be modeled as a single, stand-alone object. In contrast, the *ContextComposite* and *ContextDataComposite* classes represent context objects that are made up of multiple distinct Context or *ContextData* objects that can each be separately managed. Hierarchies of context information can be defined and related to other context information through the *HasContextData* aggregation. For example, the Context object “Communication” could have the following *ContextData* objects associated with it: PSTN, *CellularDevice*, PDA, and *ComputerDevice*, to model the characteristics of

fixed telephone lines, mobile phones, PDAs, and computers, respectively. Each of these four classes of device uses different types of media and provides different types of communication experiences, and hence different contexts.

The purpose of the *ContextDataDetails* association class is to define the particular semantics of how *ContextData* relates to Context. This enables different types of *ContextData*, each modeling a specific aspect of an overall Context, to be aggregated together with their own semantics. The *ContextSemantics* class represents data and/or knowledge that describes the behavioral aspects of the Context that this *ManagedEntity* is associated with. A similar class (*ContextDataSemantics*) is constructed for the *ContextData* hierarchy. These two classes represent a convenient point for fusing information from ontologies with data from information and data models. For example, machine-based reasoning can now be used with both of these data. They also present convenient points for either augmenting context information (e.g., tagging it with metadata to enhance information retrieval) and/or using context data to perform (for example) a set of services. Finally, these two semantics classes enable the application to declare what it needs to complete its view of context, as opposed to merely obtaining context information.

Figure 12 shows a simplified view of the DEN-ng context-aware policy model. Its purpose is to relate context changes to policy changes by selecting the set of Policy Rules that are appropriate for this particular context. Those Policy Rules are then applied by the autonomic manager to govern system behaviour. The *SelectsPolicies* aggregation defines a given set of Policies that should be loaded based on the current context. Hence, as context changes, policy can change accordingly, enabling our system to adapt to changing demands. The *PolicyResultAffectsContext* association enables policy results to influence Context.

The selected working set of Policies uses the *GovernsManagedEntityRoles* aggregation to define the appropriate roles of the *ManagedEntities* that are influenced by this Context; each *ManagedEntityRole* defines functionality of the *ManagedEntity* that can take on that role.

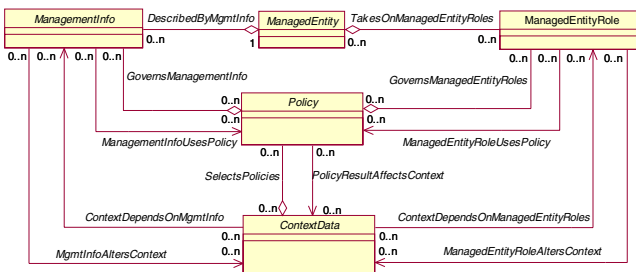


Fig. 12. Simplified DEN-ng Context-Aware Policy Model

Specifically, *Policy* is used to define which management information will be collected and examined; this management information affects policy decisions, as well as selecting which policies should be used at any given time. Once the management information is defined, then the two associations *MgmtInfoAltersContext* and

*ContextDependsOnMgmtInfo* codify these dependencies (e.g., context defines the management information to monitor, and the values of these management data affect context, respectively).

## 4.2 Service Logic Descriptions

The logic defining logic interactions and technological implications between organizations in a pervasive service is a result of using service management operations in form of policies (i.e. Service Code and Policies Distribution, Code Maintenance, Service Invocation, Code Execution and Service Assurance). The service management policies are referenced in this paper and detailed examples can be found in [5]. The service logic description corresponds to the most common management operations; however, a more extended set of functions that better reflect application requirements can be used instead.

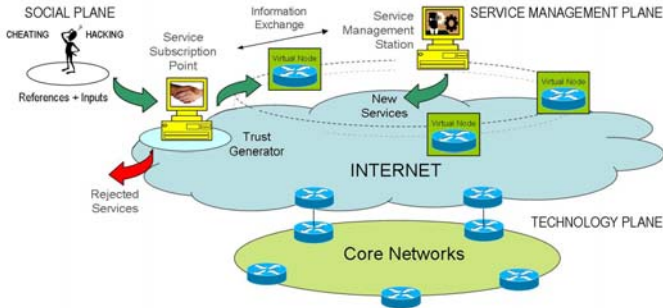
A more detailed description of these functions can be found in [5]. Implementing decisions using these logic descriptions constitute a first task of this approach towards creating a tool to support trust management. The use of service management logic when DEN-ng is being used as the policy information model includes descriptions which do not assume 'static' information for expressing user requirements. In contrast, the service management systems can process logic descriptions that can be defined dynamically as a result of user interaction.

Additionally the formal language used for expressing web-services is OWL [41]. One of the advantages of using OWL to describe logic concepts is the availability of formal tools that use OWL for parsing, reasoning and editing. We use OWL to describe information that cannot be expressed with graphic notation. We use OWL as the formal language that describes the graphic representations and logic sentences as part in the policy model supporting trust management.

## 5 Trust Model Scenario

Trust management is able to provide recommendations to governance systems when choices have an inherent security risk. Our implementation uses OWL as a formal language to represent reputation and trust using first order logic; this enables assertions about reputation and trust to be *proven*. This approach also uses policy-based management as the mechanism to execute and enforce decisions in pervasive service operations. Figure 13 shows the scope of test scenarios. These scenarios are focusing on controlling pervasive services and Internet services. System management is supported by a trust generator subsystem with the objective to decentralize access decisions, in this way the management systems delegates security decisions with the objective for improving the management operations.

The most important objectives of our trust- policy-based management system are: (1) to decentralize decisions necessary to ensure proper operation, and (2) to support the deployment of new services in policy-based management systems that can use user information from users who are trustworthy, according to the trust model, to generate new services following the preferences and requirements of those users.



**Fig. 13.** Mapping of Trust Model into Web-Services and Internet

An important type of service that we aim to support with multicriteria support is what sociologists have called *swift trust* [38]. This is a type of trust deployed by a social actor in temporary systems such as online communities or temporary working teams. In our scenarios, we use this model to support temporary services that, for example, only last for a particular session. This reflects the idea that people depend on this trust model as the alternative to avoid an indeterminate amount of time while measurements are collected that enable the system to compute who can be trusted.

In addition, communications systems must offer trust management and security in the operations performed and governed by management systems. However, this is a challenging trade-off between security and performance, due mainly to the use of proprietary information and data models to design and guide the implementation of securing the information and its associated management processes. Since there is no one common information or data modeling standard that can represent vendor-specific management and security data, securing systems remains a stovepipe design process. This is exacerbated by the increasing number of diverse technologies, each with their own associated operational and management data. Just as there will never be one programming language that all applications will use, there will never be a single information model that all vendors and application developers will use.

This scenario is based on trust management concepts, and uses policy-based principles to provide trust management mechanisms based on user reputation. The scope of this research is founded in real world social scenarios, where the application of trust and reputation models acts as inputs in communication and service management systems to detect potential users attempting to create cheating services.

## 6 Conclusions

This paper describes the research challenges for trust management in an environment that uses policy-based management to build reputation and trust using social models. The main research contribution is our approach to support multi-criteria reputations in trust management with policies-based mechanisms that offer a formal alternative for representing and implementing guide for trust management.

This research work proposes to decentralize management decisions by using a user-based trust and reputation mechanism. The main application of this model is in



the framework of autonomic solutions for future Internet services. However, the scope of the model is not limited to this scenario; it can be extended for any system requiring decentralized decisions with multi-criteria processes as generator of trust.

Social trust and reputation models are applied in communication systems in order to provide guidance for trusted users to make decisions having a security risk, as well as to help the system detect malicious users and hackers attempting to create cheating services or other disruptive services. Future research is being conducted for developing and comparing different implementations of the model and the statistical results are applied on simulations for decentralized management tasks.

## Acknowledgements

This research activity is being funded by High Education Authority (HEA) into the PRTL Cycle 4 research program in the framework of the project *Serving Society: Management of Future Communications Networks and Services*.

## References

- [1] Ruohomaa, S., Kutvonen, L.: Trust Management Survey. In: Herrmann, P., Issarny, V., Shiu, S.C.K. (eds.) *iTrust 2005*. LNCS, vol. 3477, pp. 77–92. Springer, Heidelberg (2005)
- [2] Khare, R., Rifkin, A.: Trust management on the World Wide Web. *Computer Networks and ISDN Systems Archive* 30, 651–653 (1998)
- [3] The NGOSS Technology Neutral Architecture, TMF 053, Version 5.7 (November 2006)
- [4] Strassner, J.: *Policy Based Network Management*. Morgan Kaufmann, San Francisco (2004)
- [5] Serrano, J.M., Serrat, J., Strassner, J., Foghlú, M.Ó.: Facilitating Autonomic Management for Service Provisioning using Ontology-Based Functions & Semantic Control. In: 3rd IEEE International Workshop on Broadband Convergence Networks (BCN) 2008 in IEEE/IFIP NOMS 2008, Salvador de Bahia, Brazil, April 07-11 (2008)
- [6] Strassner, J.: Introduction to DEN-ng., Tutorial for FP7 PanLab II Project (January 21, 2009)
- [7] Dressler, F., Carreras, I.: *Advances in Biologically Inspired Information Systems: Models, Methods, and Tools*. Springer, Heidelberg (2007)
- [8] Blaze, M., Feigenbaum, J., Lacy, J.: Decentralized Trust Management. In: *Proceedings of the 1996 IEEE Symposium on Security and Privacy*, Los Alamitos, California, USA, pp. 164–173. IEEE Computer Society Press, Los Alamitos (1996)
- [9] Blaze, M., Feigenbaum, J., Resnick, P., Strauss, M.: *Managing Trust in an Information-Labeling System*. European Transactions on Telecommunications (1997)
- [10] Camp, J., Genkina, A., Friedman, A.: *Social and Network Trust*, DIMACS, April 14-15, 2005. DIMACS Center, CoRE Building, Rutgers University, Piscataway, NJ (2005)
- [11] Camp, J.: *Trust and Risk in Internet Commerce*, p. 293. MIT Press, Cambridge (2000)
- [12] Mumford, E.: Participative Systems Design: Practice and Theory. *Journal of Occupational Behaviour* 4(1), 47–57 (1983)
- [13] Abowd, G.D., Dey, A.K., Orr, R., Brotherton, J.: Context-awareness in wearable and ubiquitous computing. In: *Intl. Symposium on Wearable Computers*, pp. 179–180 (1997)
- [14] Brown, P.J., Bovey, J.D., Chen, X.: Context-Aware Applications: From the laboratory to the Marketplace. *IEEE Personal Communications*, 58–64 (1997)

- [15] Chen, G., Kotz, D.: A survey of context-aware mobile computing research, Technical Report, TR2000-381, Department of Computer Science, Dartmouth College (November 2000)
- [16] Brabham, D.C.: Crowdsourcing as a Model for Problem Solving. An Introduction and Cases. *Intl. Journal of Research into New Media Technologies* 14(1) (2008)
- [17] MacLean, et al.: User-Tailorable Systems: Pressing the Issues with Buttons. In: *Proceedings of CHI, Conference on Human Factors in Computer Systems* (1990)
- [18] Li, H., Zhang, X., Wu, H., Qu, Y.: Design and Application of Rule Based Access Control Policies. In: *Proceedings of the 10th International Conference on Information and Knowledge Management, Atlanta, GA, USA, November 5-10* (2001)
- [19] Grandison, T., Sloman, M.: Specifying and Analysing Trust for internet Applications. In: *Towards the knowledge Society: eCommerce, eBusiness and eGovernment. The Second IFIP International Conference on E-Commerce, E-Business, E-Government, Lisbon, Portugal* (October 2002)
- [20] NetTrust Project, <http://www.ljean.com/NetTrust/>
- [21] Lamparter, S., Agarwal, S.: Specification of Policies for Automatic Negotiations of Web Services. In: *Proceedings of the 4th International Semantic Web Policy Workshop, Galway, Ireland, November 7* (2005)
- [22] Aberer, K., Despotovic, Z.: Managing Trust in a Peer-2-peer Information System. In: *Proceedings of the 10th International Conference on Information and Knowledge Management, Atlanta, GA, USA, November 5-10* (2001)
- [23] Bonatti, P., Duma, C., Olmedilla, D., Shahmehri, N.: An integration of Reputation-based and policy Trust Management. In: *Proceedings of the 4th International Semantic Web Policy Workshop, Galway, Ireland, November 7* (2005)
- [24] Damianou, N., Bandara, A., Sloman, M., Lupu, E.: A Survey of Policy Specification Approaches, Dept. of Computing, Imperial College of Science Technology and Medicine, London, UK (2002)
- [25] Moore, E., Elleson, J., Strassner, J.: Policy Core Information Model-Version 1 Specification. IETF Request for comments (RFC 3060) (February 2001), <http://www.ietf.org/rfc/rfc3060.txt>
- [26] Moore, E.: Policy Core Information Model-Extensions. IETF Request for comments (RFC 3460) (January 2003), <http://www.ietf.org/rfc/rfc3460.txt>
- [27] DMTF, CIM schema, can be downloaded from [http://www.dmtf.org/standards/cim/cim\\_schema\\_v220/](http://www.dmtf.org/standards/cim/cim_schema_v220/)
- [28] TMF SID schema, members only, can be downloaded from <http://www.tmforum.org/page35501.aspx>
- [29] Strassner, J., Neuman de Souza, J., Raymer, D., Samudrala, S., Davy, S., Barrett, K.: The Design of a New Policy Model to Support Ontology-Driven Reasoning for Autonomic Networking. In: *5th Latino-America Network and Operations Management Symposium (LANOMS), Salvador Bahia, Brazil* (2007)
- [30] Strassner, J., Fu, Z.: Policy Based Enforcement of Ubiquitous Role Based Access Control. In: *4th International IEEE Workshop on Managing Ubiquitous Communications and Services (MUCS), Munich, Germany, May 25* (2007)
- [31] Davy, S., Jennings, B., Strassner, J.: The Policy Continuum – A Formal Model. In: Jennings, B., Serrat, J., Strassner, J. (eds.) *Proc. of the 2nd IEEE International Workshop MACE, Multicon, Berlin. Multicon Lecture Notes, No. 6, pp. 65–78* (2007)
- [32] Dasgupta, P.: Trust as a Commodity. In: *Trust: Making and Breaking Cooperative Relations. Blackwell, Oxford* (1988)

- [33] Resnick, P., Zeckhauser, R., Friedman, E., Kuwabara, K.: Reputation Systems. *Communications of the ACM* 43(12), 45–48 (2000)
- [34] Sztompka, P.: *Trust: A sociological Theory*. Cambridge University Press, Cambridge (1999)
- [35] Schneier, B.: *Applied Cryptography: Protocols, Algorithms, and Source Code in C*, 2nd edn. John Wiley and Sons, New York (1996)
- [36] Lampson, B., Rivest, R.: SDSI - A Simple Distributed Security Infrastructure. In: *DI-MACS Workshop on Trust Management in Networks*, South Plainfield, NJ (1996)
- [37] De Paoli, S., Kerr, A.: *Conceptualizing Trust*. NIRSA Working Paper N. 40, National University of Ireland Maynooth (2008)
- [38] Meyerson, D., Weick, K.E., Kramer, R.M.: *Swift Trust and Temporary Group. Trust in Organisations*. Sage, Thousand Oaks (1996)
- [39] Sloman, M.: Policy Driven Management for Distributed Systems. *Journal of Network and Systems Management*, 215–333 (1994)
- [40] Strassner, J., Samudrala, S., Cox, G., Liu, Y., Jiang, M., Zhang, J., van der Meer, S., Foghlú, M.Ó., Donnelly, W.: The Design of a New Context-Aware Policy Model for Autonomic Networking. In: *5th IEEE ICAC*, Chicago, Illinois, June 2-6 (2008)
- [41] De Bruijn, J., Fensel, D., Lara, R., Polleres, A.: *OWL DL vs. OWL Flight: Conceptual Modelling and Reasoning for the Semantic Web* (November 2004)