F. Moyano, C. Fernandez-Gago, and J. Lopez, "Implementing Trust and Reputation Systems: A Framework for Developers Usage", *International Workshop on Quantitative Aspects in Security Assurance*, 2012.
NICS Lab. Publications: https://www.nics.uma.es/publications

Implementing Trust and Reputation Systems: A Framework for Developers' Usage

Francisco Moyano, Carmen Fernandez-Gago, Javier Lopez
NICS
www.nics.uma.es
University of Malaga,
Spain
{moyano,mcgago,jlm}@lcc.uma.es

Abstract

During the last decades, a huge amount of trust and reputation models have been proposed, each of them with their own particularities and targeting different domains. While much effort has been made in defining ever-increasing complex models, little attention has been paid to abstract away the particularities of these models into a common set of easily understandable concepts. We propose a conceptual framework for computational trust models that is used for developing a component-oriented development framework that aims to assist developers during the implementation phase.

1 Introduction

The concept of trust in computer science derives from the concept in sociological, psychological and economical environments. The definition of trust is not unique. It may vary depending on the context and the purpose where it is going to be used. Despite it is admitted of paramount importance when considering systems security, a standard definition of trust has not been provided yet. However, it is wide accepted that trust might assist decision-making processes such as those involved in access control schemes.

The concept and implications of trust are embodied in the so-called trust models, which define the rules to process trust in an automatic or semi-automatic way in a computational setting. There are different types of trust models, each one considering trust in different ways and for different purposes. The origins of trust management date back to the nineties, when Marsh [10] proposed the first comprehensive computational model of trust based on social and psychological factors. Two years later, Blaze [2] identified trust management as a way to enhance the problem of authorization, which up to that date was separated into authentication and access control. These two early contributions show that

trust can be conceived in different ways and for different purposes. From these seminal works onwards, different types of trust models have been proposed, with different purposes and targeting different settings.

This heterogeneity often leads to confusion as one might easily lose the most relevant concepts that underlie these trust models. This is precisely the motivation for this work. We aim to shed light on computational trust concepts and how they relate to each other. By trust concept or trust-related concept, we refer to any notion that has a high relevance, according to how frequently the notion arises in existing trust models. Our intention is to build the foundations towards the design of a development framework that supports the acommodation of heterogeneous trust and reputation models. We advocate that the identification of the main trust-related concepts can help in the design of such a framework. We intend to identify the main concepts that are common in most trust management models.

One issue with trust models is that they are very context-dependent, and are often designed as ad-hoc mechanisms to work in a limited range of applications. Actually, the standard is to plug a trust model into an existing, already-built application after-the-fact. This might lead to architectural mismatches between the application and the model, and the reusability of the model could also be damaged. Moreover, it is not possible for the model to exploit all the information available to the application, since there is not any systematic procedure to include the model as a holistic part of the application. As a consequence, there are no mechanisms to consider trust requirements from the very beginning of the software development lifecycle or to align the design of the model with the design of the application.

To overcome these shortcomings, we propose a component-oriented development framework that allows implementing trust models as a core part of the applications themselves. Our aim is to assist developers during the development of applications.

The rest of the paper is organized as follows. Section 2 surveys the main concepts related to trust. In order to do we provide some related work. This section also introduces our own definition. We will leverage on these concepts to design the conceptual model we propose in Section 3, which is used for eliciting the requirements and components of the framework that will assist developers to implement trust models in Section 4. Section 5 concludes the paper and outlines the future work.

2 Background on the Concept of Trust

The aim of this section is to provide the background on trust related concepts needed for providing the conceptual framework in Section 3.

2.1 Definitions of Trust

Many definitions of trust have been provided along the years. The concept is complex and it spans across several areas such as psychology, sociology, economics, law, and more recently, computer science. The vagueness of this term is well represented by the statement "trust is less confident than know, but also more confident than hope" [12].

Gambetta [4] defines trust as "a particular level of the subjective probability with which an agent will perform a particular action [...] in a context in which it affects our own action". McKnight and Chervany [11] explain that trust is "the extent to which one party is willing to depend on the other party in a given situation with a feeling of relative security, even though negative consequences are possible". For Olmedilla et al. [13], "trust of a party A to a party B for a service X is the measurable belief of A in that B behaves dependably for a specified period within a specified context (in relation to service X)". Ruohomaa and Kutvonen [15] state that trust is "the extent to which one party is willing to participate in a given action with a given partner, considering the risks and incentives involved". Finally, Har Yew [5] defines trust as "a particular level of subjective assessment of whether a trustee will exhibit characteristics consistent with the role of the trustee, both before the trustor can monitor such characteristics (or independently of the trustor's capacity ever to be able to monitor it) and in a context in which it affects the trustor's own behavior".

We propose the following definition : *trust is a subjective, context-dependent property that is required when (i) two entities need to collaborate (i.e. there is a dependence relationship between them and there exists the willingness to collaborate), but they do not know each other beforehand, (ii) and when the outcome of this collaboration is uncertain (i.e. entities do not know if they will perform as expected) and risky (i.e. negative outcomes are possible).* In this situation, trust acts as a mechanism to reduce the uncertainty in the collaboration and to mitigate the risk. As risk increases trust becomes more crucial.

2.2 Trust Model Concepts

Trust models are very hetereogeneous. This heterogeneity depends on many factors such as the trust definition they use or their application domain. Some approaches also identify the concepts that influence trust models before defining them [17]. In order to provide a conceptual framework for trust models we first establish a classification of them. However, this task is not straightforward and there are many ways to tackle it. We propose the following classification:

- *Decision Models.* Trust management has its origins in these models [2]. They aim to make more flexible access control decisions, simplifying the two-step authentication and authorization process into a one-step trust decision. *Policy models* and *negotiation models* fall into this category. They build on the notions of policies and credentials, restricting the access to resources by means of policies that specify which credentials are required to access them.

- *Evaluation Models*. These models are often referred to as *computational trust*, which has its origin in the work of Marsh [10]. Their intent is to evaluate the reliability (or other similar attribute) of an entity by measuring certain factors that have an influence on trust in the case of *behaviour models*, or by disseminating trust information along trust chains, as it is the case in *propagation models*. An important sub-type of the former are *reputation models*, in which entities use other entities' opinions about a given entity to evaluate their trust on the latter.

Making a classification is important as it eases the extraction of common features between different classes of models. It is not possible (or better said, it is not useful) to compare policy models such as PolicyMaker [2] with a reputation model such as eBay's [14], because their nature and workings are very different. However, it makes sense to extract some common features for all types of models. Each type of model exhibits its own features which allow us to identify the most meaningful ones.

3 A Conceptual Framework for Trust Models

3.1 A Conceptual Model

A trust model aims to compute trust in a given setting. This setting should have, at least, two entities that need to interact. An entity might play a role or even several ones. The basic roles are trustor (the entity that places trust) and trustee (the entity on which trust is placed). Once there is a trustor and a trustee, we claim that a trust relationship has been established. A trust relationship has a purpose, which can be for example controlling the access to a resource, the provision of a resource or the identity of an entity. It might also serve to set trust in the infrastructure (devices, hardware, etc). In the very end, the purpose of a trust model is to aid making a decision. At the higher level, it is a trust decision in the sense of answering the question: would this entity behave as expected under this context? At a lower level, an entity trusts a property of another entity. For instance its capability to provide a good quality of service. A trust model also makes some assumptions, such as "entities will provide only fair ratings" or "initial trust values are assumed to exist", and might follow different modeling methods. All of the above gives us the idea of which are the common concepts present in any type of trust model as it can be seen in Figure 1.

For the sake of simplicity, we divide our conceptual framework into the same three types of models described in Section 2.2.

3.1.1 Concepts for Trust Decision Models

As their name suggests, policy models (e.g. PolicyMaker [2]) use policies, which specify the conditions under which access to a resource is granted. These conditions are usually expressed in terms of credentials, signed logical statements that

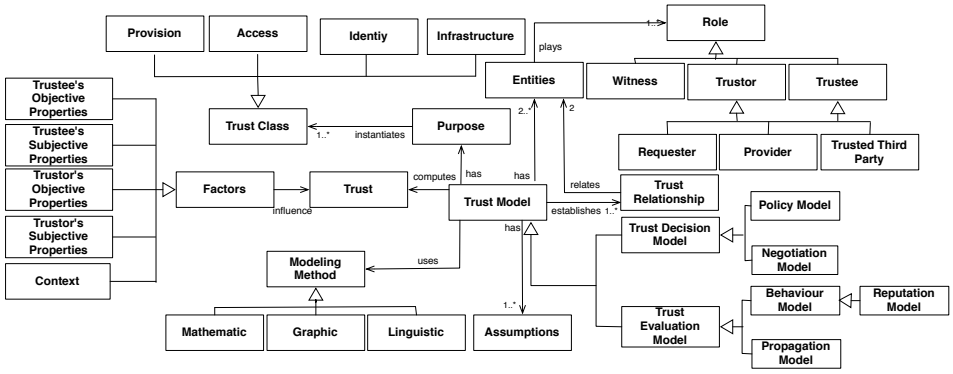

Figure 1: Common Concepts for Trust Models

assert that an entity is which it claims to be, or that it is member of a group. Credentials might have different formats, including X.509 certificates and XML. Another concept of policy models is the compliance checker, in charge of checking whether the credentials satisfy the policies. Policies are written in a policy language. Policy languages used by these models might consider policy conflicts resolution. Likewise, the model might also support the search for a credential through credential chains. Some models also include the required components to verify that a credential is valid.

The other type of trust decision models are negotiation models, being TrustBuilder [18] the first representative implementation of them. Trust negotiation models add a protocol, called *negotiation strategy*, during which two entities perform a step-by-step, negotiation-driven exchange of credentials and policies until they decide whether to trust each other or not. This strategy allows protecting the privacy of the entities as policies and credentials are only revealed when required. A later work [8] supports the implementation of different trust negotiation models. Here the authors state that trust negotiation can use evidence types, which represent information about the negotiation process (e.g. certain steps of the negotiation were already accomplished) and have a purpose (e.g. optimization of the negotiation).

The conceptual model for decision models is depicted in Figure 2.

3.1.2 Concepts for Trust Evaluation Models

As we mentioned in Section 2.2 Trust Evaluation Models can be classified into Behaviour models and Propagation Models.

Concepts for Behaviour Models Behaviour models often follow a trust lifecycle with three phases. First, a bootstrapping phase might be required to assign initial trust values to the entities of the system; some other times initial values are assigned. Trust propensity is a concept related to the bootstrapping

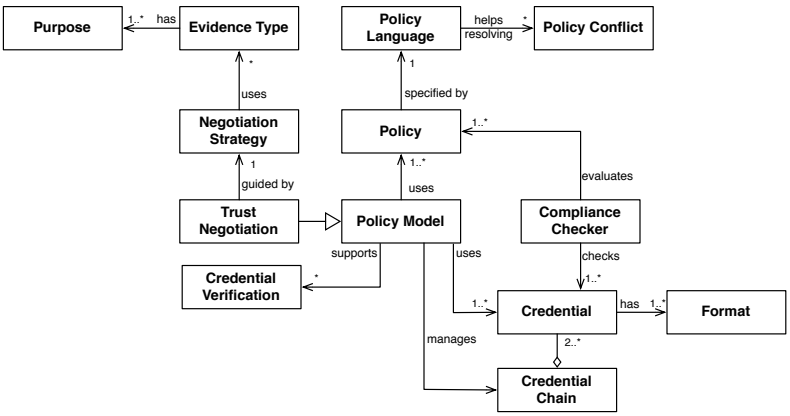

Figure 2: Concepts for Decision Models

phase and it refers to the propensity of the model towards high or low trust values in the beginning. Second, monitoring is performed to observe a variable or set of variables. Finally, an assessment process is done in order to assign values to the monitored variables and to aggregate them into a final trust or reputation score.

In behaviour trust, each trust relationship is tagged with a trust value that indicates to what extent the trustor trusts the trustee. This value can be unidimensional or multi-dimensional, and according to Jøsang [7], might have different degrees of objectivity and scope. The former refers to whether the measure comes from an entity's subjective judgement or from assessing the trusted party against some formal criteria. The latter specifies whether the measure is done against one factor or against an average of factors.

Trust values are assigned to relations using a trust assessment process, where trust metrics are used to compute them. Trust metrics use variables, such as risk or utility, and combine them in order to yield a final score for the measured attribute(s). Basic examples of attributes are trust and reputation. Attributes can be more specific, such as "quality of service provider" or "reliability of a seller". Trust metrics use computation engines, which may include simple summation or average engines, continous engines, discrete engines, belief engines, bayesian engines, fuzzy engines or flow engines. Jøsang et. al. [7] provide a summary of their features.

The source of information that feeds the metric might come from direct experience (either direct interaction or direct observation), sociological and psychological factors. Reputation models use public trust information from other entities to compose a trust evaluation. Reputation models can be centralized, where there is an entity in charge of collecting and distributing reputation information; or distributed, when there is no such an entity and each one has to maintain a record of trust values for other entities, and send this information to

the rest of entities. Regardless of which information source is used to compute the trust value, the model might consider how certain or reliable this information is (e.g. credibility of witnesses), and might also consider the concept of time (e.g. how fresh the trust information is).

Finally, a behaviour model might use a game-theoretic approach (as most existing trust models do), where relationships between entities is emphasized in terms of direct experience, feedbacks, utility, risk, and so forth; or it might be socio-cognitive, where mental models of entities are built to consider beliefs in properties. All the concepts discussed in this section are depicted in Figure 3, together with propagation models concepts, which are described next.

3.1.3 Concepts for Propagation Models

Propagation models often assume that several trust relationships have been established and quantified, although this is not always the case. They aim to create new trust relationships by disseminating the trust values information to other entities. Some models assume that trust is transitive and exploit this property, although transitivity is not, in general, considered as a property that holds [3].

Some behaviour models implement propagation mechanisms. For example, Advocato [9] is a reputation model that allows users to provide a ranking for other users. However, it is also a propagation model, since it allows computing a reputation flow through a network where members are nodes and edges are referrals between nodes.

New trust values are often computed by means of operators, and in several models, we find two of them: a concatenator and an aggregator. The former is used to compute trust along a trust path or chain, whereas the latter aggregates the trust values computed for each path into a final trust value. For example, in [1] the authors use a sequential and a parallel operator in order to compute trust along a path. Subjective logic [6] uses a discounting operator to compute opinions along different trust paths, and a consensus operator to combine them into a final opinion. All the concepts discussed are shown in Figure 3.

3.2 Models Comparison

The concepts identified in the previous section constitute a conceptual framework for the comparison of trust models. As a way to validate our framework, we have chosen a set of relevant trust models that represent the types discussed earlier, namely PolicyMaker [2], TrustBuilder [18], Marsh's model [10], Jøsang's belief model [6], Agudo et al. [1], eBay reputation model [14] and REGRET [16]. Table 1 shows the comparison among these models under the lens of their common features. In Table 2 we compare the trust decision models, whereas trust evaluation models are compared in Table 3. Note that the classification has been made according to the features explicitly presented by the corresponding authors, and that due to the diversity of the models, in some circumstances the classification for some concepts is subjective according to our own intepretation.

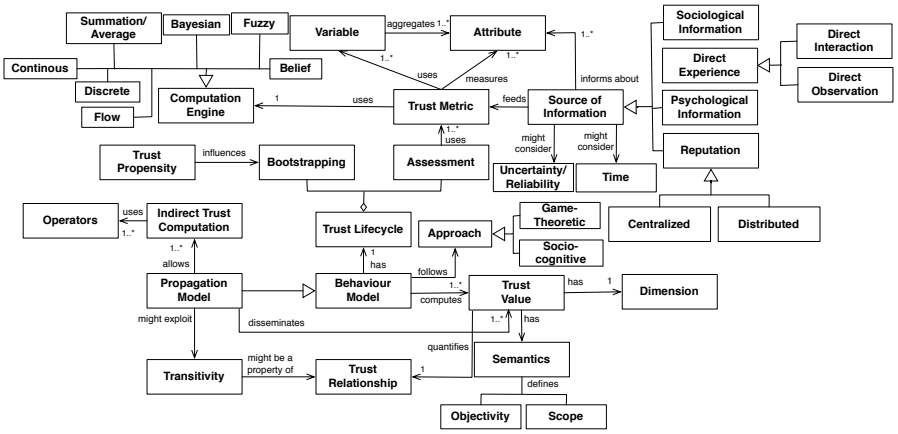

Figure 3: Concepts for Evaluation Models

Table 1: Common Features Comparison. (*T=trustor/trustee, R/P=requester/provider, W=Witness, TTP = Trusted Third Party, AT=Access Trust, IT=Identity Trust, PT=Provision Trust, PM=Policy Model, NM=Negotiation Model, BM=Behaviour Model, PrM=Propagation Model, RM=Reputation Model*)

Model	Role	Purpose	Type	Method
PolicyMaker	R, P	AT, IT	PM	Linguistic
TrustBuilder	R, P	AT, IT	NM	Linguistic
Marsh's	T, W	AT, PT	BM, PrM	Mathematic
Jøsang's	T, W	AT, PT	RM, PrM	Mathematic
Agudo et al.	T, W	AT, PT	PrM	Graphic, Mathematic
eBay	R, P, W, TTP	PT	RM	Mathematic
REGRET	R, P, W	PT	RM	Mathematic

Table 2: Decision Models Comparison. (*PC=Policy Conflict detection, CC=Credential Chaining support, CV=Credential Verification support, ET=Evidence Type, -=undefined or not explicitly mentioned*)

						Trust Negotiation	
Model	P. Language	C. Format	PC	CC	CV	Strategy	ET
PolicyMaker	PolicyMaker	PGP's sig, X.509 cert	-	-	-	-	-
TrustBuilder	XML, IBM's TPL	X.509 cert	-	✓	✓	✓	-

Table 3: Evaluation Models Comparison. (*DI=Direct Interaction, DO=Direct Observation, SI=Sociological Information, PI=Psychological Information, R=Reputation, C=Centralized, D=Distributed, GT=Game-Theoretic, I.Trust=Indirect Trust, -=undefined or not explicitly mentioned*)

				Source of Information							
Model	Approach	Dimension	C.Engine	DI	DO	SI	PI	R	I. Trust	Uncertainty	Time
Marsh's	GT	1	Continuous	✓	-	-	-	-	✓	-	✓
Jøsang's	GT	3	Belief	✓	-	-	-	D	✓	✓	-
Agudo et al.	-	1	Flow	-	-	-	-	D	✓	-	-
eBay	GT	1	Summation	-	-	-	-	C	-	-	-
REGRET	GT	1	Fuzzy	✓	-	✓	✓	D	-	✓	✓

4 Framework Requirements and Components

In this section, the component-oriented development framework for trust is presented. We concentrate on the requirements and components for it to be implemented. For the sake of simplicity, we focus on the requirements and components of evaluation models.

4.1 Framework Requirements

This section summarizes the requirements that the framework must meet in order to facilitate the implementation of evaluation models. At a high-level, the framework has to support the implementation of three types of evaluation models, namely reputation models, behaviour models and propagation models. Althought these models have commonalities, they also pose subtle differences that the framework must support.

The primary goal of reputation models is to compute reputation scores for entities. These scores must be stored (centrally or distributively) and entities should be able to access this information before interacting with other entities. On the other hand, behaviour models establish relationships between entities, and their main goal is to compute trust values for these relationships. Finally, propagation models also build on trust relationships, and their primary goal is to disseminate trust information to establish new trust relationships.

The following list of requirements describe the coarse-grained functionality that the framework should provide to developers:

- Entities management: entities hold trust values in other entities. The framework must allow the creation, binding and naming of entities.
- Trust relationships management: trust relationships might change along time. New trust relationships might be created (e.g. by propagation models), other relationships might be deleted, and it is likely that trust values change as well.

- Trust metrics definition: although the framework can provide some default built-in metrics implementations, it is important to let developers to define their own trust metrics, as they are the core concept in evaluation models.

- Variables management: a trust metric is composed of variables. It is important to let developers to create new variables, which can be used by user-defined metrics.

- Computation engines management: an engine implements a trust metric. This engine uses variables according to certain rules. Engines range from simple summation or average functions to complex fuzzy and probability distributions.

- Indirect trust computation: the framework should provide ways to determine the value of an undefined trust relationship based on defined ones by propagating trust information.

- Operators definition: indirect trust computation relies on operators that take trust paths as input and return trust values as output (and thus, a new trust relationship). Although several operators should be provided by default, the framework should allow developers to define new operators.

4.2 Framework Components

Figure 4 shows a components diagram of the framework architecture.

We can divide the framework components into *application-oriented components* and *user-oriented components*. The former are those components used by the applications in a black-box fashion, whereas the latter represent components that the user (i.e. developer) might access directly in order to change their state or behaviour. This responds to the necessity of implementing a grey-box framework where some basic functionality is built-in whereas extended functionality can be implemented by the developer to fulfil his or her needs.

Application-oriented components are detailed next:

- Reputation Model: this component encapsulates the functionality of reputation models, where the main interface provided to applications is *Manage Reputation*. This interface allows managing entities through the *Relational Manager* component, update their reputation by means of the *Metric Manager* component and store this reputation to an external database through the *External RDBMS* component.

- Behaviour Model: instead of individual entities, behaviour models manage trust relationships by means of the *Relational Manager* component. The assignment of trust values to these relations is done through the *Metric Manager* component. Applications are offered an interface to manage these relationships, which includes to compute their trust values.

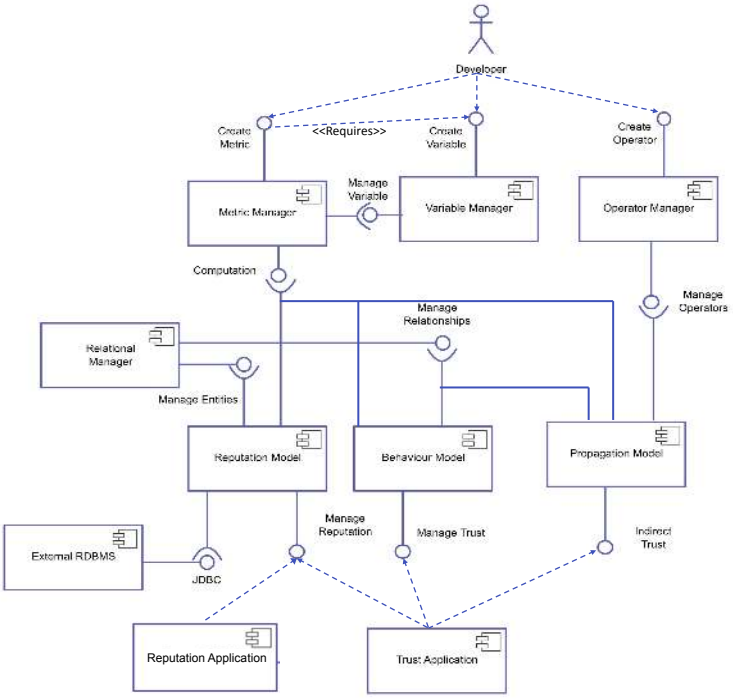

Figure 4: Evaluation Models Components

- Propagation Model: also built upon trust relations, this component provides an interface to applications that allows calculating a trust value between two entities that are not directly related. For this purpose, this component uses one interface provided by the *Operator Manager* component.

- Relational Manager: an inner component that manages entities and trust relationships, making easier and transparent to applications how this management is done.

- External RDBMS (Relational Database Management System): this component offers persistence through its JDBC (Java DataBase Connectivity) interface to the *Reputation Model*. Depending on whether the reputation model is centralized or distributed, the implementation of the interface will store a unique instance of a JDBC connector for all entities or one different connector for each entity.

User-oriented components are described next:

- Metric Manager: this component is in charge of managing trust metrics, providing an interface that every type of model can use to compute reputation or trust. Since metrics use variables, this component requires

accessing the variables managed by the *Variable Manager*. Also, as a user-oriented component, it offers an interface to create new metrics. Creating a metric might require creating new variables through the *Variable Manager*.

- Variable Manager: this component manages variables that are used by metrics. The user might define new types of variables. The number and complexity of the variables depend on the complexity of the metric to define.

- Operator Manager: as explained earlier, this component offers an interface to a*Propagation Model* in order to compute indirect trust relationships. However, it also offers an interface to users that allow them to create new operators, that is, new ways to propagate trust information and create new indirect trust relationships.

5 Conclusion and Future Work

In this paper, we propose a conceptual framework for trust. Given the high heterogeneity of trust models, it is challenging to provide a general framework for it. We first identify and relate concepts that are general enough to be common to every trust model. After classifying trust models into different types, we then identify and relate a set of concepts that are more closely related to each type of model.

Based on this conceptual model we presented a component-oriented development framework to assist developers during the implementation of applications that might require support from trust or reputation models. For the sake of simplicity and space constraints we concentrated on evaluation models. As the application is developed using the framework, trust models are aligned with the design of the application and they can exploit all the data available to the application. We elicit the requirements that the framework should meet, and also identify several components of the architecture.

As future work, we intend to validate the framework. We are interested in supporting the implementation of models where the trust values are represented by a tuple of values (multiple dimensions) rather than by a single value. We intend to allow defining a different metric for each dimension in order to provide greater flexibility. Some trust models yield an uncertainty value together with the trust value, in order to inform other entities about how certain the trust value should be considered. We plan to add support for this feature as well. Roles played by entities or the membership of entities to a given group are factors taken into account in other models to determine trust. This is a feature that we intend to also include in the near future too. Finally, we aim to add more complex built-in computation engines, including beta-probability distributions and fuzzy engines.

Acknowledgements

This work has been partially funded by the European Commission through the FP7 project NESSoS under grant agreement number 256980, by the Spanish Ministry of Science and Innovation through the research project ARES (CSD2007-00004) and by the Andalusian government through the research project PISCIS (TIC-6334). The first author is funded by the Spanish Ministry of Education through the National F.P.U. Program.

References

[1] Isaac Agudo, Carmen Fernandez-Gago, and Javier Lopez. A model for trust metrics analysis. In *5th International Conference on Trust, Privacy and Security in Digital Business (TrustBus'08)*, volume 5185 of *LNCS*, pages 28–37. Springer, 2008.

[2] Matt Blaze, Joan Feigenbaum, and Jack Lacy. Decentralized trust management. In *IEEE Symposium on Security and Privacy*, pages 164–173, 1996.

[3] Bruce Christianson and William S. Harbison. Why isn't trust transitive? In *Proceedings of the International Workshop on Security Protocols*, pages 171–176, London, UK, UK, 1997. Springer-Verlag.

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

[5] Chern Har Yew. *Architecture Supporting Computational Trust Formation*. PhD thesis, University of Western Ontario, London, Ontario, 2011.

[6] Audun Jøsang. A logic for uncertain probabilities. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 9(3):279–311, June 2001.

[7] Audun Jøsang, Roslan Ismail, and Colin Boyd. A survey of trust and reputation systems for online service provision. *Decision Support Systems*, 43(2):618–644, March 2007.

[8] Adam J. Lee, Marianne Winslett, and Kenneth J. Perano. Trustbuilder2: A reconfigurable framework for trust negotiation. In Elena Ferrari, Ninghui Li, Elisa Bertino, and YÃ$\frac{1}{4}$cel Karabulut, editors, *IFIPTM*, volume 300 of *IFIP Conference Proceedings*, pages 176–195. Springer, 2009.

[9] Raph Levien. *Attack Resistant Trust Metrics*. PhD thesis, University of California at Berkeley, 2004.

[10] Stephen Marsh. *Formalising Trust as a Computational Concept*. PhD thesis, University of Stirling, April 1994.

[11] D. Harrison McKnight and Norman L. Chervany. The meanings of trust. Technical report, University of Minnesota, Management Information Systems Research Center, 1996.

[12] Keith W Miller, Jeffrey Voas, and Phil Laplante. In Trust We Trust. *Computer*, 43:85–87, 2010.

[13] D. Olmedilla, O.F. Rana, B. Matthews, and W. Nejdl. Security and trust issues in semantic grids. In *Proceedings of the Dagsthul Seminar, Semantic Grid: The Convergence of Technologies*, volume 5271, 2005.

[14] Paul Resnick and Richard Zeckhauser. Trust among strangers in Internet transactions: Empirical analysis of eBay's reputation system. In Michael R. Baye, editor, *The Economics of the Internet and E-Commerce*, volume 11 of *Advances in Applied Microeconomics*, pages 127–157. Elsevier Science, 2002.

[15] Sini Ruohomaa and Lea Kutvonen. Trust management survey. In *Proceedings of the Third international conference on Trust Management*, iTrust'05, pages 77–92, Berlin, Heidelberg, 2005. Springer-Verlag.

[16] Jordi Sabater and Carles Sierra. Regret: reputation in gregarious societies. In *Proceedings of the fifth international conference on Autonomous agents*, AGENTS '01, pages 194–195, New York, NY, USA, 2001. ACM.

[17] Sabrina De Capitani Di Vimercati, Sara Foresti, Sushil Jajodia, Stefano Paraboschi, Giuseppe Psaila, and Pierangela Samarati. Integrating trust management and access control in data-intensive web applications. *ACM Trans. Web*, 6(2):6:1–6:43, June 2012.

[18] M Winslett, T Yu, K.E Seamons, A Hess, J Jacobson, R Jarvis, B Smith, and L Yu. Negotiating trust in the Web. *Internet Computing, IEEE*, 6(6):30–37, 2002.