Adaptiveness and Social-Compliance in Trust Management. 
A Multi-Agent Based Approach. 

Reda Yaich 

October 21, 2013 

1 Introduction 

Virtual communities (VCs) are socio-technical systems wherein distributed individuals (human and/or artificial) are grouped together around common objectives and goals [10, 6]. In such systems, participants are massively collaborating with each others by sharing their private resources and knowledge. A collaboration always bears the risk that one partner exhibits uncooperative or malicious behaviour. Thus trust is a critical issue for the success of such systems. 

In human-based virtual communities, participants rely on a various set of information while assessing their degree of trust in each others. That information represent, in most of the situations, descriptions of partners properties on which constraints are specified to distinguish trustworthy partners from untrustworthy ones. These constraints are expressed by means of trust policies and constitute the individual’s trust requirements. Such policies are often made implicit as there is no need for humans to make them explicit. But in our investigations, we are more interested in agent-based virtual communities where participants are assisted by software agents that behave on their behalf [11]. 

In such settings, participants must be able to explicitly specify the trust policies their agents should rely on to derive trust. For that purpose, trust policy languages represent a formal and flexible interface between the human and its agent in the virtual community [13]. 

The work presented in this dissertation addresses the problem of trust management in open and decentralised virtual communities (VCs). We address this problem by proposing an Adaptive and Socially-Compliant Trust Management System (ASC-TMS) adopting a multi-agent based approach. Our proposal relies on the joint use of adaptive trust policies and adaptation meta-policies, which allow VCs members to make adaptive and socially-compliant trust decisions.
2 Foundations and Problem statement

In order to further motivate our problematic, we introduce a running example that we will rely on to highlight the important of the contributions we make in this thesis.

Open Innovation is currently recognized as a new approach that companies and organizations are adopting to enhance innovation in their R&D departments by harnessing external ideas and stimulating creativity. Based on this concept, a number of commercial (e.g. Hypios.com) and free (e.g. W3C Community Group) open innovation platforms (OIP) have been proposed to facilitate the innovation process through the creation of Open Innovation Communities (OIC). These platforms constitute a showcase for R&D problems to which the community members try to find a solution and earn the reward\(^1\). The success of such ephemeral communities relies on full collaboration and intensive resources sharing, making trust a fundamental and critical issue.

2.1 Foundations

Now let us consider a concrete instantiation of the above use case. Assume that Eureka is the OIP wherein Alice, Bob and Charlie joined together and created an OIC called ABC. The aim of the ABC members is to achieve a goal \(g\) and gain the reward \(m\).

In the following, we assume that each member possesses a set of individual resources that support him in his activities. We refer to such resources as \(X.r\) where \(X\) is the member identifier and \(r\) is the resource identifier. Analogously, we suppose that the collaboration activity among the community is achieved through the creation and update of some common resources that we represent by \(C.r\) where \(C\) is the community identifier.

Starting from this basic system description, we introduce the concepts related to trust management. In our scenarios, we suppose that agents are acting on the behalf of humans, and thus that they are taking automatic decisions based on the user’s policies. Here, we are primarily concerned by decisions that relate to trust, and hence, we only consider the policies that agents use to make such decisions. Therefore, we require that each user endows its agent with a set of policies (one for each decision) that this agent will rely on for its trust decisions. Here again, we use the dot "." notation to represent the policies used by an agent (e.g. \(X.\pi\)).

Policies are statements that specify under which conditions an agent is allowed to trust other agents for some specific issue. Without loss of generality, access control requests represent the main type of issues addressed in this thesis abstract. Nevertheless, our trust model can manage any other form of trust decisions. Access control requests are messages in which a requester (an agent) asks a controller (another agent) a permission to perform and operation on a particular resource (that he owns or controls), or to simply join the community.

\(^1\)Rewards represent any incentive that drives the community members to participate.
(he belongs to). If the controller trusts the requester, this latter’s request will be accepted and the permission he asked will be delivered.

Delivering such permissions, implies that the requester succeeds in satisfying the policy used by the controller. These policies often states desirables properties that the controller requires in its partners in order to consider them trustworthy.

Considering the fact that VCs are decentralized systems wherein no one owns all the resources, and hence, every body can make trust decisions about common resources. The first question, that one is legitimate to ask is “How can Alice, Bob and Charlie regulate the access to their community (ABC), and consequently, to their collective resources?” To this purpose, we assume that similarly to individual resources, VCs members use collective policies to regulate the decisions they make within the community. Consequently, when an agent belongs to a community, he has to enforce and comply with the collective policies of its community.

2.2 Problem Statement

The system configuration described above raises several challenges and motivates most of the contributions carried out in our thesis. These challenges can be summarized in the following research Issues:

2.2.1 Policies Combination

Let us assume that the members of the same community agreed on the fact that they need a common policy (eventually many, if there are several decisions to make). The research issue here is to find a mechanism that these agents can use to build a collective policy based on their respective individual policies. If we represent such mechanism by the “?” symbol, the collective policy can be formally defined as follows:

\[ C.\pi_i = X.\pi_i ? Y.\pi_i ? ... ? Z.\pi_i \]  

Where \( i \) refers to the decision for which the collective policy has to be built.

2.2.2 Policies Combination

Starting from the supposition that the members of the community ABC are able to specify collective policies, the next question that we were led to ask relates to collective policies enforcement. Indeed, the ability of a community member to use trust conditions (in its policy) that are not too far from those specified in the collective policy of its community is very important. From the perspective of behaviour science, psychology and sociology, this kind of attitude improves the community cohesion and avoids agents exclusion. With respect to that, the requirement here would be to find a mechanism that emulates the compliance effect described in the Social Impact Theory, and particularly, the
majority influence presented in the experiments conducted by Asch and Milgram [2]. This issue can also be formally specified as follows:

\[ C_\pi i = X_\pi i \ast C_\pi i \] (2)

2.2.3 Policy Instantiation

Let us consider a resource \( r_1 \) that represents a text file in which \( alice \) takes her notes about ideas she wants to develop. Initially, each idea would be just a title or a sentence. Then progressively, the ideas will evolve to become increasingly elaborated. So intuitively, one can say that the policy governing \( r_1 \) should be different for each version of \( r_1 \). However, and unfortunately, in most general cases policies does not evolve with the resources they protect.

Given the intense and sustained paces of interactions and activities that characterizes virtual communities, policies changes such the ones illustrated above would be so frequent that the human intervention is neither possible nor desirable. Impossible because including a human in the agent decision loop represents a bottleneck strategy as the human can not be present each time an agent has to make a decision. It is moreover not desirable as the human intervention, besides being time-consuming and error-prone, means a loss of any automation advantage made possible by the use of agents technology. Thus, agents need to know whether the policies specified by their users best handle the context in which their interactions are undertaken, and if not how to change them in that perspective.

2.2.4 Policy Negotiation

In some other situations, it is not the resource change that motivates the policy adaptation but the partner (i.e. controller) with whom the agent is interacting. For instance, let us consider the request of a newcomer called \( Alex \). \( Alex \) wants to access a resource shared by \( Bob \). The problem is that the policy specified by \( Bob \) requires that \( Alex \) discloses his Identity credential to prove that he is an adult in order to gain access to \( Bob \)'s resource. The problem that arises here is that \( Alex \)'s Identity credential is a sensitive resource as it contains his personal and private information such as his age, his address and his photo. The disclosure of such information is a violation to his privacy, and thus, can be harmful for him. Therefore \( Alex \) prefers to disclose his social security card that maximizes his privacy instead of his identity.

Here again, the amount of offers and counter-offers that an agent has to make, and receives during interactions is so high that it makes unrealistic the intervention of the human user each time the agent has to make a decision. Consequently, agents must be able to manage trust negotiations, and adapt in consequence the policies they are using.
3 Contributions

The Adaptive and Socially-Compliant Trust Management Model (ASC-TMS) represent our main contribution to address the challenges highlighted in Section 2.2. This contributions made by the ASC-TMS to the discipline of trust management for multi-agent systems are summarized in this section.

3.1 Flexible Policy Language

In policy instantiation, the agent must be able to understand each of the conditions stated in its policy. Then he needs to reason on these constraint and change them if necessary. This requirements is not specific to instantiation, each issue highlighted in Section 2.2 requires that the agent understands, reason on and operates on trust policies. To that aim, our very first object was to identify from the existing literature which policy language best fit such requirement. Unfortunately, most of the policy language were designed to be changed when if necessary, but none allowed automatic changes. Consequently, we framed our own policy language wherein policies are defined using a weighted-logics like style. Our Flexible Policy Language (FPL) has four interesting properties that we summarize as follows:

- **Expressiveness**: in FPL, trust condition conveyed by a policy are not flat. We associated to each condition two argument, the first states how important is the condition with respect the trust decision, while the second states whether the condition is mandatory or not. With FPL, we tried to capture the real semantic of trust as humans never accord the same important to the requirements they require in trust.

- **Flexibility**: Unlike traditional policy languages, thanks to the weighted logics inspiration, the evaluation of an FPL policy is not binary. This feature is interesting for three reasons; at the first hand, it allows agent to be able to make decision even if the partner does not fully satisfy their policies, it enables them to rank partners based on their policy satisfaction level, and finally, it allows them to handle uncertainty as they admit that some information are not available or not relevant.

- **Semantic**: FPL was designed on the basis of weighted logics. However, the variables used in FPL propositions, their corresponding values and the operators used overs them are defined in a shared ontology. Thanks to this ontology, agent are able to understand, reason on and affect each elements contained in the policy.

- **Adaptiveness**: finally, the last feature concerns the ability of FPL policies to be modified by agents. Again, this features was made possible thought the use of weighted logics. Weighted logics is recognized to be a well-suited formal framework for specifying dynamic programmes. In our thesis, we took benefit from this property to allow agent automatically, and dynamically change the policies use.
3.2 Adaptation meta-policies

In the previous step, we specified a formal policy language framework based on which agents are able to understand, reason on and change the policies they use. Thus, VCs human user a able to specify to their respective agent how they want them to make trust decision. Although the decisions they will make are neither binary nor flat, these agents still do not know how to tackle the issues highlighted in Section 2.2.

Thus a logical continuation of our work was to find a mechanism that human users can use to specify how their agents are allowed to change their policies. Without such mechanism, these users can rapidly be overburdened by the tasks of changing trust policies, and thus, exposed to errors (e.g. too restrictive policy, rigid negotiation, not complying with the collective policy, etc.).

So the second main contribution we make in this thesis was to define, a meta-policy language over FPL. Using these meta-policies, VCs humans users can specify when, why and how a policy can be adapted. In the light of that, the intuitive solution we went for to specify ARL was to use event-condition-action (ECA) rules where the event part of the rules states when the meta-policy is triggered, the condition reflects the conditions for which the meta-policy was made active (why) and the action defines how and agent can change the policy.

Using ECA rules to adapt policies has nothing nothing innovative, as our contribution is not at this level. The contribution we make here by defining an algebra for policies adaptation. This algebra includes nines operators; seven are simple (namely, add, del, restrict, relax, lower, higher) are used the during instantiation and the negotiation, while the two remaining are complex (namely, combine, integrate) are used in combination and integration.

4 Adaptive and Socially-Compliant Trust Management System

The ultimate step was to bring together the flexible policy language and the adaptation meta-policies language to build a coherent framework. In this framework (ASC-TMS), VCs users are able to use the meta-policies to specify strategies. For instance, with respect to the negotiation issue, we designed the negotiation as an extensive game wherein the controller tries to maximizes the number of credential disclose he obtains from the requester, while this latter strategy would try to minimize it. Using meta-policies, both the controller and the requester are able to specify the ordering of offer and counter-offer they want to make, and if necessary how they are willing to change their policies (using restrict, relax, lower, higher operators). Analogously, with respect to instantiation, meta-policies are used to define how a policy can changed if some event happens (resource value or sensitivity increases).

We also used meta-polices to define strategies about policies combination (merging several policies to build a collective policy). Here, we used a voting system to build (collaboratively) the collective policy. Finally, meta-policies was
used to specify when and how collective policy is integrated with the individual one.

5 Positioning and Contributions

One of the first steps in this thesis was to survey the current state of the art in the discipline of trust management in distributed environments. To that aim, we have reviewed and described some of the most representative trust management models.

Our review of the literature evidenced how much trust is a well established mechanisms that support cooperation and collaboration in open and decentralized systems, such as virtual communities [7, 6, 15]. We also showed how the trust issue received, in the last decade, an increasing interest that gave rise to numerous trust models. Each model tried to represent the information a participant should collect about the others and proposes an evaluation scheme to derive trust from that information [15]. We explained how each of these models works; how they gather information about others and how trust is computed based on these informations.

In our review of the literature, we have also discussed the main common features of the trust model we described, as well as the most important features that characterises each of them. We also analysed their their pros and cons with respect to virtual communities scenarios.

In sum, and as a result, we built a comprehensive classification of trust models based on the informations (credentials vs declarations) the yr e on and the nature of the scheme they use on that information (policies vs functions) [1, 3, 4, 8, 9, 5, 14, 12]. We also showed how much these two dimensions were correlated as almost all policy-based approaches rely on credentials, while none of the approaches using functions relies on credentials.

Ultimately, both approaches possess considerable limitations with respect to the requirements for trust management in virtual communities. Therefore, we opted in our thesis for an hybrid approach in which we tried to get benefit from the best part of each.

6 Conclusion

The doctoral research presented in this thesis focuses on the problem of designing policies for virtual communities. To address this issue, we framed a MAS architecture to support trust management in virtual communities [15]. We used MAS concepts to formalize a theoretical framework for the specification of flexible trust policies where policies are automatically adapted using meta-policies. In future research, we will try to experiment our Adaptive and Socially-Compliant Trust Management System in a real life virtual community scenario to evaluate how benefit to collaborating communities the system could be.
References


