Local vs Global Policies and Centralized vs Decentralized Control in Virtual Communities of Agents

Guido Boella
Dipartimento di Informatica
Università di Torino- Italy
E-mail: guido@di.unito.it

Leendert van der Torre
SEN-3 - CWI Amsterdam
The Netherlands
E-mail: torre@cwi.nl

Abstract

We are interested in the design of policies for virtual communities of agents based on the grid infrastructure. In a virtual community agents can play both the role of resource consumers and the role of resource providers, and they remain in control of their resources. We argue that this requirement creates a distinction between two dimensions: global vs local and centralized and decentralized control by means of policies. The providers should be enabled to specify their local policies on their own resources, but their policies should be consistent with the global policies. At the same time, some aspects of the decentralized control should be delegated to specialized providers; this delegation requires a distinction between the authorization to access a resource and a permission to do so.

1 Introduction

Pearlman *et al.* [11] define a virtual community on the grid infrastructure as a large, multi-institutional group of individuals who use a set of rules, a policy, to specify how to share their resources, such as disk space, bandwidth, data, online services, *etc.*.

In this paper, we distinguish two dimensions in the policies for virtual communities: the local vs global distinction and the centralized vs decentralized dimension.We motivate these distinctions and argue that they are both necessary in the definition of policies to control the system and that they provide answers to complementary requirements. Finally we propose informal definitions of the concepts composing the policies for virtual communities.

This paper builds on our previous work on local policies ([4], extended by [5]) and on the notions of authorization and permission ([8]). Here, we relate the two approaches showing their dialectic interplay in the definition of policies.

2 Policies for virtual communities

Policies in virtual communities become more complex than in distributed systems due to, e.g.:

- Every agent can play both the role of resource consumer and of resource provider. Providers retain the control of their resources and specify in local policies the conditions of their use.
- A central manager permits access according to its policies if it owns all the resources. In contrast, in virtual communities, no one owns all resources. Moreover, a centralized administration could be too heavy and affect the core business activities of the system.
- Resource providers implement local policies according to the community's security policies. However, they should not be overburdened by the task of updating the policies as they change and new members join.
- Agents who participate to the community are heterogeneous and change frequently, so they cannot be assumed to be always cooperative and to stick to the system policies, concerning both requesting access to resources and providing access to their resources.
- Decentralized authorities can cope in a better way with local idiosyncratic situations: "each party of the network can decide in each circumstance whether to accept credentials presented by a second party" [1].
- It is possible that global policies become more easily obsolete [10]: "in real life, many policies are routinely ignored because of the perception that changing circumstances have made them redundant."
- The agent prefer not to give up their own power to enforce local policies for the access to the resources they control. E.g., as [13] argue, there are "cases where security administrators are not fully trustworthy".

The problem of designing policies for virtual communities has been recently raised, e.g., by Pearlman *et al.* [11] and Sadighi Firozabadi and Sergot [12]. Pearlman *et al.* [11] argue that the solution is "to allow resource owners to grant access to blocks of resources to a community as a whole, and let the community itself manage fine-grained access control within that framework". The centralized management of resources owned by resource providers is performed by a *Community Authorization Service* (CAS) : "A community runs a CAS server to keep track of its membership and fine-grained access control policies. A user wishing to access community resources contacts the CAS server, which delegates rights to the user based on the request and the user's role within the community. These rights are in the form of capabilities which users can present at a resource to gain access on behalf of the community", [11]. As, [11] argue "the exercise of rights is effective only if the resource provider has granted those rights to the community".

However, the proposed granting access is not sufficient for maintaining the control of the system. As discussed above, the CAS should not be overburdened in its task, and part of the work should remain at the level of the local policies of providers. For a set of agents to be a *virtual community*, local access policies should be organized according to some global policies which define how the resources should be shared among the participants. This requirement must be traded off with the need of leaving autonomy to the participant resource providers. At the same time, providers should not be entirely trusted in their implementation of global policies by means of local ones. Since there is no plausible way to enforce the respect of global policies by constraining the architecture, it is necessary to have a *normative control mechanism* [9] able to specify global policies about local policies. In fact, local resource providers (such as a web server) cannot be coerced to provide their services or to deny them to users: rather they can be only motivated by rewards and sanctions (e.g., the sanction can be the exclusion from the community). So it is necessary that the local authorities are provided with incentives to implement the global policies by means of local ones.

Given the requirement that local providers retain the ability to issue local policies, the idea that the CAS delegates rights becomes troublesome. In a virtual community, since an agent maintains the control of its resource, a request is granted only if access is permitted by the local policy of the agent. Hence, the authorization by the CAS contained in the capability is not enough for assuring that a request of a user will be granted. This shows that the authorization issued by the CAS is conceptually different from the local permission granted by the resource provider and that the resource provider delegates to the CAS the power to issue authorizations rather than to issue permissions, since the latter power requires being in control of a resource.

3 Global vs local policies

Consider the following example. An agent a_2 joins some virtual community; it will both use the resources provided by the community, say downloading shared files, and provide its resource to the other members of the community, say some of its disk space to store files: agent a_2 plays both the role of a resource consumer, and that of a resource provider. Since agent a_2 controls its disk space (it is the only one who can decide that storing or retrieving files take place), it regulated the access to the disk by means of some local policy: prohibitions and permissions. E.g., it prohibited to read files during the day and it permitted to store files not exceeding 2.5Mb.

When agent a_2 joins the community, its contract for the participation by the CAS a_1 prescribes that it should provide access to its resources to all the members of the community. But consider the following case. Another participant agent, say a_3, tries to access the system. However, previous experiences before joining the community advice agent a_2 that agent a_3 could damage its resource: should agent a_2 grant agent a_3 access to its resources?

In this scenario the management of the community is organized in (at least) two levels: the global level (agent a_1) and the local one (agent a_2). Agent a_1 is a distinguished agent playing the role of a global authority (CAS) which issues global policies and negotiates the conditions for the participation of agents to the virtual community. Agent a_2 is a provider of some resource it is in control of. Moreover all the agents (a_1, a_2 and a_3) can also play the role of users of the resources of the community. What distinguishes agents a_2 and a_1 is the fact that they are providers: they are in control of some resources. The control of resources consists in not providing the service if the provider does not want to (e.g., a files cannot be accessed if the server does not provide an answer to a request). But also that an agent may influence negatively the behavior of other agents. E.g., at the local level agents depend on the provider for the current and future access to the local resource. Moreover, all agents depend on the global level for their membership to the system. If they do not stick to its global policies they are denied citizenship. In [14]'s terminology, other agents depend on it. In our model this is the essential precondition for the ability to issue policies. We formalize the notion of dependence in [2].

Global policies concern the behavior of participants: for example, participants should not communicate their passwords, or distribute copyrighted files by means of the system. Or else they are banned from the community.

At the local level policies forbid, e.g., agents to store files exceeding 1Gb on a file sharing service. Or they permit participants of the community to download copyrighted files from the web server.

But as [15] argue, and as it is shown by our scenario, there are also other kinds of global policies besides these examples. There are policies that apply to other policies: global policies that constrain or permit local policies. In the scenario above agent a_1 obliges agent a_2 to permit members of the community to access its resources. Analogously, the global authority could oblige local providers to forbid access, permit to permit access, or permit to forbid access.

4 Centralized vs decentralized control

When the local provider has to implement a local policy like that members of the community must be permitted to access the resource, there is the risk that it is overburdened by the task of modifying its policy each time a new member join the system or its conditions of participation change. In fact, even if the problem of authenticating which are the current users of the community can be dealt with by some trusted third party who gives them e-certificates, it remains the problem of which members of the community are the ones which the community currently wants that they can access the resource and under which conditions they can do so. The complexity of modifications could also introduce unwanted errors in the local policy of agent a_2.

What is needed is a solution which transfers part the burden of implementing the global policies to other agents, playing the role of authorities, like the CAS, which have the knowledge and resources to perform this task. However, it is impossible to say that an authority a_1 changes the local prohibitions and permissions posed by local provider a_2. Moreover, a_1 is not in control of the local resource so it cannot impose sanctions on the users to motivate their respect of local prohibitions. Finally, agent a_2 wants to preserve its autonomy, so that it does not accept that someone else can change the norms regulating access to its resource.

The solution is that agent a_2 creates a local permission saying that authorized agents can access the resource. But the decision to authorize agents to access the resource is delegated to the authority a_1 which has up to date knowledge on the system policies and members. Delegating the decision to authorize is easier than delegating permissions: the authorization is not a norm of the agent a_2 but just a belief which can be induced by the authority by issuing e-certificates and capabilities to the agents which are authorized. Moreover, it does not require that the delegated agent is in control of the resource.

When the set of agents which can be authorized changes as a consequence of new community policies, agent a_2 does not have to change the norms regulating access: new authorizations are created when the authority a_1 issues new capabilities (or, in [8]'s terminology, a_1 declares them authorized). The capabilities are recognized by agent a_2 as the proof that the local permission to access the resource applies to a consumer a_3 requesting access.

Authorizations, thus, are the means used by authorities to regulate the access of consumers to resources which they do not control. But there is no way to make authorized users access a resource without a local permission by the resource provider which controls the resource: hence, authorizations are distinct from and presuppose local permissions. An authorization is useless unless the resource provider locally permits authorized agents to access the resource it controls: authorizations change what is prohibited to an agent and legitimate but without introducing or removing any norm.

The notion of authorization to access a resource and the notion of permission should be kept distinct to have a correct model of the situation and to prevent dangerous misunderstandings in designing access policies.

The distinction between authorization and permission allows relieving local policies from the burden of maintaining an up to date version of what is prescribed by global policies. So it plays a role which balances the one of the global vs local distinction, which aims at relieving the central authority from the (impossible) requirement of controlling each resource.

On the other hand, the authorization mechanism by itself must be warranted by global policies which oblige local providers to implement local policies which take into account authorizations carried by the capabilities assigned to the users by the CAS. For example, a global policy could oblige a provider to permit access to authorized users only and to forbid it to unauthorized ones.

In summary, there is a dialectic interplay between the two dimensions: local policies are made necessary by the architecture of a virtual community composed by different resource providers. This requirement in turn imposes the distinction between authorizations by a central authority and local permissions by providers. At the same time, global policies about local ones are necessary to enforce the correct use of authorizations by the resource providers.

5 Definitions

In this final section we present in an informal way the definition of the key notions used in [5, 8]:

Local obligation is defined as a goal of resource providers. In [3] this is paraphrased as: Your wish (goal, desire) is my command. The unfulfillment of the goal is considered as a violation and is sanctioned.

Local permission is behavior which not considered by a provider as a violation and thus it is not sanctioned. The main role of permissions is to provide exceptions to prohibitions in a given context.

Authorization is a belief of a provider which appears as a condition in some permission it issued.

Declaration of authorization is an action of an authority which states that an agent can be considered authorized according to its policy.

Delegation is the change of authority. The declaration of the authority turns into a belief of the resource provider that an agent is authorized [9]. A provider delegates the authority to decide which agents are authorized when it joins the community.

Motivational aspects of norms have been analyzed in [3] in the context of multiagent systems composed of heterogeneous agents: norms are useless unless they are supported by sanctions. And sanctions must be modelled as actions of the normative system, since it is not possible to presuppose that they are mere consequences of violations. Hence, in [3] we attribute to the normative system the status of an agent who decides whether the behavior of agents counts as a violation, and thus deserves to be sanctioned by it.

But what do global policies refer to? Which are the conditions for their satisfaction? It is not sufficient that the global obligation to permit or oblige access is satisfied by the fact that the local authority issued a permission or an obligation. In fact, norms are ineffective if they are not enforced by the authority who issued them: violations of norms should be recognized as such and sanctioned.

Hence, global policies refer not to the fact that a local norm exists but to the fact that it is enforced by the local authority by recognizing and sanctioning violations.

A global obligation by agent a_1 that agent a_2 obliges agent a_3 to do a is expressed as an obligation that agent a_2 considers $\neg a$ as a violation and sanctions it. Since the local obligation of a_2 is expressed in terms of goals that something is considered as a violation, the global obligation by agent a_1 is defined as the goal that agent a_2 considers $\neg a$ as a violation of a_3 and the goal that if a_2 does not do that, then its behavior is considered a violation by agent a_1.

A global permission by agent a_1 that a_2 obliges that agent a_3 does a is expressed as a permission by a_1 to consider $\neg a$ as a violation: agent a_1 has the goal that agent a_2 is not considered a violator by a_1 if it considers a_3 as a violator.

The local authority, however, can still violate this global policy and forbid access to users if it prefers to face the sanction with respect to permit access; in the scenario above it is possible that agent a_2 does not grant agent a_3 the resource it is entitled to by the global policy: facing a sanction by the global authority (e.g., being excluded by the community for a certain period of time) is preferred to the possibility that a_3 damages the systems (e.g., a_3 could create a denial of service).

To help the central authority in its task of enforcing the global policies, it is possible that it defines roles which have the goal of checking violations and applying sanctions. We discuss this issue in [6, 7].

References

[1] M. Blaze, J. Feigenbaum, and J. Lacy. Decentralized trust management. In *Procs. of Security and Privacy*, 1996.

[2] G. Boella, L. Sauro, and L. van der Torre. Power and dependence relations in groups of agents. In *Procs. of IAT'04*, Beijing, 2004.

[3] G. Boella and L. van der Torre. Attributing mental attitudes to normative systems. In *Procs. of AAMAS'03*, pages 942–943, Melbourne, 2003. ACM Press.

[4] G. Boella and L. van der Torre. Decentralized control: Obligations and permissions in virtual communities of agents. In *Procs. of ISMIS Conference*, pages 618–622, 2003. Springer Verlag.

[5] G. Boella and L. van der Torre. Local policies for the control of virtual communities. In *Procs. of IEEE/WIC Web Intelligence Conference*, pages 161–167. IEEE Press, 2003.

[6] G. Boella and L. van der Torre. Norm governed multiagent systems: The delegation of control to autonomous agents. In *Procs. of IEEE/WIC Intelligent Agent Technology Conference*, pages 329– 335. IEEE Press, 2003.

[7] G. Boella and L. van der Torre. Attributing mental attitudes to roles: The agent metaphor applied to organizational design. In *Procs. of ICEC'04*, 2004.

[8] G. Boella and L. van der Torre. Permission and authorization in policies for virtual communities of agents. In *Procs. of Agents and P2P Computing Workshop at AAMAS'04*, New York, 2004. Springer Verlag.

[9] G. Boella and L. van der Torre. Regulative and constitutive norms in normative multiagent systems. In *Procs. of KR'04*, Whistler (CA), 2004.

[10] J. Cole, J. Derrick, Z. Milosevic, and K. Raymond. Author obliged to submit paper before 4 july: Policies in an enterprise specification. In *Procs. of POLICY'01*. 2001.

[11] L. Pearlman, V. Welch, I. Foster, C. Kesselman, and S. Tuecke. A community authorization service for group collaboration. In *Procs. of Policies for Distributed Systems and Networks*. 2002.

[12] B. Sadighi Firozabadi and M. Sergot. Contractual access control. In *Procs. of Workshop of Security Protocols*, Cambridge (UK), 2002.

[13] B. Sadighi Firozabadi, M. Sergot, and O. Bandmann. Using authority certificates to create management structures. In *Procs. of Workshop of Security Protocols*, pages 134–145, 2001.

[14] J. S. Sichman, R. Conte, C. Castelfranchi, and Y. Demazeau. A social reasoning mechanism based on dependence networks. In *Procs. of ECAI'91*, pages 188–192, 1991.

[15] M. S. Sloman. Policy driven management of distributed systems. *Journal of Network and Systems Management*, 2(4):333–360, 1994.