Permission and Authorization in Policies for Virtual Communities of Agents

Guido Boella[1] and Leendert van der Torre[2]

[1] Dipartimento di Informatica - Università di Torino- Italy. E-mail: guido@di.unito.it
[2] CWI Amsterdam and TU-Delft - The Netherlands. E-mail: torre@cwi.nl

Abstract. We study the design of policies for virtual communities of agents based on peer-to-peer systems or the grid infrastructure. In a virtual community agents can play both the role of resource consumers and the role of resource providers. Moreover, the agents remain in control of their resources, and therefore we distinguish between the authorization to access a resource given by the virtual community and the permission to do so issued by the resource providers. We propose a logical multiagent framework for virtual communities that distinguishes *three* roles: resource consumption, provision, as well as authorization.

1 Introduction

Peer-to-peer systems and the grid infrastructure allow to create virtual communities. For example, Pearlman *et al.* [1] define a virtual community as a large, multi-institutional group of individuals using a set of rules, a policy, to specify how to share their resources, such as disk space, bandwidth, data, online services, *etc*. In order to control the distributed nature of peer-to-peer or grid systems, policies are defined using norms, i.e., deontic notions like obligations, prohibitions and permissions. Inspiration comes from, amongst others, computer security and distributed systems [2, 3]. For example, in the Kazaa file sharing system a user is obliged by the system to share files, otherwise his bandwidth for downloading files is reduced as a sanction.

However, policies in virtual communities are more complex than policies in traditional distributed systems, due to the following reasons.

- Every agent in the community can play both the role of a resource consumer as well as that of a resource provider. Resource providers retain the control of their resources and they specify in local policies the conditions of use of their resources.
- When there is a central manager, it permits agents to access the resources which it owns and controls, according to the policies defined by itself. In contrast, in virtual communities, access control cannot be directly implemented, since nobody owns all the resources.
- Resource providers implement local access control 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 the community.
- 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.

The problem of designing policies for virtual communities has been raised recently, e.g., by Pearlman *et al.* [1] and Sadighi Firozabadi and Sergot [4]. Pearlman *et al.* 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 the single resource providers is performed by a *Community Authorization Service* (or 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".

In this paper we discuss the design of virtual communities policies composed by prohibitions, permissions and authorizations. We address the following problems.

1. Is the task of *authorizing* requests performed by the CAS - henceforth called also authority - identical to the task performed by a resource provider when it *permits* access? How should permissions and authorizations be distinguished and how are they related? What is the relation between the CAS and the resource providers?
2. How can a resource provider delegate to the CAS the power of authorizing resource consumers and why can the power to issue permissions not be delegated?

We analyze these distinctions using our framework for normative multiagent systems [5]. As Jin and Liu [6] notice, multiagent systems "have been widely used in peer-to-peer computing" and "it is regarded as a perfect match to integrate peer-to-peer computing and agent-based systems, because since their inception, multiagent systems have been always thought of as network of peers". We extend the use of multiagent systems in peer-to-peer to normative multiagent systems, i.e., multiagent systems regulated by norms.

We use the following example in this paper, based on Pearlman *et al.*'s description of the process of accessing a resource in a virtual community. When a resource provider a_3 wants to join a community, it informs the CAS a_2, which replies with the requirements on how its resource must be shared with the community. When a resource consumer a_1 wants to access the resource of agent a_3, it must not only authenticate itself with agent a_2 providing its credentials, but it must also get a proof that its request conforms to the community's access policy. This proof is expressed by a *capability* (e.g., a X.509 certificate) provided by agent a_2 to a_1, which identifies the agent and states that it is *authorized* to access the resource. Now, agent a_1 can make the actual request to a_3, forwarding it the capability. After checking the truthfulness of the capability, agent a_3 replies to a_1. In a virtual community, agent a_3 maintains the control of its resource: the request is granted only if it is also permitted by the local policy of agent a_3. Hence, the authorization by agent a_2 contained in the capability is not enough for agent a_1's request being granted.

This paper is organized as follows. In Section 2 we discuss the notion of authorization. In Section 3 we introduce the formal agent model with the definition of norms (prohibitions and permissions), authorizations and delegation, which is illustrated by the above scenario. In Section 4 we discuss the theoretical foundations of the design and in Section 5 we summarize the results of the paper.

2 Authorization

A first cue that authorization and permission have different properties is found in the ordinary use of the terms authorization and permission. E.g., for the Cambridge Advanced Learner's Dictionary [7] permitting is “to allow something”, “to make it possible for someone to do something, or to not prevent something from happening”, while authorizing means “to give someone *official* permission to do something”. Moreover, dictionaries of law like [8] argue that authorizations and permissions are related but different concepts, and that authorizations do not create new permissions.

In virtual communities, the authorization issued by the CAS is conceptually different from the permission granted by the resource provider, because the power of issuing permissions requires being in control of a resource. Resource providers delegate to the CAS the power to issue authorizations, but not the power to issue permissions. The notion of authorization to access a resource and the notion of permission should be kept distinct to correctly model of the situation. Moreover, they should be kept apart to prevent dangerous misunderstandings in the design of access policies.

Consider the following example. An agent a_3 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_3 plays both the role of a resource consumer, $c(a_3)$, and that of a resource provider, $p(a_3)$. Since agent $p(a_3)$ 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 prohibited to store files exceeding 2.5Mb.

When agent a_3 joins the community, it agrees that also some other members use its disk space resource. In principle, agent $p(a_3)$ could modify the policy regulating the access to its resource: e.g., by maintaining the prohibitions to read file during the day and to store large files, and by adding the permission about which members of the community can store and retrieve files on its disk space. However, this solution imposes an heavy burden. 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, another problem remains: 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. Moreover, the community's access policies may change with time, so that agent $p(a_3)$ should be kept informed and should modify the norms (prohibitions and permissions) regulating access to the resource it owns. The complexity of modifications could also introduce unwanted errors in the access policy of agent $p(a_3)$.

What is needed is a solution which transfers the burden of managing the community policies to other agents, playing the role of authorities, which have the knowledge and resources to perform this task. However, it is impossible to say that the CAS $u(a_2)$ changes the prohibitions and permissions posed by agent $p(a_3)$: in our model [5] norms are defined in terms of the goals which resource providers want to achieve concerning the use of their resources. The difficulty is that nobody can change the goals of an autonomous agent. Moreover, $u(a_2)$ is not in control of the resource so it cannot impose sanctions to motivate the respect of prohibitions. Finally, agent $p(a_3)$ wants to preserve its autonomy, so that it does not accept that someone else can change the prohibitions and permissions regulating access to its resource.

The solution is that agent $p(a_3)$ creates a permission saying that authorized agents can access the resource. But the decision to authorize agents to access the resource is delegated to the CAS $u(a_2)$ 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 goal of the agent $p(a_3)$ but just a belief which can be induced by the CAS 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 $p(a_3)$ does not have to change the prohibitions and permissions regulating access: new authorizations are created when the CAS $u(a_2)$ issues new capabilities (or, in our abstract terminology, $u(a_2)$ declares them authorized). The capabilities are recognized by agent $p(a_3)$ as the proof that the permission to access the resource applies to a consumer $c(a_1)$ requesting access to it.

Authorizations, thus, are the means used by authorities like the CAS 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 permission by the resource provider which controls the resource: hence, authorizations are distinct from and presuppose permissions. An authorization is useless unless the resource provider permits authorized agents to access the resource it controls: authorizations change what is prohibited to an agent and legitimate an action but without introducing or removing any prohibition and permission.

Finally, nothing requires that agent $u(a_2)$, who is delegated the authority to authorize other agents, is itself permitted nor authorized nor delegated to authorize itself. The separation of institutional power from the permission to exercise it, identified by Makinson [9], is important for virtual communities. An organization could, e.g., outsource some administrative task such as assigning access rights to some agent without allowing it to have those access rights. In summary, the key notions are:

Prohibition is defined as a goal of resource providers. This is paraphrased as: Your wish (goal, desire) is my command (prohibition). The unfulfillment of the goal is considered as a violation and is sanctioned.

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 own policy. Using Searle [10]'s terminology, in [5], we say that a declaration generates an actual authorization if it "counts as" an authorization for the resource provider.

Delegation establish who is considered as an authority. The declaration of someone recognized as an authority turns into a belief of the resource provider that an agent is authorized. A provider delegates the power to authorize to the CAS when it joins the community.

3 A formal model

3.1 Individual agent design

In virtual communities there is no separation of resource providers from resource consumers, and they can play the role of authorities too. So we introduce a single set of agents, which can each play one or more roles. For the individual agent design we are inspired by the BOID architecture [11]. However, in contrast to the BOID architecture, prohibitions are not taken as primitive concept. Beliefs, desires and goals are represented by conditional rules.

Definition 1 (Agents). *Let $\mathcal{A} = \{a_1, a_2, \ldots, a_n\}$ be a set of n agents. An agent $a_i \in \mathcal{A}$ can play three roles:*

1. *Resource consumer, denoted as $c(a_i)$: it can access resources to achieve its goals, is subject to norms regulating security, prohibitions and permissions, and also endowed with authorizations to access resources.*
2. *Resource provider, denoted as $p(a_i)$: it can provide access to the resources it owns. We call this the* normative role, *since it can issue norms, i.e., prohibitions and permissions about the access of a resource, and enforce their respect by means of sanctions, and delegate the power to authorize resource consumers.*
3. *Authority, denoted as $u(a_i)$: it can declare resource consumers authorized when they are requested to do so. They know that their declarations are considered as authorizations by the resource providers since they have been delegated the power to authorize resource consumers on behalf of resource providers.*

Actions, represented by *decision variables,* can have conditional and indirect effects with a non-monotonic character. We assume that the base language contains boolean variables and logical connectives. The variables are either *decision variables* of an agent, which represent the agent's actions and whose truth value is directly determined by it, or *parameters*, which describe the state of the world and whose truth value can only be determined indirectly. Our terminology is borrowed from Lang *et al*. [12]. Institutional facts, a subset of the parameters, represent the legal classification of reality made by agents.

Definition 2 (Decisions). *Let $A_i = \{m, m', m'', \ldots\}$, the decision variables of $a_i \in \mathcal{A}$, and $P = \{p, p', p'', \ldots\}$, the parameters, be $n+1$ disjoint sets of propositional variables. Let the institutional facts I be a subset of P. A literal is a variable or its negation. $d_i \subseteq A_i$ is a decision of agent a_i.*

The consequences of decisions are defined by the agent's epistemic state, i.e. its beliefs about the world: how a new state is constructed out of previous ones given a decision is expressed by a set of *belief rules*, denoted by B_i. Belief rules can conflict and agents can deal with such conflicts in different ways. The epistemic state therefore also contains an ordering on belief rules, denoted by $\geq_i^B$, to resolve such conflicts.

Definition 3 (Epistemic states). *Let a rule built from a set of literals be an ordered sequence of literals $l_1, \ldots, l_r, l$ written as $l_1 \wedge \ldots \wedge l_r \rightarrow l$ where $r \geq 0$. If $r = 0$, then* *we also write* $\top \rightarrow l$. *The* epistemic state *of agent* a_i, $1 \leq i \leq n$, *is* $\sigma_i = \langle B_i, \geq_i^B \rangle$, *where* B_i *is a set of rules;* $\geq_i^B$ *is a transitive and reflexive relation on the powerset of* B_i *containing at least the subset relation.*

Example 1. Let $s = \{p\}$ be the current state, $d_1 = \{a\}$, $B_1 = \{a \rightarrow q, a \wedge p \rightarrow \neg q\}$ and $\geq_1^B = \{a \wedge p \rightarrow \neg q\} > \{a \rightarrow q\}$: q is a consequence of action a, unless p is true: the second rule is an exception to the first one. The new state resulting from the decision d_1 in state s given the belief rules B_1 is $\{p, \neg q\}$: the applicable rules are $\{a \rightarrow q, a \wedge p \rightarrow \neg q\}$, but since they are conflicting only the rule $a \wedge p \rightarrow \neg q$ with higher priority in the ordering $\geq_1^B$ is applied.

The agent's motivational state contains two sets of rules for each agent. *Desire* (D_i) and *goal* (G_i) *rules* express the attitudes of the agent a_i towards a given state, depending on the context. When facing a conflict between their motivations, different agents prefer to fulfill different goals and desires. We express these agent characteristics by a priority relation on the rules which encode, as detailed in Broersen *et al.* [11], how the agent resolves its conflicts.

Definition 4 (Motivational states). *The* motivational state M_i *of agent* a_i $1 \leq i \leq n$ *is a tuple* $\langle D_i, G_i, \geq_i \rangle$, *where* D_i, G_i *are sets of rules,* $\geq_i$ *is a transitive and reflexive relation on the powerset of* $D_i \cup G_i$ *containing at least the subset relation.*

The decision process of an agent a_i tries to minimize (according to the ordering $\geq_i$ on goal and desire rules) the goal and desire rules in G_i and D_i which remain unsatisfied given a certain decision d_i.

Definition 5 (Unfulfilled motivational states). *Let* $U(R, s)$ *be the unfulfilled rules of state* s $U(R, s) = \{l_1 \wedge \ldots \wedge l_n \rightarrow l \in R \mid \{l_1, \ldots, l_n\} \subseteq s$ *and* $l \notin s\}$ *The* unfulfilled mental state description *of agent* a_i *is* $U_i = \langle U_i^D = U(D_i, s), U_i^G = U(G_i, s) \rangle$.

Example 2. Given $\langle D_1 = \{\top \rightarrow z\}, G_1 = \{\top \rightarrow x, y \rightarrow w, z \rightarrow u\}, \geq_1 \rangle$ as the motivational state of agent a_1, the unfulfilled motivational state of agent a_1 in state $s = \{x, y\}$ is $U_1 = \langle U_1^D = \{\top \rightarrow z\},\ U_1^G = \{y \rightarrow w\} \rangle$

In calculating which are the effects of a decision d_i given an initial state s, the agent uses the belief rules B_i and the ordering on them $\geq_i^B$ to resolve the possible conflicts. Moreover, agent a_i must predict the decisions of the agents acting after itself by recursively modelling ([13]) them using the information on their belief, goal and desire rules captured by their motivational states. The reader can find the details of the qualitative decision model in [5].

3.2 Norms

Prohibitions and permissions are defined in terms of goals and desires of the bearer of the norm and of the normative role, together with two auxiliary concepts. The first concept is *violation*. The normative role can decide whether something is considered a violation or not.

Definition 6 (Violation variables). *The violation variables of agent $p(a_j)$ are a subset of the decision variables of $p(a_j)$ written as V_j=$\{V_j^i(x) \mid x$ a literal built out of a propositional variable in $P \cup A_i$ $\}$: x is a violation by agent $c(a_i)$.*

The second concept is *sanction*. Since it is not possible to assume that all agents are cooperative and respect the norms, sanctions provide motivations to fulfill the norms. A sanction is an action negatively affecting an agent, i.e., the agent desires the absence of the sanction.

Definition 7 (Conditional prohibition with sanction). *Agent $c(a_i)$ is prohibited by agent $p(a_j)$ to decide to do x (a literal built out of a variable in $P \cup A_i$) with sanction s (a propositional variable) under condition q (a proposition), $F_{(i,j)}(x, s \mid q)$, iff:*

1. *$q \to \neg x \in G_j$: if agent $p(a_j)$ believes that q it has as a goal that agent $c(a_i)$ adopts $\neg x$ as its decision.*
2. *$q \wedge x \to V_j^i(x) \in G_j$: if agent $p(a_j)$ believes that $q \wedge x$ then it has the goal $V_j^i(x)$: to recognize x as a violation done by agent $c(a_i)$.*
3. *$V_j^i(x) \to s \in G_j$: if agent $p(a_j)$ decides $V_j^i(x)$ then it has as a goal that it sanctions agent $c(a_i)$.*
4. *$\top \to \neg s \in D_i$: agent $c(a_i)$ has the desire not to be sanctioned.*

A permission to do x is an exception to a prohibition to do x if agent $p(a_j)$ has the goal that x does not count as a violation under some condition.

Definition 8 (Conditional permission). *Agent $c(a_i)$ is permitted by agent $p(a_j)$ to decide to do x (a literal built out of a propositional variable in $P \cup A_i$) under condition q (a proposition), $P_{(i,j)}(x \mid q)$, iff $q \wedge x \to \neg V_j^i(x) \in G_j$: if agent $p(a_j)$ believes $q \wedge x$ then it wants that x is not considered a violation done by agent $c(a_i)$.*

The permission overrides the prohibition if the goal that something does not count as a violation ($q \wedge x \to \neg V_j^i(x)$) has higher priority in the ordering on goal and desire rules $\geq_j$ with respect to the goal of a corresponding prohibition that x is considered as a violation ($q \wedge x \to V_j^i(x)$): $\geq_j \supseteq \{q \wedge x \to \neg V_j^i(x)\} > \{q \wedge x \to V_j^i(x)\}$. We do not consider here the problem of how normative role's characteristics can be generated; e.g., see [14] for a discussion of the problem of the legal sources of norms.

3.3 Resource, authorization and delegation

We introduce now the notion of resource, of control of a resource, of authorization and delegation of the institutional power to authorize access to a resource. An agent who manipulates a resource by means of some action is called a *resource consumer*:

Definition 9 (Resources). *Let RS be a set of resources. Let $RA_j = \{f_j(r) \mid r \in RS\}$ be a set of resource actions of agent $c(a_j)$ on $r \in RS$.*

The possibility to punish violations by means of some sanction s is among the preconditions for creating a prohibition; for this reason, it appears in the notion of controlling a resource, which is a precondition for issuing norms concerning access control. An agent who controls a resource is a *resource provider*.

Definition 10 (Control of resource). *Agent $p(a_j)$ controls a resource action f_i of agent $c(a_i)$ on resource $r \in RS$, $control_j(f_i(r))$, iff Agent $p(a_j)$ can negatively influence agent $c(a_i)$ when it executes $f_i(r)$ by means of some decision variable or parameter which it can control $s \in A_j \cup P$ such that $\top \rightarrow \neg s \in D_i$: agent $c(a_i)$ desires not to be sanctioned.*

As a particular case, $s = \neg p$ can be a literal built out of a parameter representing the failure of accessing a resource: e.g., reading a file has the desired effect of knowing the content of the file, and blocking the reading action results in the impossibility of knowing the information contained in the file. $c(a_i)$ believes that $p(a_j)$ with $m \in A_j$ prevents to achieve the effect p of $f_i(r)$ which $c(a_i)$ desires; $f_i(r) \rightarrow p \in B_i$, $\top \rightarrow p \in D_i$ and m has the effect $\neg p$: $m \rightarrow \neg p \in B_i$ and $\geq_i^B \supseteq \{m \rightarrow \neg p\} > \{f_i(r) \rightarrow p\}$.

Besides issuing norms, an agent which controls a resource can consider other agents authorized to access the resource it controls; authorizations, are a legal classification of reality for agents, and, thus, are represented by institutional facts:

Definition 11 (Authorizations). *Let the institutional facts I contain a set of so-called authorization variables: $H_j = \{u_j(f_i(r)) \mid a_i \in \mathcal{A}$ and $f_i(r) \in RA_i$ and $control_j(f_i(r))\}$ They are institutional facts representing that the resource provider $p(a_j)$ considers agent $c(a_i)$ authorized to access r with action f_i. An authorization has a meaning only if it appears among the conditions of a permission.*

Instead, declaring an agent authorized does not have the requirement to control a resource.

Definition 12 (Declarations). *Let the decision variables of agent $u(a_k)$ contain a set of so-called declaration variables $T_k = \{g_k(f_i(r)) \mid a_i \in \mathcal{A}$ and $f_i(r) \in RA_i\}$. Here $g_k(f_i(r))$ means that agent $u(a_k)$ declares agent $c(a_i)$ authorized to access r with action f_i.*

The point of declaring agents authorized is that a declaration generates an actual authorization if it counts as an authorization for the normative role controlling the resource. An example of this relation is the fact that a signature by the head of the department on a purchase order counts as the institutional commitment of the department to pay for that order: the head of the department has the institutional power to buy on behalf of the department.

Definition 13 (Counts as relation). *A decision variable $x \in A_k$ of agent $u(a_k)$, counts-as q, where q is a literal, for agent $p(a_j)$, $counts\text{-}as_j(x, q)$, only if $x \rightarrow q \in B_j$: agent $p(a_j)$ believes that x has q as a consequence.*

An agent who has been delegated the institutional power to authorize access is called an *authority*. It is not requested to control any resource.

Definition 14 (Delegation of authorization). *Agent $u(a_k)$ is delegated by agent $p(a_j)$ the institutional power to authorize agent $c(a_i)$ to do $f_i(r) \in RA_i$ by means of declaration $g_k(f_i(r)) \in T_k$ ($u_j(f_i(r)) \in H_j$), $Del_{(k,j)}(g_k(f_i(r)), u_j(f_i(r)))$, iff we have $counts\text{-}as_j(g_k(f_i(r)), u_j(f_i(r)))$.*

3.4 Applicative scenarios

In this section we sketch an applicative scenario in our model in the context of a virtual community. As shown in Figure 1 we have three agents: agent a_3, a provider $p(a_3)$ of resource f (a file), the resource consumer $c(a_1)$, and the CAS agent a_2: an authority $u(a_2)$. Agent $p(a_3)$ can block $c(a_1)$'s attempt of accessing with action r_1 ($r_1(f)$) the resource, since it is in control of the resource. When agent $p(a_3)$ joined the community (step 1) it maintained the prohibition to access f ($F_{(1,3)}(r_1(f), \neg info(f) \mid \top)$) but it agreed to share $r_1(f)$ the resource with the other members of the community by means of a permission to do $r_1(f)$. Unfortunately, $p(a_3)$ does not know which are the current members (in this case whether agent a_1 is a member) nor which is the access policy of the community concerning the resource f which $p(a_3)$ is sharing. Thus, agent $p(a_3)$ decides to consider what agent $u(a_2)$ says (or declares, in our terminology: $g_2(r_1(f)) \in T_2$) about $c(a_1)$'s access to f as an authorization $u_3(r_1(f))$ by itself: $u(a_2)$'s action $g_2(r_1(f))$ counts as $u_3(r_1(f)) \in H_3$ for $p(a_3)$. Then it permits agent $c(a_1)$ to access only if it is authorized: $P_{(1,3)}(r_1(f)|u_3(r_1(f)))$.

Agent $c(a_1)$ compares the different alternatives for achieving its goal of knowing the content of the file $info(f)$: requesting the resource by doing $r_1(f)$ alone (step 5) or first asking agent $u(a_2)$ for a declaration ($ask_1(g_2(r_1(f)))$)(3) and then requesting to access the resource (5). It knows that a request for access is considered as a violation $V(r_1(f), c(a_1))$ by agent $p(a_3)$ and, thus, sanctioned by negating the information contained in the file ($\neg(info(f))$). For this reason, it decides to ask agent $u(a_2)$ for an authorization to access agent $p(a_3)$'s resource. Agent $u(a_2)$ will provide $c(a_1)$ with the declaration since it is a goal of the $u(a_2)$ to cooperate with resource providers in enforcing the policy ($ask_1(g_2(r_1(f))) \rightarrow g_2(r_1(f)) \in G_2$). Finally, agent $c(a_1)$ knows that the declaration $g_2(r_1(f))$ of agent $u(a_2)$ is considered as an authorization $u_3(r_1(f)$ by agent $p(a_3)$: $g_2(r_1(f)) \rightarrow u_3(r_1(f)) \in B_3$.

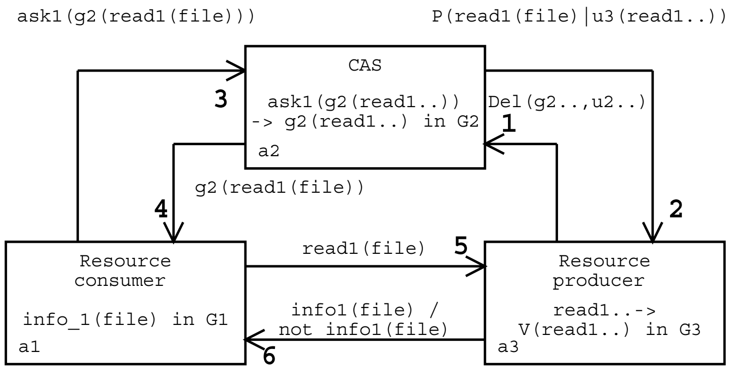

Fig. 1. Requesting access.

4 Related work

This use of the terms right, authorization and permission as synonyms is frequent in policies for managing access control in distributed systems (e.g., [15]). In this paper we show why and how these concepts should be kept distinct in the context of virtual communities.

The necessity of a fine grained analysis of the concepts of permission and authorization in the security policy field is witnessed also by Sadighi Firozabadi and Sergot [16] who argue that a mere permission given by the resource provider to access a resource which it controls must be distinguished from the *entitlement* to access the resource: an agent is entitled and not merely permitted when the policies regulating the virtual community prohibit the resource provider not to permit the agent to access the resource.

Another distinction comes from deontic logic. Jones and Sergot [17] distinguish permissions from powers in the sense of having been delegated the *institutional power* to do something: "when we say that the Head of Department is authorized[1] to purchase equipment, we mean first and foremost that he has been granted by the institution the power to enter valid purchase agreements". Instead, "sometimes when we say that an agent is authorized to do such-and-such we mean no more than that he has been granted permission to do it".

Law studies argue that a further distinction must be drawn also in this last sense of the term authorization as mere permission. The [8]'s dictionary of law argues that adding or removing an authorization does not change the normative status of an agent while a new permission does; i.e., authorizations do not change the norms (prohibitions and permissions), an agent is subject to; rather, authorizations lift the legal obstacles and limitations, thus legitimating an action of the agent: they change the sphere of what is prohibited or permitted to the agent without adding or removing norms. This is possible since norms have a conditional character so that what is currently considered as a violation or not depends on which are the norms that have their conditions satisfied in the current situation. The fact that authorizations do not modify the existing norms, but change what is prohibited and permitted to an agent anyway, means that authorizations enable the conditions of some permissions. Hence, the institutional power to authorize can be delegated to other agents who do not directly control the resources, since creating an authorization does not require to change prohibitions and permissions.

Some scholars argue, instead, that the power to create permissions can be delegated. [18], for example, propose a framework where this power can be delegated as any other power to create institutional facts. We show in this paper that once prohibitions and permissions are not considered as primitive logical entities, the preconditions for their creation emerge. When we define them in terms of goals of the normative role, it emerges that controlling a resource is necessary for issuing a norm.

In Section 1, we highlighted that according to Cambridge Advanced Learner's Dictionary [7] *officiality* seems to be the first dimension distinguishing permissions from authorizations: the official character of authorizations depends on the fact that they are *institutional facts*, and this character distinguishes them from permissions; an autho-

[1] Note that Jones and Sergot use the term "authorization" in another sense with respect to this paper, i.e., as a synonym of "having the institutional power".

rization is an institutional fact which appears among the conditions of some permission issued by a normative role: if the normative role believes this fact, the permission is enabled so that what is permitted in the current situation is changed by the authorization. The creation of institutional facts is a commonplace feature of legal systems and norm-governed organizations. According to [17], “it is that particular agents are empowered to create certain types of states by mean of the performance of specific types of acts. Typically, the states created will have a normative character”.

An authority has been *delegated the power to create an institutional fact* if the institution recognizes the authority’s action as *counting as* something else (as in Searle [10]’s notion of construction of social reality). E.g., the fact that an authority declares an agent authorized counts as an authorization by a normative role. For Jones and Sergot [17], the counts as relation expresses the fact that a state of affairs or an action of an agent “is a sufficient condition to guarantee that the institution creates some (usually normative) state of affairs”. They suggest this relation can be considered as “constraints of (operative in) [an] institution”, and they express them as conditionals embedded in a modal operator.

5 Summary

We discuss policies for virtual communities based on peer-to-peer systems and the grid infrastructure with a community authorization service (CAS). Pearlman *et al.* [1] use the term ‘right’ both for the authorizations provided by the CAS and the permissions granted by the resource providers: “the user effectively gets the intersection of the set of rights granted to the community by the resource provider and the set of rights defined by the capability granted to the user by the community.” We base our model on the distinction of the notions of permission and authorization, which leads to three roles for each agent: as a resource provider, a resource consumer and authority. The role played by the CAS in virtual communities is formalized in terms of what we called the authority role.

The task of *authorizing* requests performed by the authority CAS is not identical to the task performed by a resource provider when it *permits* access. Permissions and authorizations are distinguished, because authorizations can be delegated. They are related to prohibitions by the resource providers. The relation between the CAS and resource providers is that resource provides can be sanctioned. A resource provider can delegate to the CAS the power of authorizing resource consumers by declarations. The power to issue permissions cannot be delegated, because issuing permissions is restricted to control of the resource.

There are several issues for further research. First, delegation of authorization should be regulated, since not all authorities can be authorized themselves. Second, we can model hierarchies of policies to represent norms issued by local resource providers which can be constrained by obligations and permissions posed at the global level [19, 20]. In [5] we discuss the counts as relation and constitutive rules in normative systems. Finally, in [21] we explore how to formalize our model using the standard BDI$_{CTL}$ logic [22] for agent verification.

References

1. Pearlman, L., Welch, V., Foster, I., Kesselman, C., Tuecke, S.: A community authorization service for group collaboration. In: Procs. of the IEEE 3rd International Workshop on Policies for Distributed Systems and Networks. (2002)
2. Moffett, J., Sloman, M.: Policy hierarchies for distributed systems management. IEEE Journal of Selected Areas in Communications **11(9)** (1993) 1404–1414
3. Samarati, P., De Capitani di Vimercati, S.: Access control: Policies, models, and mechanisms. In Focardi, R., Gorrieri, R., eds.: Foundations of Security Analysis and Design LNCS 2171. Springer Verlag, Berlin (2001)
4. Sadighi Firozabadi, B., Sergot, M.: Contractual access control. In: Procs. of Workshop of Security Protocols, Cambridge (UK) (2002)
5. Boella, G., van der Torre, L.: Regulative and constitutive norms in normative multiagent systems. In: Procs. of KR'04. (2004) 255–265
6. Jin, X., Liu, J.: The dynamics of peer-to-peer tasks: an agent based perspective. In: Procs. of Agents and Peer-to-peer Computing. (2004) 84–95
7. Press, C.U.: Advanced Cambridge Learners' dictionary. Cambridge University Press (2002)
8. del Giudice, F.: Nuovo dizionario giuridico. Simone Editore (2001)
9. Makinson, D.: On the formal representation of rights relations. Journal of philosophical Logic **15** (1986) 403–425
10. Searle, J.: The Construction of Social Reality. The Free Press, New York (1995)
11. Broersen, J., Dastani, M., Hulstijn, J., van der Torre, L.: Goal generation in the BOID architecture. Cognitive Science Quarterly **2(3-4)** (2002) 428–447
12. Lang, J., van der Torre, L., Weydert, E.: Utilitarian desires. Autonomous Agents and Multiagent Systems (2002) 329–363
13. Gmytrasiewicz, P.J., Durfee, E.H.: Formalization of recursive modeling. In: Procs. of ICMAS'95. (1995) 125–132
14. Boella, G., van der Torre, L.: Permissions and obligations in hierarchical normative systems. In: Procs. of ICAIL'03, ACM Press (2003) 109–118
15. Li, N., Grosof, B.N., Feigenbaum, J.: Delegation logic: A logic-based approach to distributed authorization. TISSEC **6(1)** (2003) 128–171
16. Sadighi Firozabadi, B., Sergot, M.: Power and permission in security systems. In: Security Protocols. Springer Verlag (1999) 48–53
17. Jones, A., Sergot, M.: A formal characterisation of institutionalised power. Journal of IGPL **3** (1996) 427–443
18. Sadighi Firozabadi, B., Sergot, M., Bandmann, O.: Using authority certificates to create management structures. In: Procs. of Workshop of Security Protocols. Volume 2467., Berlin, Springer Verlag (2001) 134–145
19. Boella, G., van der Torre, L.: Local policies for the control of virtual communities. In: Procs. of IEEE/WIC WI'03, IEEE Press (2003) 161–167
20. Boella, G., van der Torre, L.: Local vs global policies and centralized vs decentralized control in virtual communities of agents. In: Procs. of WI'04. (2004)
21. Boella, G., van der Torre, L.: Game specification in normative multiagent system: the trias politica. In: Procs. of IAT'04. (2004)
22. Rao, A.S., Georgeff, M.P.: Decision procedures for BDI logics. Journal of Logic and Computation **8** (1998) 293–343