A Model for Security Policy Management In MASs
(An Architecture for Security Policy Management in MASs)
(Access Control Policy Management and Negotiations in MASs)

Fatih TURKMEN

Department of Information Technologies
University of Trento

Abstract
I will be discussing the security policy negotiations between agents and the environment. Agent interaction is a definitive feature of multi-agent systems. As interaction involves multiple parties exchanging information and requesting/providing a particular service, there is a need of a matching mechanism between the agents' requirements. This matching mechanism will be used for the matching of the communication language, policy matching etc. Security requirements, specifically privacy policy matching is an important concern.

A trust relationship can be at hand, but an agent environment can be so broad that no trust relationship can be deployed. Even in cases where it is possible to establish trust relationships, a number of security requirements must be matched for communication between agents.

Introduction

The proliferation of agent applications and analogous (computing paradigm) implementations revealed the importance of security more in MASs (Multi Agent System). This is heavily supported by the widening of the agent definition as it caused the application domain expansion of MAS. As a result, security of MASs has got more attention of researchers in the field, not only on the communication security but also on the access control and the usage.

Comment [f1]: I call analogous implementations because the security issues are very similar with each other in them, Especially with web services and agents

As an important part of security studies, access control deserves high priority in a system to be called secure. To preserve the confidentiality and deny disallowed actions, the security concerns are presented in the forms of policies. In general terms, the policies are the views of the entities that they are defined on and provided to the interested parties by the owner or the manager of the entity. From a security perspective, policies basically arrange the access rules and the possible actions on the entity in a system.

Although policies are issue-specific and can vary according to the need and the domain, the policy types defined in [1] for the Web can cover at least the current needs of MAS applications. These policy types include Privacy, Rights, Identity, Access Control and the

Context. Despite they are defined separately, they have so much in common in their meaning.

Access control policies deal with the access permissions to the resources and the operations on them. In a MAS setting, they are the governing rules of agent actions in the environment and the plans (intentions) of the agent to achieve its goals. In this respect both the agent and environment come up with a policy.

Policy Negotiations

Negotiations in a system are done because of mutual benefit of involved parties in the negotiation process. The benefits of negotiations are obvious from real world applications in international relations, economy and business processes etc. In a MAS environment, typically with dynamic features, negotiations must take place to arrive at an agreement for the intended mutual benefit.

In our setting, the policy negotiations can take in several forms according to the negotiators involved:

- Agent ←→ Agent
- Agent ←→ Environment

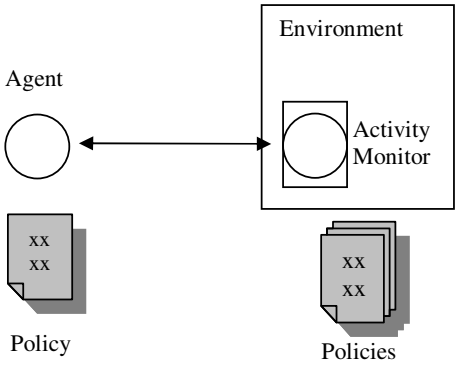

Figure 1. Agent – Environment Policy Negotiation

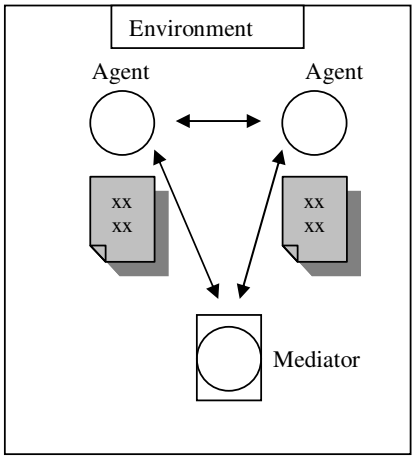

Figure 2. Agent – Agent Policy Negotiation

Policy negotiations are handled by two active entities in a MAS: Activity Monitor and Mediator. Their definition is a design choice. They can be either an artifact as part of the environment or an agent with a dedicated function.

The forms of policy negotiations are very similar to each other in the sense there are two or more policies at hand and a negotiation process is happening on them. After an agreement is reached the enforcement is continuously done.

As artifacts are part of the environment, everything related with the artifacts are handled in the environment policy. Depending on the meaning loaded on the notion of artifacts, they are the building structures and the internal dynamics of the environment.

Related Work

In [2], the main approach for the access control is treating users as agent classes which play the roles assigned to them. The role depends on the context and the coordination. For that reason, there is one-to-one match between the role and agent coordination context (ACC). Agent coordination context is composed of the rules and the actions managed by these rules. The rules are determined by both the other agents and the artifacts within the environment. For that reason, whenever there is an agent entering to the environment he is asked for authentication to be enabled to play a role, he wants/needs to provide some action requests that should be available within the environment. These requests are in the form of policies that present the requirements of the agent for the environment. On the other hand, whenever there is a request for an ACC (role) from an agent, the environment has also some requirements in the form of policies.

ACCs are studied in a wider perspective and extended to a mechanism for the organization and security support for MAS environments in [3]. They are defined with three driving aspects; Interface, Contract and Configuration in which the interfaces define the possible operations in the environment, the contracts are the description of the relations between the agents and the environment and configurations are the run-time parameters revealing the ACC state. With this broader view, it becomes reasonable to use RBAC (Role Based Access Control) approach for security as the correspondence between the definition of ACC and the role is clear.

The policies can be thought in two ways:

- If RBAC is selected as the security model in which agents play certain roles in the environment, then the notion of policies constitute of the requirements and the rules to be met. According to these, an agent is either allowed or denied to play a role in the environment. In brief terms, these requirements and the rules are the policy entities which must be compromised between the agent and the environment. The involved parties in this agreement process (agent and environment) can have certain levels of rigidity in their policy negotiations as a whole or partly. (SOME MORE COMMENT HERE) After the agent is enabled to play a role, the control mechanism is not finished. As stated in [4], access control in general must be done in a continuous manner, especially in environments which have dynamicity as one of the main factors. UCon (Usage Control) is the approach to be deployed in dynamic environments.

  For that reason, policy management is not working in a “do it once, use it forever” manner in agent environments. Instead, there is a continuous checking of access control and policy enforcement. Also even when an agent is playing a role in the environment, there can be new policies raised for specific actions (FIND ANOTHER TERM) and new negotiations for these new policies.

- If there is no RBAC or any other security model (WHAT KIND OF MODEL), then the policies of the agent and the environment just reflect the access and usage of the resources which are affected by the actions. The policy of the agent represented in a

certain policy specification language is composed of actions that are desired by the agent to reach its goals. The policy of the environment determines the rules of resources usage (in what and how way of use). (example, soccer game, field and the stands are the environment, players are the agents) and allowed actions in the environment.

Policy Management Architecture

As an important feature of agents, autonomy makes the access control difficult in the sense it puts the environment into indeterministic states. From the perspective of the environment, the more static (stable) the environment, the easier to manage. Unpredictability is not only a matter of agent but also an environment issue. Because of agent actions or internal dynamics of the environment, there can be modifications in the environment which requires updates in policies. New resources can enter into the environment or leave the environment. To manage policies within the environment, there should be a central, environment specific policy manager with the following tasks:

- *Environment Monitoring*: Continuous monitoring of the environment is done by the policy manager to hold a clear picture of the resources existing in it.
- *Policy Management*: Environment policies shaped according to all resources in the environment are stored by the policy manager. Policy modifications (updates) are managed by the policy manager. The environment changes must be reflected in the environment policy. Basically, it acts as a repository of policies. It stores all modified policies resulted after negotiations which are specific for the agents.

Activity Monitor is responsible for two tasks:

- Negotiations are managed by activity monitor prior to or in the environment. It consults to the policy manager for the policy details.
- Controlling accesses to the environment resources. It uses environment policies for the usage management.

The Mediator defined in agent-to-agent negotiations is an implicit part of Activity Monitors. Its main function is managing the negotiation process on the policies they got from their negotiators, possibly proposing solutions for the problems that can occur during the negotiations and distributing the resulting policy after the process to the negotiators. As shown in Figure 4, Activity monitor is composed of two components; the Action Watcher is responsible for the environment tracking and the Mediator as defined before.

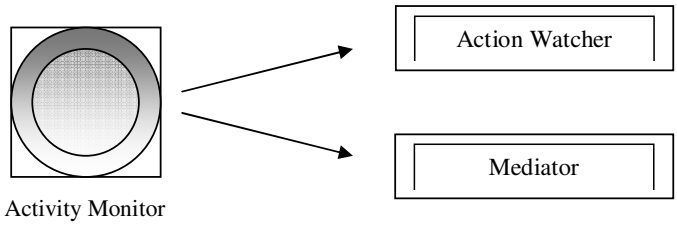

Figure 4. Activity Monitor

With the use of activity monitor, a forbidden access to the resources of the environment is prevented. On the hand, negotiations with the agents can provide some type of flexibility in the resource usage. Also up-to-date policies can handle unexpected agent activities and environment changes.

Policy manager as the center for security policy within the environment provides a centrally managed access control mechanism.

Figure 5 briefly summarizes the processes related with the policies in a typical MAS application.

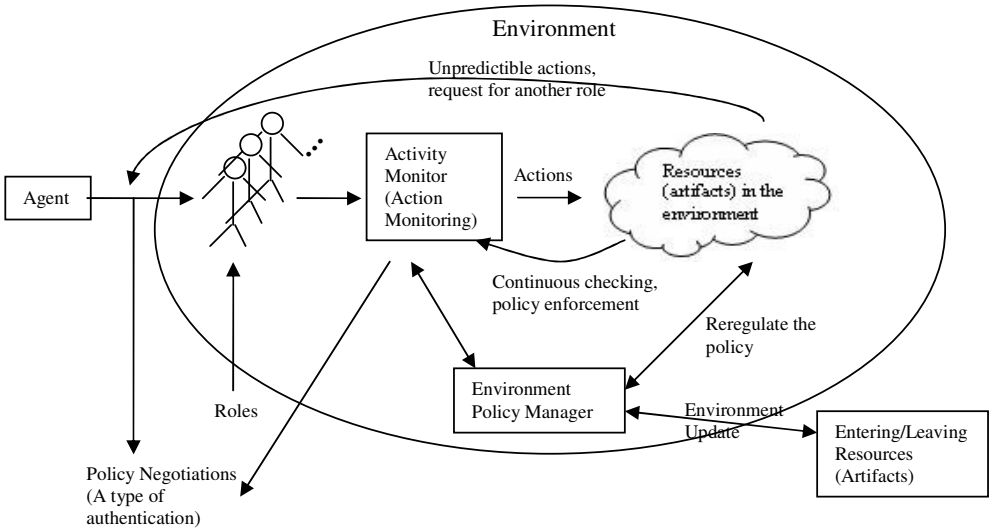

Figure 5 Policy Exchange/Negotiation/Monitoring Processes

The benefits of this access control approach in agent studies give the following benefits:
The notion of the role creates a good abstraction for the access control. All the advantages of RBAC model are obtained in one hand such as separation of duties and role hierarchies (THINK ABOUT IT). Additionally, regarding artifacts as resources gives a clear idea of protection which are the action transport elements of agents. In this way, agent effects to the environment are kept under control.

AOC (Agent Oriented Computing) Abstractions
The following AOC abstractions are done and used in the policy specification of the architecture provided:

- Artifact → Any resource in the environment can be regarded as an artifact. The difference between traditional artifact definition and our definition is the concreteness of the existence. In traditional definition, artifacts are concrete objects. But in our

definition artifacts can be both abstract (like a feature of an object) and concrete. Examples of resources in the environment according to our definition can be given as follows,

- Abstract: Data/Information (Any information which can be put into confidential category, a file, personal information etc.)
- Concrete: Any object within the environment (a table, a drawer)

- Agent → Agents percepting/acting in the environment. Agents are also effected by the actions of other agents (DISCUSS THIS). The main features of agents are from AOC and are bounded after policy negotiations. After negotiations they are put into the shape of roles whose actions are regulated by the policy agreed. Examples: A clerk, a travel agency etc.

- Actions → From agent perspective, actions are any acts that lead them towards their goals. From environment perspective, they are the reasons that can cause any change (state change) inside them, any attempt to use a resource or .

Example Policies (Policy Tuples or Logical Expressions for Access Control – WHICH ONE)

Roles (Agents)	*Resources* (Artifacts)	*Permissions*	*Context*
Clerk	Computer	Read	

Conclusions and Future Directions

In this work we presented an architecture for the management of policies in MASs occupied in an environment. This architecture deals mainly with access control policies but can be extended / converted to other policy management systems.

There are two entities introduced in the paper for handling policies in agent applications; Activity Monitor and Policy Manager. Activity monitor watches the agents in the environment for possible rule breakings and Policy manager acts as a repository for the policies.

As a future direction there are two topics to be discussed:

- Possible extension of Respect (Tucson) for security policies in MAS applications
- Authentication of agents: The discussion of agent authentication in a space can result with different findings as they can have different credentials from other authentication domains in which there are usernames-password pairs or fingerprints as examples.

REFERENCES

1. Iannella R., Henricksen K., Robinson R., "A Policy Oriented Architecture for the Web: New Infrastructure and New Opportunities", W3C Workshop on Languages for Privacy Policy Negotiation and Semantics-Driven Enforcement, October 2006 - Ispra, Italy

2. Viroli M., Omicini A., Ricci A., “An organisation infrastructure for Multi-Agent Systems based on Agent Coordination Contexts”, AI*IA 2005: Advances in Artificial Intelligence, 9th Congress of the Italian Association for Artificial Intelligence (AI*IA 2005). LNAI 3673, Springer, September 2005.

3. Ricci A., Viroli M., Omicini A., “Agent Coordination Contexts in a MAS Coordination Infrastructure”, Applied Artificial Intelligence, 2006.

4. Park J., Sandhu R., "The $UCON_{ABC}$ usage control model", ACM Transactions on Information and System Security (TISSEC), 2004.