Sixth IFIP Conference on Intelligence in Networks (SmartNet'2000), Vienna, Austria, September 18 - 22, 2000

A Multi-agents System for Network Security Management

K. Boudaoud and Zahia Guessoum
[1]*Institut Eurécom*
2229, route des crêtes - 06904 Sophia Antipolis France
Phone: 33 (0) 4 93 00 26 38 / Fax: 33 (0) 4 93 00 26 27

[2]*LIP6-OASIS*
Case 169, 4 place Jussieu, 75252 PARIS Cedex 05 - France
Phone: (33) 1 44 27 87 43 / Fax: (33) 1 44 27 70 00

Abstract: The openness of business toward telecommunication networks in general and Internet in particular is performed at the prize of high security risks. Every professional knows that the only way to secure completely a private network is to make it unreachable. However, even if this solution was undertaken for many years, nowadays it is not possible to close private network especially for business purpose. Thus, security management becomes an important issue that must be considered carefully. The focus of our work concerns one critical security management issue that is intrusion detection. Some drawbacks of existing systems reveal the necessity of designing a new generation of self-adaptive systems. In fact, self-control, flexibility, adaptability, autonomy and distribution are the main features to be addressed in a suitable architecture that fulfils these requirements. The introduction of multi-agents system (MAS) in a network seems so promising to enable network entities to perform adaptive and "intelligent" behavior. "Intelligence" means that network entities provide reasoning capabilities, exhibit behavior autonomy, adaptability, interaction, communication and co-operation in order to reach specified goals. In this context, we propose a new approach for security management using intelligent agent (IA) technology. This approach provides a flexible integration of multi-agent technique in a classical network to enhance its protection level against inherent attacks.

Key words: Intelligent agents, multi-agents system, network security management, intrusion detection, distributed network management.

1. INTRODUCTION

During these last years, computer systems and networks have not ceased evolving, particularly in terms of number of users and offered services that are continuously increasing in complexity. Systems and networks become more complex (number of machines, number of users, number of connections...), making them more vulnerable to various kinds of complex security attacks. Therefore, security management of these systems and networks, particularly intrusion detection, requires more sophisticated models.

To deal with these requirements, multi-agents systems (MAS) are well adapted. They provide a powerful paradigm for the modeling and the development of complex systems. They are based on the decomposition of systems into several interacting and autonomous entities called agents. An agent refers to an entity that functions continuously and autonomously in an environment in which other processes take place and other agents exist. Recent applications show the growing interest of this paradigm in the network management domain [1].

This paper deals with the use of multi-agents system to detect security attacks. We first discuss some requirements for an efficient security management. In section 3 and 4, we describe the organizational model of our MAS and the functional model of a security agent. We give an overview on our implemented security management system in section 5. Finally, we conclude with some remarks and future work.

2. SOME REQUIREMENTS FOR AN EFFICIENT SECURITYMANAGEMENT

Security management aims to maintain the integrity, confidentiality and availability of systems and services. Securing a network involves protecting it against all possible attacks. But, in practice it is not possible to have a completely secure network. So, applying security management is a two-fold activity: 1) the security architecture is to be deployed to protect networks by detecting attacks; and 2) when attacks are detected the security architecture deals with these attacks in real time by taking security measures.

An attack can be defined as any non-standard activity, which compromise the information *confidentiality*, data *integrity* and *availability* of a resource. There are various kind attacks that can be classified in *Network attacks***,** *System attacks and Web attacks* [2]. In this paper, we are interested in *Network attacks*, such as: *ICMP flooding*, *doorknob* and *Ping sweep* [2]. The aim of IDSs is to detect security attacks, especially in real-time fashion.

Among the existing systems, we can cite DIDS (Distributed Intrusion System Detection) [3] and CSM (Co-operating Security Manger) [4]. DIDS was designed to supervise a local area network LAN. Its centralized control represents a major disadvantage. In the case of WAN networks, where the communications with the entity manager, it can congest the network. CSM was developed for a distributed environment. However, it cannot be easily adapted to new environment. To enhance the security of computer systems and networks, we must deal with their distributed nature and dynamics, which are two important characteristics. The dynamics of networks is due to their increasing evolution in terms of offered services, used resources and number of users. In fact, users, known or not, have complex behaviors that vary considerably. Particularly a recent aspect, viz. the mobility of users, enhances this complexity. Once a user, known in a static network, moves, the knowledge related to his behaviors becomes different. This complexity and dynamics of networks make them more vulnerable to various kinds of security attacks. Therefore, the security policies may change over time. Moreover, some requirements are important for detecting attacks efficiently:

- **Distribution**: Many network attacks are characterized by abnormal behaviors at different network elements [7]. Detecting them by a single system, running on a single component, is too complicated. So it is easier to distribute monitoring and processing tasks among a number of entities at different points. This important aspect is provided by most existing IDSs [3]. For example, in DIDS, data collection is assured by several entities but a centralized director performs the analysis.
- **Autonomy:** Excessive data traffic between distributed entities can cause network congestion problems. So, it is more judicious to let the entity monitoring a network element perform local analysis and detect intrusive behaviors. Thus, distributed entities must be autonomous. The CSM approach has shown the necessity to use ***autonomous*** entities [4]. In CSM, there is no established central director but individual managers that are responsible of making local intrusion detection.
- **Delegation:** The high level of dynamics in networks requires modifying, at any time, security management functions to adapt them to changes occurring in the monitored network. The model of delegation among various management entities allows fulfilling this requirement. The delegated tasks are sent to the autonomous entities. Each one has to execute its own task. When new tasks must be added or existing one must be modified, this is done dynamically. The *Delegation* feature is not found in existing IDSs.
- **Communication and cooperation**: Coordinated attacks cannot be detected easily by an individual entity, which has a restricted view of the network. It is therefore necessary to correlate various analyses made by the autonomous entities. So communication and cooperation between

entities are needed to detect coordinated intrusive behaviors. In the CSM system, each security manager detects local intrusions and cooperates with other CSMs by exchanging information in order to detect cooperatively intrusive activities [4].

- **Reactivity:** The aim of efficient intrusion detection is to react against an attack before serious damages can be caused.
- **Adaptability** and **Flexibility**: When a new security policy is added or modified, intrusion detection and monitoring tasks must be adapted. Moreover, when new services and resources are added, the IDSs must consider these variations dynamically. IDSs must also detect new attacks when they happen. So it is important to learn new patterns of attacks. These two features are not provided by existing systems, which can not be *upgraded easily* and cannot *easily adapt* their intrusion detection tasks to changes in networks and user behaviors. In addition, they do not have the ability to learn new attacks.

The above-described features are considered as the main requirements for detecting attacks efficiently. They show that multi-agents systems are a suitable solution. Multi-agent system properties (distribution, cooperation...) and agent properties (adaptability, autonomy, pro-activity...) [1][13] match the whole requirements.
The adopted approach for designing our MAS is based on two levels [8]:

1) A **Macro** level, which describes the organizational and functional structure of the MAS;
2) a **Micro** level, which describes the architecture of a security agent.

The two levels are described below.

3. THE ORGANIZATIONAL MODEL OF THE PROPOSED MULTI-AGENTS SYSTEM

The *Macro* level defines the MAS organization. The latter defines a set of roles and relations between them [9]. A role can be defined as a set of tasks that an agent must do in order to make the organization reaching its objectives. To describe the different roles, it is necessary to identify the tasks that must be done by the MAS.

3.1 Identification of roles

We distinguish two types of tasks: **monitoring and management.** The monitoring tasks are identified correspondingly to the kind of activities to monitor. We identify five types of *monitoring*:

- *External monitoring* for the monitoring of external activities;
- *Exranet monitoring* for the monitoring of *extranet* activities;
- *Intranet monitoring* for the monitoring of *intranet* activities;
- *Internal monitoring* for the monitoring of internal activities;
- *Local monitoring* for the monitoring of local activities.

And two type of *management* tasks:

- *Policies management* for security policies management of the network;
- *Security management* for security management of a distributed or local area network.

According to the previous tasks, we identify the following roles:

- **Security policy manager** manages the security policies and communicates with the security officer.
- **Extranet manager** describes the security management functions of a distributed network. These functions concern the detection of complex attacks happening in a high level. Thus, an agent having this role, will have a global view of the network and will detect coordinated attacks. It also will specify monitoring and detection tasks to the low-level agents. This role manages the security of the distributed network with external networks and between LANs of the distributed network.
- **Intranet manager** manages the security of the LAN constituted of several domains. It concerns activities monitoring and detection of coordinated attacks within a LAN and between its various domains.
- **Local extranet monitor** includes *external* and *extranet monitoring* functions within the LAN. This role is associated to the detection of attacks originating or in direction to an external or extranet network.
- **Local intranet monitor** represents *internal and* i*ntranet monitoring* functions within the LAN. It concerns also detection of attacks directed or originating from other LANs of the same distributed network.
- **Local internal monitor** that defines the *local monitoring* functions. It concerns the detection of attacks, which are local to a domain.

3.2 Organizational structure

The proposed security management architecture (MANSMA)[1] consists of several agents structured hierarchically. Agents, which have various roles, are located at specific network entities and distributed at different points of the network. The hierarchical organization of agents enables local as well as global intrusion analysis and detection. Each agent has its own perception of the network, which is limited by the domain to monitor. In MANSMA, we

[1] Multi-agents System-based Network Security Management Architecture

distinguish, accordingly to agent roles identified in the previous section, two functional layers: a *manager layer* and a *local layer* .

- The **manager layer** manages the global security of a network. In this layer we identify three levels of *manager agents*: a *security policy manager agent*, an *extranet manager agent* and several *Intranet manager agents.* The *extranet manager agent* controls *intranet manager agents*, which report pertinent analysis. It performs then another analysis to confirm the detection of an attack. It can also ask for more data processing and delegate new monitoring tasks to the *intranet manager agents.* The *extranet manager agent* is also responsible for distributing a set of *local agents* to each *intranet manager agent.* The *intranet manager agent* controls *local agents* and analyses the monitored events reported by these agents.
- The **local layer** manages the security of a domain. It is composed of a group of *local agents*, which have specific monitoring roles. We distinguish three kinds of *local agents*: *extranet local agent, intranet local agent* and *internal local agent.*

Domains are defined by the agents of the *manager layer*. In the *local layer*, a domain represents a group of sub-hosts, which are gathered either according to the organization chart of the company in terms of departments or according to security levels specified by the security policies of the company. In the *manager layer*, a domain represents either a distributed network or a LAN of this same distributed network.

In this hierarchical multi-agent model, each *manager agent* has the ability to control specified agents and to analyze data, whereas the *local agents* monitor specified activities. In each level, agents communicate and exchange their knowledge and analysis for detecting intrusive activities in a cooperative manner. The interaction between these two layers allows the detection of global attacks by correlating the various analyses of the *local layer*.

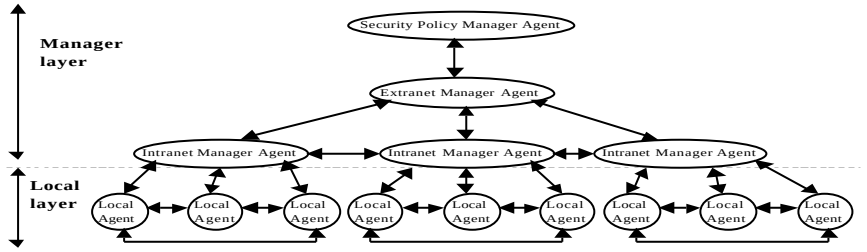

Figure 1: MANSMA Functional Architecture

4. THE AGENT MODEL

To model intrusion detection, agents must combine cognitive abilities (knowledge-based) to reason about complex attacks with reactive capacities (stimulus-response) to react rapidly to the environments changes. So, an agent has three functions: 1) a *filtering* function that filters security events, 2) an *interaction* function that manages its interactions with its environment and other agents and 3) a *deliberation* function that enable it to analyze new data and detect attacks. These functions are described in the following paragraphs.

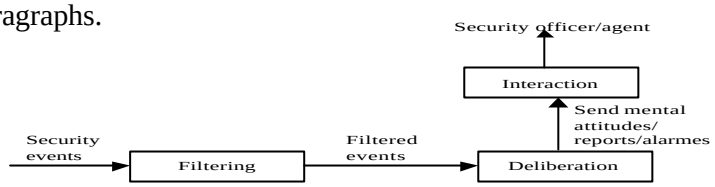

Figure 2: Interactions between agent functions

4.1 Event filtering function

A *security event* is characterized by its type, its observation point, a temporal attribute (representing the event occurring moment), and a set of non-temporal attributes. According to the event type and its observation point, we identify various event classes (see diagram below).

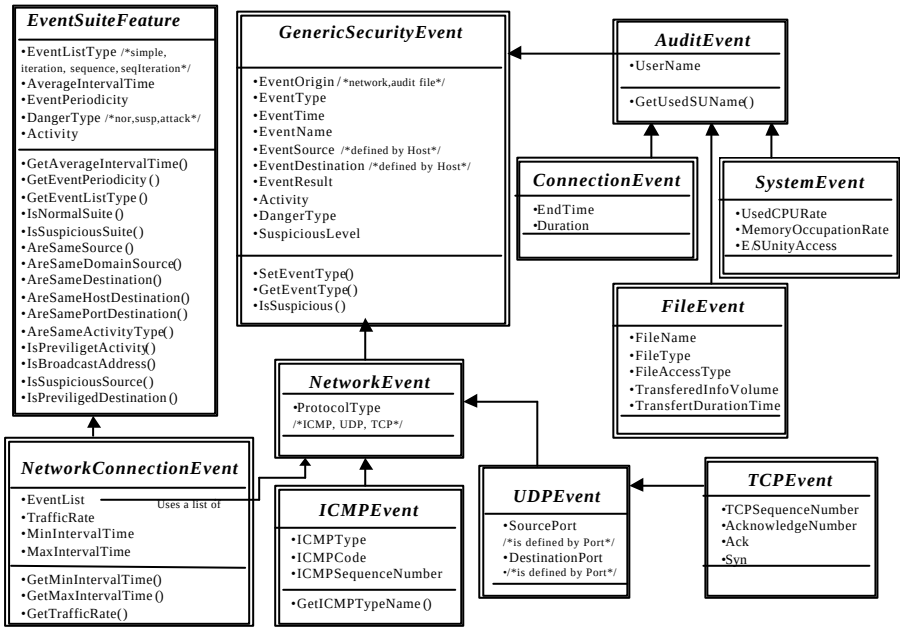

Figure 3: UML classes of security events

The *event filtering* function filters security events produced in the network, according to event classes specified in a detection goal. Indeed, the events occurring in network are not all collected. In fact, when a detection goal is sent to an agent, a set of event classes to observe is specified to it. Thus, when an event occurs in the network, the agent tests if it matches the event classes specified in the goal. If it matches, it is collected. The filtered events are then stored, waiting to be treated by the deliberation function.

4.2 Interaction function

This function describes interactions between the above-described agents. It allows them to communicate their analyses and knowledge and mental attitudes (beliefs, suspicions…). In fact, m*anager agents* interact with *local agents* by:

- sending goals, derived from security policies;
- delegating specific functions of monitoring/detection and specifying the various domains to monitor;
- asking particular information: the suspicion level of a specific user, the list of events generated by a user, etc.;
- and receiving the relevant reports or analyses results and alarms.

Interaction function also permits interactions between the security officer and s*ecurity policy manager agent/ extranet manager agent.* It ensures the reception of specifications and requests from the security officer such as security policies to apply. It allows the delivery of security reports and alarms when an attack is detected. The security officer can also ask for additional information (asking for the current security state of the network, the list of suspicious users…).

4.3 Deliberation function

As it has been outlined in section 2, security management must deals with significant network characteristics such as: 1) its continuous variation, particularly in terms of users and offered services; 2) and variation of its security problems such as new vulnerabilities and increasingly complex attacks. Considering the unpredictable character of the agent environment behavior (network), we adopted a BDI solution [10][11] for modeling the security management system. Thanks to the *deliberation* function the agent is able to reason and extrapolate by relying on its mental attitudes, built knowledge and experience, in a rational way, to find the adapted answers. The agent uses its beliefs resulting from the filtered events and beliefs of the neighboring agents for reaching its specified goals. When a goal is reached (an attack is detected), it executes appropriate actions.

In this section, we will start by describing the knowledge base of the agent and then the BDI-based information model.

4.3.1 Knowledge base

The knowledge base of the agent contains two types of knowledge:

- **Immediate character knowledge** that represent the observations made by the agent (events produced in the network) on its environment. This knowledge has a limited validity lifetime;
- **Permanent character knowledge**, which represent the necessary knowledge for managing security of the network (such as list of known user/user groups, list of administrators, list of known hosts, list of known addresses, prohibited addresses, reserved addresses...).

4.3.2 BDI-based information model

This model represents the mental attitudes of the security agent: *beliefs, goals, intentions,* su*spicions* and *policies.*

4.3.2.1 Beliefs

Beliefs represent the perception that the agent has on the network behavior and its security state. They indicate also knowledge that it has on other agents and itself. We distinguish three types of beliefs:

- **Personal beliefs** express its knowledge on its own state (information relating to it, in particular the domain, which it must monitor).
- **Relational beliefs** represent what the agent knows on other agents with which it communicates. These are all information (role, competencies...) that it needs to communicate with them
- **Environmental beliefs** include *local environmental beliefs* and *environmental beliefs of the others. Local beliefs* indicate what the agent believes on the behavior and the security state of the network whereas the *beliefs of the others* represent perceptions which have the other agents on the network. Within the framework of this project, we distinguish two types of *environmental beliefs*:
  - **Schema beliefs**, which are a description of attack scenarios to detect. These beliefs will not be instantiated until a goal is sent;
  - **Scenario beliefs,** which represent the sequences of security events. A *scenario belief* is associated to one or *several schema beliefs. Scenario beliefs* have a temporal validity, which depends on the temporal validity of the events constituting the event sequence

4.3.2.2 Goals

Goals represent the state that must be reached by the agent viz. its objectives. We distinguish three types of goals:

- **Monitoring goals**, which ask for monitoring specific activities (activities of a certain user, going-in activities...).
- **Informational goals** that request specific information on the security state of the network (detected attacks during a certain period of time, suspected users, current external connections...);
- **Detection goals**, which specify the attacks to detect and the measures to take if an attack is detected. They are the most significant goals within the framework of intrusion detection. A detection goal allows the instantiation of a *schema belief.*

4.3.2.3 Intentions

Intentions represent the list of actions that must be executed by the agent when it achieves its goal. These actions can be: sending alarms to the security officer or manager agent, closing a connection established by an attacker, reconfiguring a firewall...

4.3.2.4 Suspicions

This mental attitude, introduced within the framework of intrusion detection, expresses the suspicion that has an agent on a *scenario belief*. When an agent observes a sequence of events which corresponds neither to a normal sequence, nor to a known attack, then it identify it as suspicious sequence. To confirm that this *suspicion* is an attack, the agent needs further information or confirmations from other agents. The agent will then say to other agents: *"I suspect that this sequence of events is an attack"*. A *suspicion* is associated to a *schema belief* and is the result of the analysis of a *scenario belief* compared to a *schema belief.*

4.3.2.5 Policies

Policies represent the guiding mental attitude of the MAS behavior to manage the security of the company. Starting from the specified security policies, a set of *goals* are created and derived in order to maintain a certain security state of the network.

5. IMPLEMENTATION

The presented agent model has been implemented with the multi-agent platform DIMA [13]. The latter is mainly characterized by a modular agent

architecture. DIMA proposes the extension of the single behavior of an active object into a set of behaviors. In our implementation, each agent has three behaviors:

- The *filtering* behavior filters security events. When an event occur in the network, it is collected only if it matches the event classes specified in the detection goal.

  EventFilter {
  Repeat
  security-event := get(security-event-to-filter);
  If is-in-list-of-event-types-to-filter(security-event)
  Update-list-of-filtered-event(security-event);
  end repeat }
- The *interaction* behavior manages the interaction between the agent and the other agents. It defines the mailbox of the agent and the way the messages are received and enqueued for later interpretation. An agent may need some others information to refine its analysis. In this case, it asks other agents to give it the necessary information.
- The *deliberation* behavior represents beliefs, goals, intentions and knowledge of the agent. It is responsible 1) for generating adequate responses to the messages received from the other agents and 2) for achieving the agent goal(s).

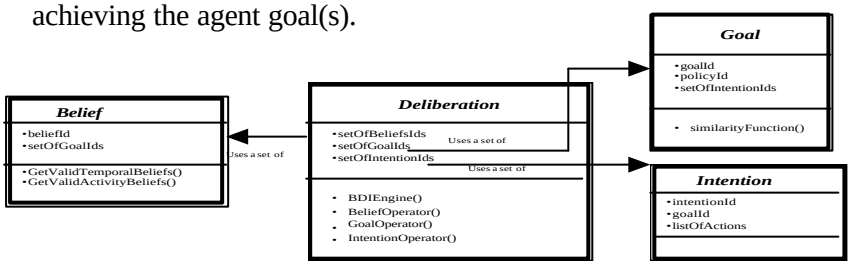

Figure 4: UML classes used by the deliberation function

When an agent receives a *detection goal,* it updates a set of event classes to filter. Then, when an event occurs it is filtered by the filtering module and sent to the deliberation module. This one, updates/creates agent *scenario beliefs* and then test if this belief matches a *schema belief.* If it matches, then a *detection goal* is reached and a list of *intentions* are sent to the interaction module for being executed (see figure 4 and 5).

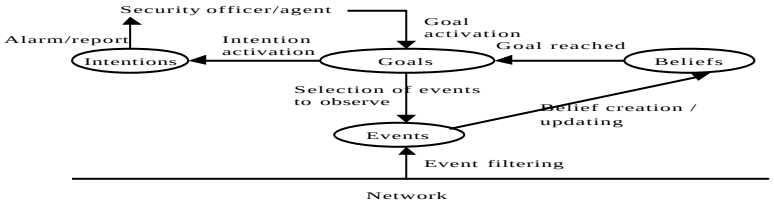

Figure 5: Interactions between mental attitudes

Our implemented system detects well-known attacks such as doorknob rattling, ping sweep and ICMP flooding attacks.

6. CONCLUSION

In this paper, we presented a multi-agents system, which aims to detect intrusions in a complex network. To deal with this complexity, we argued that multi-agents systems provide a suitable solution. So, we applied a well-known multi-agent methodology and showed thus that this methodology is useful for real-life application. Moreover, to model agent knowledge, we used the BDI theoretical model. This model required a hard work to deduce a practical implementation. The implemented system detects well-known attacks. We are now working on a new adaptive version, which deals with learning new attacks and react to non well-specified attacks.

References

[1] R. F. Teixeira de Oliveira, "Gestion des Réseaux avec Connaissance des Besoins: Utilisation des Agents Logiciel", PhD thesis, Eurécom institute, France, 1998.

[2] K. Boudaoud, H. Labiod, Z. Guessoum and R. Boutaba, "Network Security Management with Intelligent Agents", In proceedings of 2000 IEEE/IFIP Network Operations and Management Symposium (NOMS'2000), Honolulu, Hawaii, 10-14 April 2000.

[3] L.T. Heberlein, B.Mukherjee, and K.N.Levitt, "Network Intrusion Detection", IEEE Network Journal, pp. 26-41, May/June 1994.

[4] Maj.Gregory B. White, Eric A. Fisch and Udo W. Pooch, "Cooperating Security Managers: A Peer-Based Intrusion Detection System", IEEE Network journal, pp. 20-23, January/February 1996.

[5] M. Crosbie, E. H. Spafford, "Defending a Computer System using Autonomous Agents". Technical report CSD-TR-95-022, Computer Sciences Department, Purdue University.

[6] S.Corley and al, "The Application of Intelligent Agent Technologies to Network and Service Management", 5th IS&N Conference, Antwerpen, Belgium, 25-28 May 1998.

[7] C. Ko, D. A. Frincke, T. Goan, L T. Heberlein, K. Levitt, B. Mukherjee, C. Wee, "Analysis of an Algorithm for Distributed Recognition and Accountability", 1st ACM Conference on Computer and Communication Security, pp. 154-164, 1993.

[8] J. Ferber and O. Gutknecht , "A meta-model for the analysis and design of organizations in multi-agen systems", ICMAS, 1998.

[9] J. Ferber,"Les Systèmes Multi-Agents, Vers une intelligence collective", InterEd. 1995.

[10] A. S. Rao and M. P. Georgeff, "Modeling Rational Agents within a BDI-Architecture", Technical Note 14, 1991.

[11] A. Rao and M. Georgeff, "BDI Agents:From Theory to Practice", Tech. Note 56, 1995.

[12] M. Wooldridge and N. R. Jennings. "Intelligent Agents : Theory and Practice ". Knowledge Engineering Review, 10(2):115 -152, 1995.

[13] Z. Guessoum and J.-P. Briot. "From Active Object to Autonomous Agents", *IEEE Concurrency*, volume 7 N° 3, pp. 68-78, July/September, 1999.