Towards Secure Interactions In Agent Societies

Bastin Tony Roy Savarimuthu, Martin Purvis, Marcos De Oliveira, Maryam Purvis

Abstract—**Modern workflow management systems driven by multi-agents lack proper security mechanisms. In an agent based workflow system, the resource agents that perform various tasks can form societies. The communication between agents belonging to a particular society should be secure. In this paper we propose mechanisms to authenticate the communication between agents - in particular concentrating on the interactions in multi-agent based workflow societies using Public Key Infrastructure.**

Index Terms— **Agents, Workflow, Societies, PKI**

I. Introduction

MULTI agent systems have been discussed by many researchers for many years[1,3]. Agent based workflow systems have been discussed by some researchers [6,9,10]. A few researchers have concentrated on the security aspects of the agent-based systems [13,14,15]. Formation of agent societies is a new area of research [16,17,19,20] and the security mechanisms for interaction of agents have not been discussed before.

In this paper we describe security mechanisms associated with the interaction of multi agents using the Public Key Infrastructure (PKI).

II. Background

A. Multi-agent systems

Agent systems provide an open, flexible and distributed framework [1,3]. In the context of Workflow Management Systems (WfMSs), agent technology has been used in different ways [4,5]. In some cases the agents fulfill particular roles that are required by different tasks in the workflow. In these cases the existing workflow is used to structure the coordination of these agents. An example of structuring the co-ordination of agents using a workflow process is the work by M. Nissen in which a set of agents have been designed to perform activities associated with the supply chain process in the area of E-Commerce [18].

Manuscript received August 20, 2004. Bastin Tony Roy Savarimuthu is with the University of Otago, Dunedin, New Zealand, P O Box (e-mail: tonyr@infoscience.otago.ac.nz).

Martin Purvis is with the University of Otago, Dunedin, New Zealand P O Box (e-mail: mpurvis@infoscience.otago.ac.nz).

Marcos De Oliveira is with the University of Otago, Dunedin, New Zealand P O Box (e-mail: moliveira@infoscience.otago.ac.nz).

Maryam Purvis is with the University of Otago, Dunedin, New Zealand P O Box (e-mail: tehrany@infoscience.otago.ac.nz).

B. Agent based workflow systems

The need for adaptive workflow systems and possible solutions to these problems have been discussed by researchers [4, 12]. A few researchers have provided solutions to some aspects of adaptability [2, 6]. By using agents as the building blocks of our framework, we have a flexible and open architecture where new process models can be incorporated dynamically which makes the system more adaptable [2,7,8].

C. Agent societies

Even though the computational paradigm defined as agent societies has been investigated for quite a while it is still a challenge for the research community. To implement a well-organized environment where agents can join and leave according to their goals has been proved to be a demanding task [19,20]. The main reasons are:

a) Security concerns during the communication process, development in a distributed and heterogeneous environment

b) Difficulty of implementing a common understanding for the agents about the different contexts that they are operating

c) Necessity of developing standards such as agent abstract architectures, Agent Communication Languages (ACLs) and conversation protocols

d) Difficulty of implementing autonomy in individual agents that join unknown environments.

Recent studies on the implementation of management mechanisms for agent interactions define concepts such as institutions [16] to define rules in an agent society and commitments [17] between agents to enforce obligations among actors during agent conversations. The main focus of those studies is the organization of communication among artificial agents. The enforcement of security on those interoperations is still not deeply approached.

D. Secure multi-agents

Lack of standards for multi-agents has given rise to the multitude of implementations. Foundation of Intelligent Physical Agents (FIPA) [23] security specifications are limited and there has been no implementation of the specification available till date. Some work has been done by researchers to secure agent interactions [13, 14, 15]. But, secure authentication mechanisms for agent-based societies have not been considered and this is relatively a new field of research.

The need for secure communication in an agent-based society arises when the policies of a society are transferred from one member of the society to the other and preventing a malicious agent from distributing a fake policy to other agents in the society.

III. PROPOSED AUTHENTICATION MECHANISMS

A workflow agent platform as shown in figure 1, is composed of many societies such as worker's society, resource broker's society etc and also individual agents such as process agent, monitoring agent and the controlling agent. These agents work collaboratively to enact a process. In order to have a secure exchange of messages between agents the messages should be authenticated. In this section we describe the authentication mechanism from the agent society point of view using the public key infrastructure (PKI)

A. Agent based workflow society

Workflow systems typically consist of several resources that perform various tasks. Depending upon the tasks that they perform, the resources form a society. The architecture of the agent based workflow system is given in figure 1.

The manager agent provides all functionality the workflow manager needs such as creation and deletion of tasks, roles and process definitions, instantiation of new process instances and creation of resource agents. The process agent executes a process instance. Each resource in the system has its own resource agent. Every resource in the system gets registered to one of the broker agents that allocate the resources to the process. The storage agent manages the persistent data that is needed. The monitor agent collects all the process specific data and sends them to the storage agent. The control agent continuously looks for anomalies to the criteria specified by the human manager and reports the violations to these criteria to the manager agent. The manager agent provides information to the human manager, which can be used for the feedback mechanism. More details about the agent based workflow system can be found in our previous works. [2,7,8]

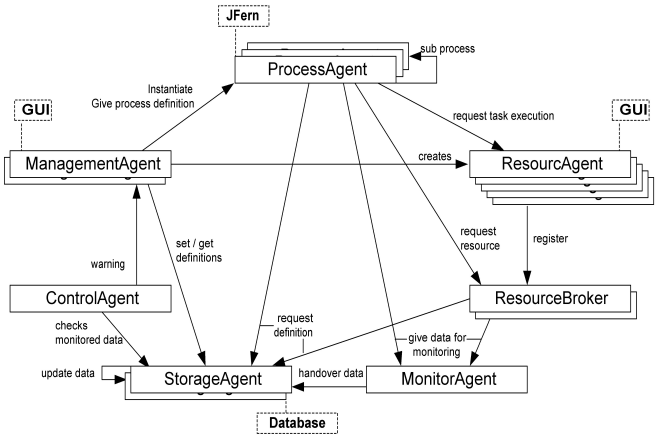

Fig. 1. Architecture of an agent based workflow management system.

In the workflow context, the resource agents can form a society and the resource agents can form societies depending upon the "roles" that they carry out.

Consider the software engineering development environment as an example for the formation of a society. A software firm consists of many levels of resources such as project managers, domain specialists, architects, programmers, testers etc. To simplify the scenario, we consider managers, programmers and testers. A society of managers is formed by the "role" played by the manager. There is an entry-level check for a manager to join this society. The manager agent has to meet certain the requirements in order to join the society. When the manager agent joins the society, it is expected to obey the rules laid down by the society. It is similar to the social obligations that a person must meet in the real world. Similarly the programmers and testers form their own respective societies. Figure 2 shows various societies in a software development environment. The bigger circles depict a society and the smaller circles within them depict the agents in the society. A platform consists of agents belonging to different societies as well as agents that do not belong to any of these societies.

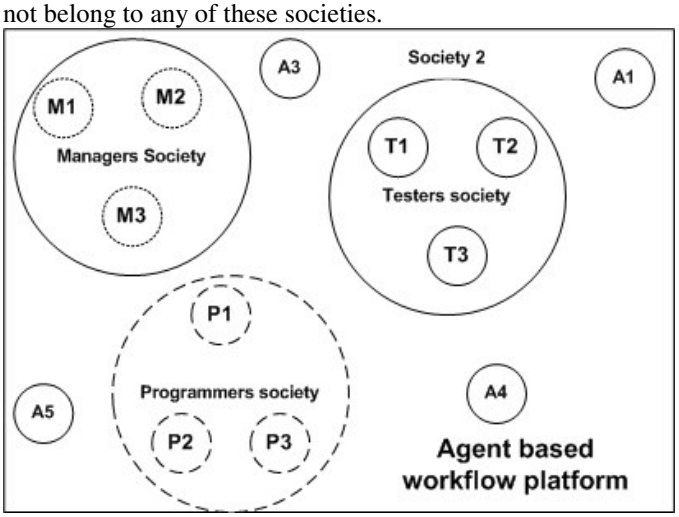

Fig. 2. Multi-agent based workflow society.

B. Authentication mechanism for agent interactions in an agent based system.

1) Creation of agents in the platform:

The agent platform P, is responsible for the creation of agents. It instantiates all the agents in the platform. Assume that it creates two agents A1 and A2. When the agent platform creates these agents, it also creates the public and private keys associated with each agent.

$$P \xrightarrow{i} A1$$

$$P \xrightarrow{i} A2 \text{ (i implies the instantiation)}$$

2) Availability of Keys:

The platform stores the public keys ($K_{A1(pb)}$, $K_{A2(pb)}$) in the Platform Specific Agent Certification Authority (PSACA). The PSACA is the central repository and also the authorizer of key certificates. A PSACA can be modeled as an agent or can be implemented as an extension to the core platform. In our system all Agent Certification Authorities (ACA) are modeled as agents.

Each agent possesses its own private key ($K_{A1(pr)}$, $K_{A2(pr)}$).

$P \xrightarrow{s}$ PSACA: [$K_{A1(pb)}$, $K_{A2(pb)}$] (s implies the storage)

When A1 sends a message to A2 it obtains the public key of A2 from the PSACA and encrypts the message with it's private key and then by the public key of A2 ensuring that only A2 will be able to read messages from A1. It also adds a unique session key to the message, with which A2 has to reply with. The session key is used to prevent replay attackers from reading the responses. This is depicted as steps 1, 2 and 3 in figure 3.

$A1 \xrightarrow{m}$ A2: [$K_{A1(pr)}$, $K_{A2(pb)}$,] , m the message

Similarly when A2 sends a reply message to A1, it encrypts the message with the session key sent by A1, then with its private key and then by A1's public key. This ensures that the message to A1 is indeed from A2. This is depicted as steps 4,5 and 6 in figure 3.

$A2 \xrightarrow{m}$ A1: [K_s, $K_{A2(pr)}$, $K_{A1(pb)}$], m the message

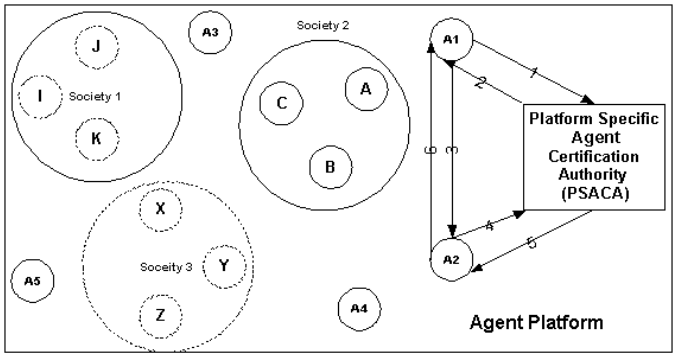

Fig. 3. Authentication mechanism between two agents in a agent society.

In the workflow context, resource broker contacts a resource agent to perform a particular task. The resource may or may not agree to perform a task. This information of agreement or disagreement can be forged, repudiated by a malicious agent. So there needs to be a mechanism that deals with the secure interaction of agents across societies.

Agent societies are formed when agents with similar interests come together to work towards a common goal. This implies that these agents in a society should adhere to the rules laid down by the society. These rules or protocols should be transmitted in a secure way. There might be a malicious agent, which might want to break the rules of the society or may want to send a false protocol to the other agents in the society.

This gives rise to a security protocol for interactions among the agent societies. One should address the following questions so that appropriate authentication protocols can be arrived at.

a) How does a new agent join a society?
b) How does another agent in the workflow context contact an agent in a society?
c) How does one agent from one platform contact an agent in another platform?
d) What happens when an agent leaves the society?
e) What happens when an agent drops out of the society?

The authentication mechanism for the above mentioned scenarios are described below.

C. Scenario I: Creation of secure agent societies

To elaborate the concept of creating secure agent societies, let us take an example of the workflow scenario. In a workflow system, there are typically many resource agents that might be performing certain tasks. These resource agents may form various societies among themselves such as worker society, manager society etc. Typically these societies can be formed depending upon the 'role' performed by the agent in the workflow society. 'Worker', 'Accountant','Manager' can be the roles performed by various agents.

When a 'worker' society is to be formed, a Society Specific Agent Certification Authority (SSACA) is created for each society. For all the agents that join the society a public-private key combination specific to that society is created. The public key of all agents in the society is stored in the SSACA. The SSACA is registered and authorized by the PSACA. This forms a hierarchy of certification authorities within an agent-based society.

Figure 4 shows a hierarchy of trust relationship between different certification authorities in two different platforms. PSACA's use the same inter agent communication as other agents and one of the PSACA's acts as the trustworthy source that supplies the authentication keys for the other PSACAs. Alternatively, we can make use of a central, higher-level Agent Communication Authority (ACA) that can manage the keys of all PSACAs.

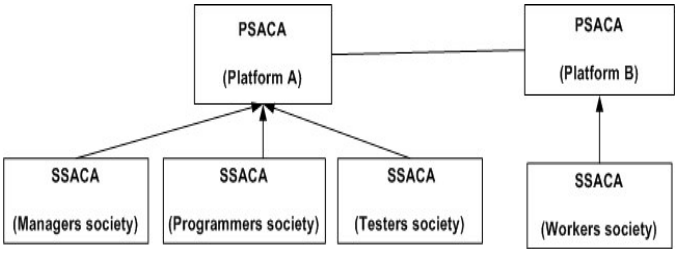

Fig. 4. Trust relationships between various certification authorities in a platform.

There are two modes of sending messages in the society. If the agent knows that the information to be sent is not society specific it encodes the message by the mechanism described in section B. If the message sent by an agent in a society is pertinent to the society such as sending a 'changed protocol' or accessing information such as credit card details, or accessing sensitive databases or accessing the internal information of a company, then the interaction should be secured using the authentication mechanisms.

Let us assume that agent A in a society wants to send a secure message to agent B in the same society. It first encrypts the message with its private key ($K_{Apr(soc)}$)in the society, the by the

public key of the receiver in the society ($K_{Bpb(soc)}$). The public key is available with the SSACA.

This mechanism ensures that the message is indeed sent by an agent in the society. When an agent possesses a key associated with a society, it is implicit that the agent belongs to the specified platform (P) as it had to be an agent in the platform for obtaining a key associated with the society.

$$A \xrightarrow{m} B: [K_{Apr(soc)}, K_{Bpb(soc)}]$$

Similarly when B replies by sending a message to A, it encrypts the message with its private key for the society, with B's public key for the society. This ensures that the message to A is indeed from B. This is depicted as steps 4,5 and 6 in figure 5.

$$B \xrightarrow{m} A: [K_s, K_{Bpr(soc)}, K_{Apb(soc)}]$$

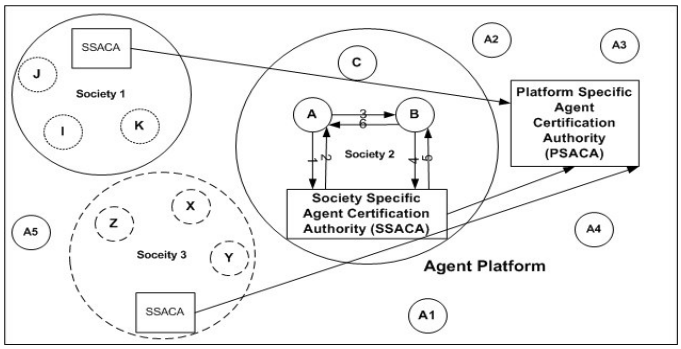

Fig. 5. Authentication mechanism between two agents in a society.

D. Scenario II : An agent joining the society

Each society has a designated moderator agent. The role of the moderator agent is to authorize agents in a society. Any new agent, which wants to join a society, contacts the appropriate SSACA. The details of the societies are available with the PSACA. This is shown in steps 1 and 2 of figure 6.

The new agent A sends a message to the SSACA requesting to join the society. SSACA provides the reference to the moderator of the society and the new agent contacts the moderator of the society (steps 3 to 5 of figure 6)

Step 6: The moderator (M) of the society sends the protocol/rules for the society to the new agent A.

$$M \xrightarrow{m} A\ [K_{M(pr)}, K_{A(pb)}]$$

The new agent either agrees or does not agree to the protocol

$$A \xrightarrow{m} M: [K_s, K_{A(pr)}, K_{M(pb)}]$$

If it agrees, then the moderator agent generates public-private keys for the new agent and stores the public key with the SSACA and the private key is sent to the new agent. The new agent verifies the validity of the new key by sending a message to the moderator agent. This is shown in steps 1-6 of figure 6.

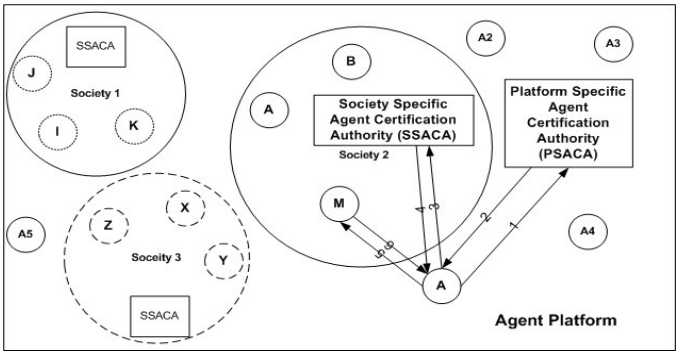

Fig. 6. Authentication mechanism for a new agent joining the society

E. Scenario III: An agent from one society communicating with an agent in another society:

In the context of the workflow, a resource broker agent (A) is responsible for finding out the best possible resource for a particular task. The resource broker finds the best possible resource from the resource log (based upon the history data in a persistent storage) and contacts the resource (B) in the resource broker society. When a non-societal agent needs to interact with an agent that belongs to a particular society or when an agent from Society 1 wishes to communicate with an agent in Society 2 the mechanism described in figure 3 of section B is used.

F. Scenario IV: An agent from one platform contacting an agent in another platform

The agents are built using the agent framework called Otago Agent Platform (OPAL)[21]. The scenario described is the communication of two agents between two instances of OPAL platforms. One of the platforms could be located in New Zealand and the other in Germany. In the scenarios described, we have not considered the secure communication between agents created in different agent architectures such as Java Agent Development Framework (JADE) [22]. Steps 1 to 5 of Figure 7 describe the agent interaction for this scenario.

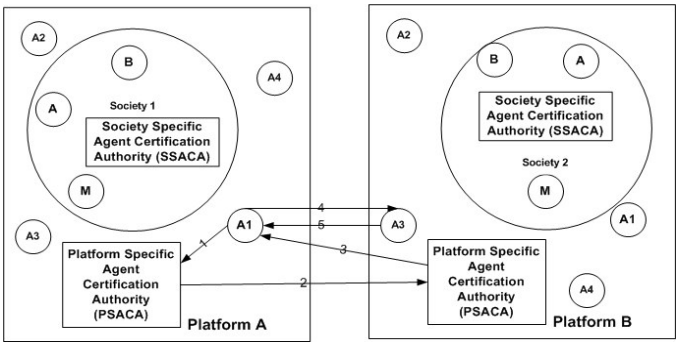

Fig. 7. Scenario describing secure agent interactions between two platforms

When an agent A1 of platform A wants to send a message to agent A3 in platform B, it contacts the PSACA to find the public key of the A3. The PSACA of platform A contacts the PSACA of platform B to find the public key of A3. Agent A1 obtains the public key of A3 from PSACA of platform A. Then A1 uses the authentication mechanism as described in section B to send and receive messages from A3.

Similarly when an agent from society 1 of platform A, wants to contact an agent from society 2 of platform B the above described authentication mechanism is used.

G. Scenario V: An agent leaving the society

When the agent A leaves the society willingly, it sends a message to the society moderator that it no longer wants to be a part of the society. The moderator then contacts the SSACA to revoke the certificate for that agent. Once the certificate for the agent is revoked, the keys are no longer valid and every agent checks for the validity of the certificate of the sender for authentication purposes. Figure 8 shows the sequence diagram of the scenario.

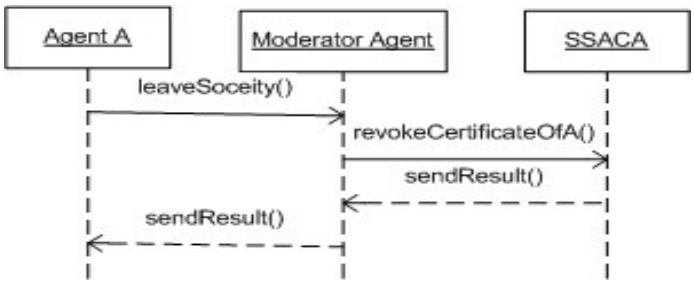

Fig. 8. Scenario describing an agent leaving the society

H. Scenario VI: An Agent dropping out of the society

Figure 9 shows the scenario of an agent dropping out from the society. When an agent fails to communicate with another agent in the society for predetermined number of replays then the sender agent, sends a message to the moderator to find if the receiver agent is still active. The moderator sends a message to the receiver agent and if that too finds that there is no response from the agent, it is considered to have dropped out and the message is sent to revoke the agent's certificate to the SSACA.

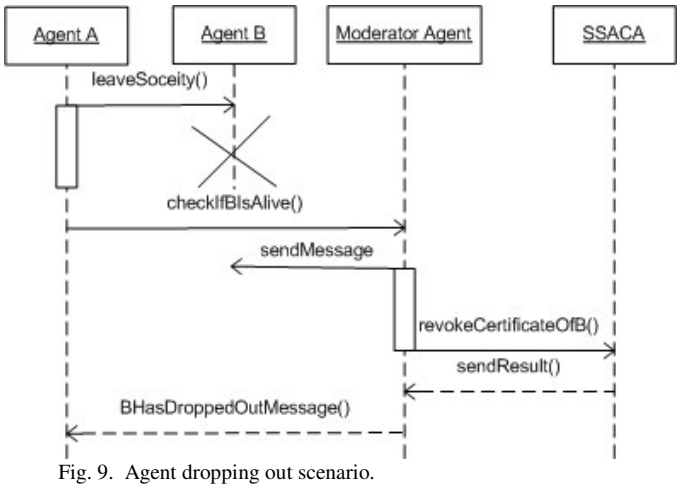

Fig. 9. Agent dropping out scenario.

IV. Conclusion

Agent communication has to be authenticated so that information can be transferred in a secure way. Agent societies need appropriate security mechanisms when sensitive data is transferred. We have described mechanisms, which describe how agent interactions can be authenticated in various scenarios in an agent based workflow society. It may be noted that not all agent interactions need secure authentication mechanisms. Only those interactions that deal with the transfer of sensitive information need to be authenticated.

In future, we would look into the secure communication across agents implemented using different agent architectures such as JADE. This can be achieved by sending the security protocols that need to be adopted to the agents implemented using other frameworks through messages.

We are also working to minimize the overheads caused by the encryption and decryption by employing mechanisms such as decision making systems in the agent framework which help in the deciding whether a message requires a particular level of encryption is needed or not.

References

[1] Bradshaw, J., *An Introduction to Software Agents*, in *Software Agents*, J. Bradshaw, Editor. 1997, MIT Press: Cambridge. p. 3-46.

[2] Ehrler, L., Fleurke, M., Purvis, M. A. and Savarimuthu, B.T.R., "Agent-Based Workflow Management Systems(Wfmss) : JBees- A Distributed and Adaptive WFMS with Monitoring and Controlling Capabilities", to be published in a special issue of the *Journal of Information Systems and e-Business on Agent-Based Information* (2005).

[3] Shoham, Y., "An Overview of Agent-Oriented Programming", in *Proc. Software Agents*, J. Bradshaw, Editor. 1997, MIT Press: Cambridge. p. 271-290

[4] Stormer, H. *AWA* – "A flexible Agent-Workflow System". in *Workshop on Agent-Based Approaches to B2B at the Fifth International Conference on Autonomous Agents (AGENTS 2001)*. 2001. Montreal, Canada.

[5] Wang, M. and Wang, H. "Intelligent Agent Supported Flexible Workflow Monitoring System" in *Advanced Information Systems Engineering: 14th International Conference, CAiSE 2002*. 2002. Toronto, Canada: Springer Verlag GmbH

[6] Paul Buhler, José M. Vidal, and Harko Verhagen. Adaptive workflow = Web Services + agents. In *Proceedings of the International Conference on Web Services*, pages 131-137. CSREA Press, 2003.

[7] Fleurke, M and Ehrler, L, Purvis, M. A. (2003). "*JBees - An Adaptive and Distributed Agent-based Workflow System*", in Proceedings of the International Workshop on Collaboration Agents: Autonomous Agents for Collaborative Environments (COLA 2003), Halifax, Canada, October 2003. IEEE/WIC Press. Ghorbani, A. And Marsh, S., Ed.

[8] Savarimuthu, B.T.R., Purvis, M. A. and Fleurke, M. (2004), "Monitoring and Controlling of a Multi-agent Based Workflow System", *Proceedings of the Australasian Workshop on Data Mining and Web Intelligence (DMWI2004)*, Conferences in Research and Practice in Information Technology, Vol. 32, Australian Computer Society, Bedford Park, Australia (2004) 127-132.

[9] Paul Buhler and José M. Vidal. Integrating agent services into BPEL4WS defined workflows. In *Proceedings of the Fourth International Workshop on Web-Oriented Software Technologies*, 2004.

[10] Paul Buhler and José M. Vidal. Enacting BPEL4WS specified workflows with multiagent systems. In *Proceedings of the Workshop on Web Services and Agent-Based Engineering*, 2004.

[11] Martin Fleurke. JBees, an adaptive workflow management system - an approach based on petri nets and agents. Master's thesis, Department of Computer Science, University of Twente, P.O. Box 217, 7500 AE Enschede, The Netherlands, 2004.

[12] N.C. Narenda, Adaptive Workflow Management . An integrated Approach and System Architecture, Proceedings of the 2000 ACM symposium on Applied computing, March 2000.

[13] Min Zhang, Ahmed Karmouch, "Adding Security Features to FIPA Agent Platforms", 2001, http://www2.elec.qmul.ac.uk/~stefan/fipa-security/rfi-responses/ Karmouch-FIPA-Security-Journal.pdf

[14] Wen, Wu and Mizoguchi, Fumio, An Authorization-Based Trust Model for Multiagent Systems, Applied Artificial Intelligence, 14:909-925, 2000

[15] Yuh-Jong Hu, "Some thoughts on agent trust and delegation", *Proceedings of the fifth international conference on Autonomous agents*, p.489-496, May 2001, Montreal, Quebec, Canada

[16] Mario Verdicchio, Marco Colombetti: A Logical Model of Social Commitment for Agent Communication. Workshop on Agent Communication Languages 2003: 128-145

[17] Marco Colombetti, Nicoletta Fornara and Mario Verdicchio.(2002) The Role of Institutions in Multiagent Systems *Ottavo Convegno Associazione Italiana per l'Intelligenza Artificiale AI*IA*, Siena, Italy

[18] Nissen, M.E. "Supply Chain Process and Agent Design for E-Commerce" in *33rd Hawaii International Conference on System Sciences*. 2000. Maui, HI, USA

[19] De Oliveira M., Purvis M., Cranefield S., Nowostawski M. (2004). Institutions and Commitments in Open Multi-Agent Systems. To be published in the proceedings of the Intelligent Agent Technology (IEEE/WIC/ACM IAT-2004). Beijing – China.

[20] De Oliveira M., Purvis M., Cranefield S., Nowostawski M. (2004). A Distributed Model for Institutions in Open Multi-Agent Systems. To be published in the proceedings of the Workshop on Multi-Agent Systems, Ontologies and Conflict Resolution (MASOCR at KES-2004). Wellington – New Zealand.

[21] Purvis, M., Cranefield, S., Nowostawski, M., and Carter, D., "Opal: A Multi-Level Infrastructure for Agent-Oriented Software Development", *Information Science Discussion Paper Series*, Number 2002/01, ISSN 1172-6024, University of Otago, Dunedin, New Zealand (2002).

[22] F.Bellifemine et.al, "JADE – A FIPA-compliant agent framework" Proceedings of PAAM'99, London, April 1999, p.97-108.

[23] Joint Submission, FIPA Specification, http://www.fipa.org, 1997