Agents Ownership Setting by User Fingerprints

S. Vitabile, G. Pilato
Istituto di CAlcolo e Reti ad alte prestazioni
Italian National Research Council
Viale delle Scienze, 90128, Palermo, Italy
Email: {s.vitabile, g.pilato}@icar.cnr.it

V. Conti, C. Ferrara, F. Sorbello
Dipartimento di Ingegneria Informatica
University of Palermo
Viale delle Scienze, 90128, Palermo, Italy
Email: {conti, ferrara}@csai.unipa.it; {sorbello}@unipa.it

***Abstract*— Agent ownership is a difficult task that touches legal domain. Agent ownership implies that a specific person or organization (the owner) is responsible for the agent's actions. Security requirements in the agent ownership setting process are the identification of the owner and the protection of the identity information carried by an agent. The first issue deals with users authentication process while the second issue deals with information encryption techniques, certificates etc. In this paper a new user authentication system requiring owner fingerprint in addition to the standard username/password pair is addressed. Fingerprint based identification systems exploits two basic premises: persistence, i.e. the basic characteristic of fingerprints do not change with time, and individuality, i.e. a fingerprint is unique to an individual. The implementation of the enhanced authentication system in the JADE-S platform, a FIPA compliant platform formed by the combination of the standard version of JADE with the JADE security plug-in, is also outlined.**

I. Introduction

Agent ownership is a difficult task that touches legal domain. An agent is an autonomous computational entity that executes tasks and actions to benefit a human entity. Agent ownership implies that a specific person or organization (the owner) is responsible for the agent's actions. Yip and Cunningham in [22] exert two different type of agent's ownership: in the first one, an agent can be treated as a piece of software and, as such, it is an intellectual property which belongs to its owner and is governed by the law on intellectual property; in the second one, the user of an active software agent owns the agent's services and actions.

A common application for agent-based technologies is to provide services for their owners. Human business agents models are being re-applied to the computer network and software agents are replacing human agents. In a framework supporting agents ownership, agents can be legal entities employed to bring his own principal into contractual relations with third parties [22]. The authentication process establishes the identity of each owner and, consequently, of each agent, while a policy, based on the previous identity, can determine the access level of an agent in the system, the permission to access to certain resources or perform certain tasks.

Many researchers treat ownership as a passive component of trust. For example, Rahman and Hailes [25] propose a distributed trust model based on the assumption that an agent will be able to keep a history of interaction with other agents and hence assign different level of trust to the peers. However, it is not always possible to build up the necessary record to decide the level of trust toward an agent, especially when there is no direct relationship between the real world legal entity and the software agent representing it. In other research [26], [27], [28] it is argued that in the design of multi-agent systems, in order to support rich collaborative behavior it is essential to provide individual agents with various forms of social awareness. Mamdani and Pitt have suggested that an agent should be able to express the fact that an agent is owned by some human entity and be able to reason about the consequences and responsibilities of the delegation. As stated in [22], ownership and trust can be viewed as a mutual relationship, where ownership helps to sustain trust model by providing a legal foothold.

Security requirements in the agent ownership setting process are the identification of the owner and the protection of the identity information carried by an agent. The first issue deals with the users authentication process while the second issue deals with information encryption techniques, certificates etc. In this paper a robust authentication system based on user fingerprint is outlined.

High security authentication system design still remains an open problem. Complex password are easy to forget while simple password are easily guessed by unauthorized persons. An unauthorized person, stealing a trusted username/password pair, can access into a system, runs his/her malicious agents, purchases or sells something; the person owns the trusted username/password pair will be the legal responsible of these actions.

Fingerprints have been widely used in personal authentication tasks, due to their uniqueness and immutability [13], [14], [15], [16]. Since fingerprint features do not change, many good human identification systems have been based on fingerprint. A fingerprint is composed by ridges and valley, having a curvature value. Ridges may have bifurcations or may be interrupted. These features, called minutiae, are used to identify a person since they change from person to person. Fingerprints can be classified using the existence and the spatial position of two singular points, called Core and Delta [23]. NIST has proposed five standard classes: Right Loop, Left Loop, Whorl, Arch and Tented Arch.

Giving more details, in the paper a two security level authentication system is outlined. Agent ownership is established after an enhanced authentication process requiring username, password and user fingerprint. Initially, the system verifies the username/password pair supplied by the owner and then, only if username/password verification is positive, it requests owner fingerprint. Fingerprint, acquired through a sensor, is compared with the related item stored in a database. A public/private key pair is used to sign each fingerprint stored in the database, so the system, is able to control the authenticity of the signed stored fingerprint before fingerprint matching. Fingerprint matching is performed via a set of algorithms proposed by the authors, exploiting both the adaptive Local Energy Threshold (LET) and a new matching operator, based on the Tanimoto distance [1], [2].

As framework we have chosen the JADE-S platform, a FIPA (Foundation of Physical Intelligent Agents) compliant multi-agent platform supporting the user authentication process and the agents ownership issue. JADE-S, the JADE Secure Agent Platform [19], is formed by the combination of the standard version of JADE with the new JADE security plug-in [19], [6]. Obviously, the enhanced authentication system for ownership setting can be implemented and used with other multi/agent system supporting the above features.

The paper is organized as follows: in section 2, fingerprint verification algorithms proposed by the authors are described; in section 3 the proposed authentication system is described, in section 4 the implementation of the enhanced authentication system in the Jade-S platform is outlined. Finally, in section 5 the conclusions are reported.

II. The Fingerprint Verification Task

Fingerprints are made by a set of lines having endpoints and bifurcations (see Figure 1-a). Fingerprints are used in two different kind of tasks for establishing person identity: the verification task and the identification task. In the identification task, the system has only a total or partial portion of the fingerprint and it must establish the identity of the person comparing the processed fingerprint with each fingerprint images, stored in a database. In the verification task, the system has to recognize an acquired fingerprint image using personal data information to select a fingerprint database item. A fingerprint matching is successively performed between the acquired fingerprint image and the selected database item.

A verification system able to process the NIST 4 fingerprint database with good results was proposed by the authors [1], [2]. Fingerprint verification was performed in three main phases: the preprocessing phase, the minutiae extraction phase, and the matching phase.

The pre-processing phase aims to obtain a binary image containing a set of ridges whose thickness is only one pixel. The whole phase is based on the automatic computation of the Local Energy Threshold (LET) for fingerprint image binarization. The LET is the average energy of the eight neighbors of each processed pixel. If pi is the brightness of the central pixel of a 3x3 mask, the new value for pi is calculated as follows:

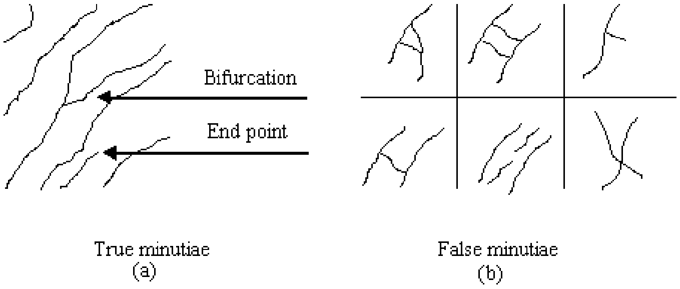


Fig. 1. In (a) are shown true fingerprint minutiae, such as endpoint and bifurcation. In (b) are shown some false recurrent features, such as triangles, structure ladders, spurs, bridges, interrupted ridges and forks.

$$p_i = \begin{cases} 255 \quad \text{if } \frac{1}{8} \cdot \sum_{k=0,k\neq i}^{8} (p_k) \geq LET \\ 0 \qquad otherwise \end{cases} \tag{1}$$

The minutiae extraction phase aims to obtain a *n*-dimensional vector, where *n* is the number of the extracted minutiae and the vector components are the spatial coordinates and ridge direction of extracted minutiae. The considered features are only endpoint and bifurcation, called true minutiae, while other elements as triangles, structure ladders, spurs, bridges, interrupted ridges and forks are considered false minutiae and, consequently, discarded. Figure 1 shows both the real and the false minutiae.

The matching phase aims to verify if a processed fingerprint pairs belong to the same person. The matching rate is given by an operator that extends the concept of the Tanimoto distance [18].

Let V the set of records $r(X,Y,\Theta)$ representing the current image, W_i the set of records $r(X,Y,\Theta)$ of the i-th database image:

$$F\left(W_j\right) \equiv V \bowtie W_j \tag{2}$$

where $\bowtie$ is a new intersection operator defined as follows:

$$k_i \in V \bowtie W_j \Leftrightarrow \begin{cases} |X - X_i| \leq T_x \\ \\ |Y - Y_i| \leq T_y \\ \\ |\Theta - \Theta_i| \leq T_\theta \end{cases} \tag{3}$$

with T_x, T_y and T_θ noisy (translation and/or rotation) immunity thresholds and X, Y, and Θ the spatial coordinates and the ridge direction of the processed minutia, respectively.

With more details, the main phases of fingerprint verification process are:

1) the preprocessing phase
   a) directional image extraction;
   b) fingerprint image segmentation;
   c) fingerprint image binarization, based on LET;
   d) fingerprint image median filtering;
   e) fingerprint image thinning;
2) the minutiae extraction phase

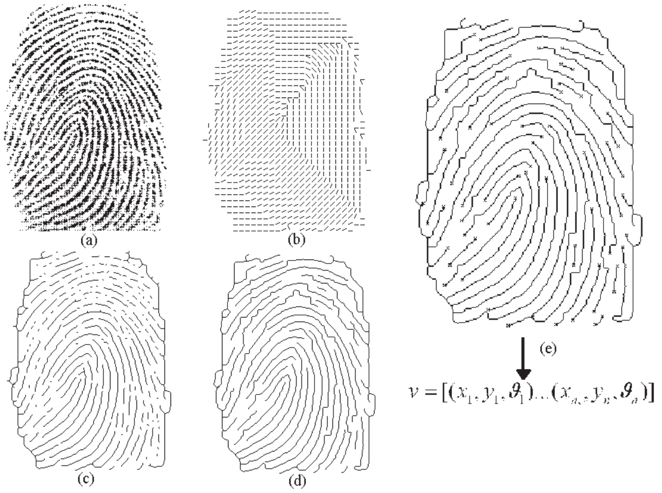


Fig. 2. The obtained results after the preprocessing phase and the minutiae extraction phase. In (a) is shown the original image, in (b) is shown the extracted directional image, in (c) is shown the final result of preprocessing phase. In (d) and (e) are shown the extracted minutiae and the related vector v. The vector will contain two spatial coordinates and the direction of each extracted minutia.

   a) false minutiae erasing process;
   b) true minutiae extraction process;
   c) fingerprint image codifying process giving an *n*-dimensional vector;

3) the matching phase.

In Figure 2 are depicted the obtained results after the preprocessing phase and the minutiae extraction phase.

III. The Proposed Authentication System

Multi-agent systems (MAS) authentication process is only based on username/password pair. Due to their uniqueness and immutability, fingerprints can be used in personal authentication systems [13], [14], [15], [16]. A Fingerprint Verification System (FVS) can be added to existing MAS authentication systems to enhance the access security level and, consequently, to improve the agent ownership setting process security.

Users authentication task can be performed using two files: the password file, containing the username/password pair and the fingerprint file, containing the username/fingerprint features pair.

We have developed a new version of the algorithms described in the previous section, in order to process fingerprint images acquired through a sensor.

Authentication operations require two phases: the first one deals with the system setup in which users fingerprint are acquired, signed, via a public/private key pair, and then stored in the fingerprint file; the second phase deals with on-line user fingerprint sensor acquisition and with fingerprint pair verification process.

A. The Fingerprint Verification System

Each user is identified by the triplet (*username*, *password*, *fingerprint*) set by the administrator in the registration phase. An efficient authentication system will verify first the username/password pair supplied by the user, while his/her fingerprint will be verified only when the first verification process result is positive. The sequential authentication phases are illustrated in Figure 3.

A public key can be used to verify the minutiae vector integrity before the matching process. Also in this case, the user authentication process will be interrupted, if the vector integrity check is negative. As pointed before, the matching process is performed using the algorithms described in the previous section. The main phases of the matching process are shown in Figure 4.

IV. Case Studied

As framework we have chosen the JADE-S platform, a FIPA (Foundation of Physical Intelligent Agents) compliant multi-agent platform supporting the user authentication process and the agents ownership issue. JADE-S is formed by the combination of the standard version of JADE with the new JADE security plug-in [6], [19]. The main features of JADE-S platform are the following:

- Authentication: an user must be authenticated, providing a username and password, to be able to own or perform actions on a component of the platform. Only authenticated users can own AMS, DF, containers and other agents;
- Authorization: JADE-S uses the concept of Principal as an abstraction for a user account, an agent or a container. A Principal must be authorized by the Java Security Manager. The security manager allows or denies the action according to the JADE platform's policy;
- Permissions and Policies: a permission is an object that describes the possibility of performing an action on a certain resource such as pieces of code, but also who executes that code. A policy specifies which permissions are available for various principals;
- Certificates and Certification Authority: the Certification Authority (CA) is the entity that signs all the certificates for the whole platform, using a public/private key pair. When the CA signs a document, it first makes a digest of it, which is a shorter non-reversible version of the document, a kind of a checksum. Then the digest is encrypted with the private key;
- Delegation: this mechanism allows the ”lending” of permissions to an agent. Besides the identity certificate, an agent can also own other certificates given to it by other agents;
- Secure Communication: communication between agents on different containers/hosts, are performed using the Secure Socket Layer (SSL) protocol. This enables a solid protection against malicious attempts of packet sniffing.

The FVS, described in the previous section, has been implemented in the JADE-S platform. We have modified the platform start-up process introducing two security level: a first level in which username/password are requested and a

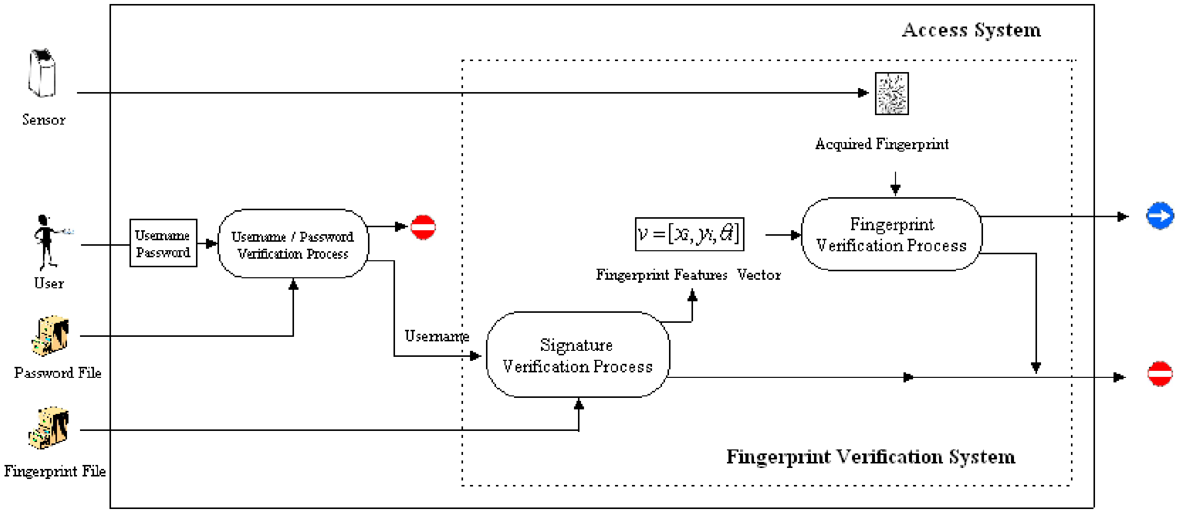

Access System
Sensor
Acquired Fingerprint
User
Username Password
Username / Password Verification Process
$v=[x_i,y_i,\theta_i]$
Fingerprint Features Vector
Fingerprint Verification Process
Password File
Username
Signature Verification Process
Fingerprint File
Fingerprint Verification System

Fig. 3. The proposed Fingerprint Verification System.

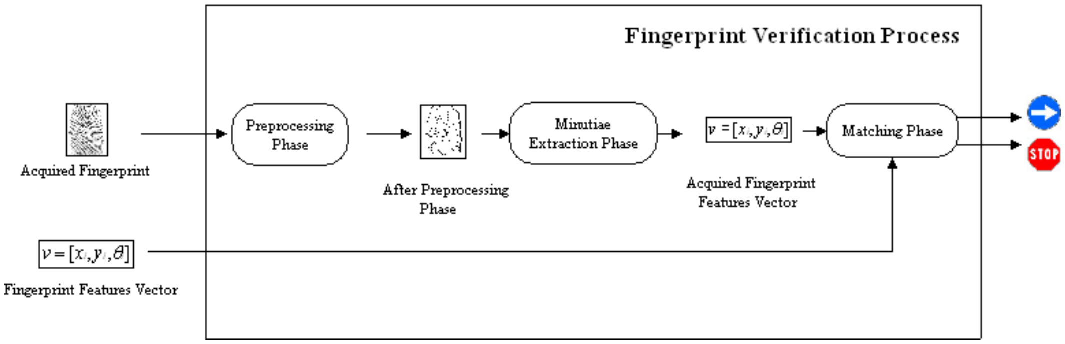

Fingerprint Verification Process
Acquired Fingerprint
Preprocessing Phase
After Preprocessing Phase
Minutiae Extraction Phase
$v=[x_i,y_i,\theta_i]$
Acquired Fingerprint Features Vector
Matching Phase
STOP
$v=[x_i,y_i,\theta_i]$
Fingerprint Features Vector

Fig. 4. The sequential phases of the Fingerprint Verification Process.

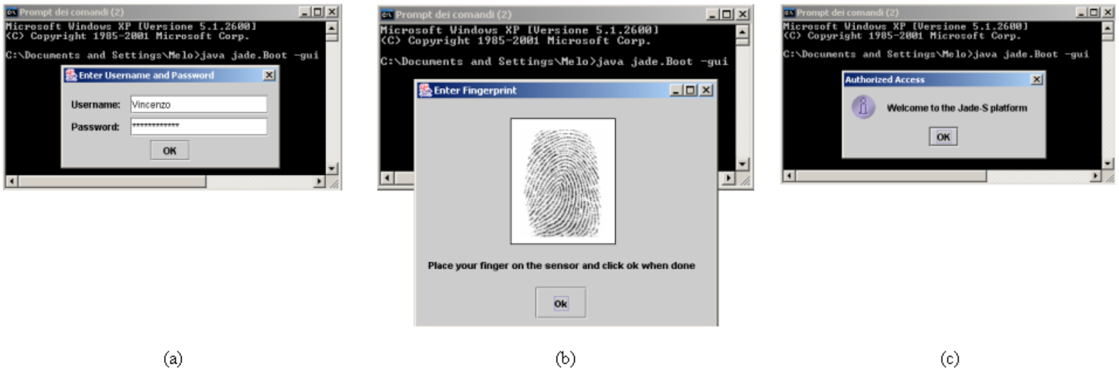
second level in which user fingerprint is requested. In JADE-S the user who starts-up the platform owns the AMS, the DF and the main container [6]. Using the modified user authentication system, JADE-S platform can run only when the whole authentication procedure is correctly performed: the main container will be opened and the RMA, DF and AMS will be activated, owned by the authenticated user. Specifically, the platform access procedure is the following:

1) when an user is going to run the JADE-S platform, the system requires username/password pair and verify it (first security level in Figure 5-a);
2) if the username/password verification result is positive and the related signed fingerprint has not been modified and replaced, the user fingerprint will be required and acquired;
3) the acquired fingerprint will be compared with the stored ones (second security level in Figure 5-b) using the previous described algorithms;
4) if the fingerprint matching process gives negative result, the access will be denied, otherwise the Jade-S platform will be launched (see Figure 5-c).

The JADE-S platform integrated with the proposed authentication system was installed on an Intel Pentium III 1,13 GHz processor, with 384 MB SDRAM, under Windows XP operating system. The SecuGen Eyed Hamster sensor [9] has been used to acquire 300x260 fingerprint images. The whole user authentication process requires 2.59 s. The most of the processing time is required by the fingerprint minutiae extraction process.

V. Conclusions

Agent ownership implies that a specific person or organization (the owner) is responsible for the agent's actions. Security requirements in the agent ownership setting process are the identification of the owner and the protection of the identity information carried by an agent. In this paper a fingerprint based identification system has been proposed, so agent ownership is fixed by the triplet (*username*, *password*, *fingerprint*). The Fingerprint Verification System has been successfully implemented and tested in the JADE-S platform: a first security level deals with user username/password pair, while the second, added, security level deals with owner fingerprint. Owner fingerprint can represent a link between the human world and the agent world, so that information like ownership can be used for legal aspect as well as for authorization and permission issues in multi-agent platform.



```
Microsoft Windows XP [Versione 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\Documents and Settings\Melo>java jade.Boot -gui
```

Username: Vincenzo

Password: *************

Place your finger on the sensor and click ok when done

Welcome to the Jade-S platform

(a) (b) (c)

Fig. 5. In the figure is shown the whole access platform procedure. In (a) the first security level with username/password requesting is shown; in (b) the second security level with the user fingerprint requesting is shown; finally in (c) a welcome access window is shown.

References

[1] V. Conti, G. Pilato, S. Vitabile and F. Sorbello. A Robust System for Fingerprints Identification. In *Proc. Knowledge-Based Intelligent Information Engineeering System and Allied Technologies compatibility*, pages 1162–1166, 2002.

[2] V. Conti, G. Pilato, S. Vitabile and F. Sorbello. Verification of ink-on.paper Fingerprints by Using Imange Processing Techniques and a New Matching Operator. In *Proc. AI*IA*, 2002.

[3] S. Vitabile, V. Conti, G. Pilato and F. Sorbello. A Fingerprint Based Authentication System for the Jade-S Platform. In *Agentcities ID3*, 2003.

[4] F. Bellifemmine, A. Poggi and G. Rimassa. JADE - A FIPA compliant agent framework. *Proc. of 4th International Conference and Exhibition on the Practical Application of Intelligent Agents and Multi-Agents*, 1999.

[5] A. Poggi, G. Rimassa and M. Tomaiuolo. Multi-User and Security Support for Multi-Agent Systems. In *Proc. AI*IA - TABOO*, 2001.

[6] G. Vitaglione. Jade Tutorial Security Administrator Guide - September, 2002

[7] P. Novak, M. Rollo, J. Hodik, T. Vlcekand and M. Pechoucek. X-Security Architecture in Agentcities.

[8] H.Chi Wong and K. Sycara. Adding Security and Trust to Multi-Agent System. *Proc. of Autonomous Agents*, 1979.

[9] W.M. Farmer, J.D. Guttman and Swarp. Security for Mobile Agent: Authentication and State Appraisal. In *Proc. of the European Symposium on Research in Computer Security (ESORICS)*, volume 1146 in LNCS, pages 118–130, 1996.

[10] T.D. Peddireddi and J.M. Vidal. Multiagent Network Security System using FIPA-OS. In *IEEE Proc. Southeast Conference*, 2002.

[11] H. Peine. Security Concepts and Implementation in the Ara Mobile gent System. In *7th IEEE Workshop on Enabling Technologies: Infrastructure for Collaborative Enterprises*, 1998.

[12] V. Roth and M.J. Sohi. *Concepts and Architecture of a Security-centric Mobile Agent Server*. Proc. 5th International Symposium on Autonomous Decentralized System (ISADS), 2001.

[13] A. Jain, L. Hong and R. Bolle. On-Line Fingerprint Verification. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, volume 19, number 4, 1997.

[14] A. Jain, L. Hong, S. Pankanti and R. Bolle.. An Identity-Authentication System Using Fingerprints. *Proc. of the IEEE*, volume 85, number 9, 1997.

[15] Z.M. Kovacs-Vajna. A Fingerprint Verification System Based on Triangular Matching and Dynamic Time Warping. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, volume 22, number 11, 2000.

[16] X. Tan, B. Bhanu. Robust Fingerprint Identification. In *IEEE International Conference on Image Processing*, 2002.

[17] M. Ballan and F. Ayhan Sakarya. A Fingerprint Classification Technique Using Directional Images. *IEEE*, 1998.

[18] K.R. Sloan Jr and S.L. Tanimoto. Progressive Refinement of Raster Images. *IEEE Transaction on Computers*, volume 28, number 11, pages 871–874, 1979.

[19] http://sharon.cselt.it/projects/jade/

[20] http://www.secugen.com/home.htm

[21] Java Security http://java.sun.com/security/

[22] A. Yip and J. Cunningam. Some Issue on Agent Ownership. *LEA Workshop on the Law of Electronic Agents*, 2002.

[23] http://www.nist.gov/srd/nistsd4.htm.

[24] Sycara. Multi-agent Infrastructure. Agent discovery, Middle Agents for Web Services and Interoperation. *Lecture Notes in Artificial Intelligence: Multi-Agent System and Applications*, pages 17–49, 2001.

[25] A. Abd ul-Rahman and S. Hailes. A Distributed Trust Model, New Security Paradigms. http://citeseer.nj.nec.com/347518.html, 1997.

[26] C. Castelfranchi and R. Falcone. Socio-Cognitive Theory of Trust. ALFEBIITE Deliverable report D1, 2001.

[27] A. Mamdani and J. Pitt. Responsible Behavior for Networked Agents A Distributed Computing Perspective. *IEEE Internet Computing* Vol. 4, No. 5, Sept./Oct. 2000, pp. 27-31.

[28] F. Dignum, D. Morley, E.A. Sonenberg and L. Cavedon Toward socially sophisticated BDI agents. *Proceedings of the ICMAS 2000*, pages 111-118, 2000.