A Trust Model with Dynamic Decision Making for Ubiquitous Environments

Weiwei Yuan, Donghai Guan, Le Xuan Hung, Youngkoo Lee*, Sungyoung Lee
Department of Computer Engineering, Kyung Hee University, Korea
{weiwei, donghai, lxhung ,sylee}@oslab.khu.ac.kr, yklee@khu.ac.kr

Abstract - This paper presents a novel role-based architecture for the trust model in ubiquitous environments. The role is dynamically assigned to the service requester according to the context. If the service requester requests a service that is not permitted for his role, Naive Bayes classifier is used to make decision based on the service provider's own prior knowledge as well as the recommendations for the service requester. Our trust model can dynamically make decisions due to the changing context and each service provider's own judging standard. It is also scalable since the role-based architecture used in our trust model is potential for reducing the complexity in large network. In addition, our trust model is suitable to make decision with limited information, which is usually the case in a real scenario.

1. INTRODUCTION

In ubiquitous computing environments, there may include interaction with devices and services without the same owners, and no prior knowledge of the character or background of each other's impersonate identity [1]. Traditional authentication and access control are effective only in situations that the system knows in advance which users are going to access and what their access rights are. Hence computational models of trust have been proposed for ubiquitous environments which are capable of deciding whether to provide services to requesters who are either unfamiliar with the service providers or do not have enough access rights to certain services.

Trust is the quantified belief by a trustor with respect to the competence, honest, security, and dependability of a trustee within a specified context [2]. Previous trust models used various time-consuming approaches to evaluate the trust value by considering different factors that may affect the trust decision. However, a common failing is that these existing models did not address ubiquitous applications where context is dynamic and the trust decision on an entity must continuously adapt based on the context. At the same time, since different service requesters have different acceptance levels to the ubiquitous environments, the threshold for each service requester to make the trust decision should be dynamically changed.

The object of this paper is to propose a trust model that can dynamically make trust decisions based on different context as well as each service provider's own accept level in ubiquitous environments. This paper sets the stage by introducing a novel role-based architecture for the trust model. If the service requester requests a service that is not permitted for his role, Naive Bayes classifier is used to make decision based on the service provider's own prior knowledge as well as the recommendations for the service requester. The advantages of our trust model are: (1) It is flexible since it can dynamically make decisions due to the changing context and each service provider's own judging standard. (2) It is scalable since role-based architecture is potential for reducing the complexity in large network. (3) It can make use of limited information in decision making, which is usually the case in a real scenario.

The rest of the paper is organized as follows. We introduce some example scenarios to indicate our motivation in section 2. And we present the proposed trust model in details in section 3. Section 4 briefly introduces the related works. Finally, conclusions and future work are presented in Section 5.

2. TRUST MANAGEMENT CHALLENGES FOR UBIQUITOUS ENVIRONMENTS

We first discuss the following example scenarios to illustrate the motivation of our research. Suppose a guest wants to use the scanner in a ubiquitous supported smart office as shown in Fig.1, he should first use his mobile device (e.g. cell phone) to send a request to a service agent in this smart office. The service agent (e.g. USEC server in Fig.1) is in charge of several service providers in this smart office. Since the service provider's role is not enough to use the scanner, the service agent asks other members whose role is enough to use the scanner to give recommendations for the guest. Based on his own knowledge and the recommendations given by others, the scanner is able to make the trust decision on service providing. However, thresholds for different services providers to provide services may not be the same, e.g. the threshold for providing fax service may be higher than the threshold for enabling scanner service. For the same service provider, its threshold to provide service may also change from time-to-time, e.g. the threshold for scanner may be raised since it has been frequently mis-operated by users recently. The change in threshold values is related to the changes in acceptance level of service providers to the whole ubiquitous environment. The raising of the scanner's threshold means that its

* Corresponding Author

acceptance level to the smart office has been decreased due to the previous unsuccessful interactions with the users. Hence we should dynamically make the decision due to the change in usage pattern.

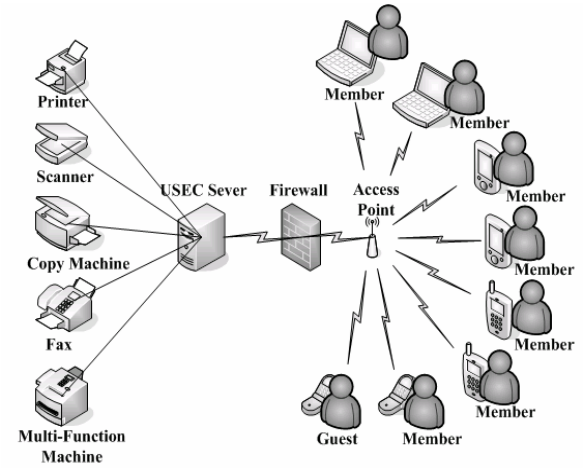


Fig.1. Ubiquitous Supported Smart Office

Recommendations and the past interaction history are the basis for the trust model to make decision, while service requester's context information should also be considered. Let us extend the above scenario into a ubiquitous supported smart building, which consists of many ubiquitous supported smart offices including professor offices, department offices, seminar rooms, classrooms and labs as shown in Fig.2. A Context-Aware Middleware for Ubiquitous Systems (CAMUS) [3] is used to capture, process and store the information about the users in this building and their activities. The possible service requesters to a service provider are the members in this smart office, the visitors from other smart office in this building or the guests to this smart building. As shown in Fig.2, when Professor A is in his office, based on the current context, CAMUS will regard his role as a professor. By using his cell phone, Professor A can freely use all the services in his smart office. When he moves to the department office, since the context has changed, he is regarded as a staff. Using his cell phone, Professor A can only use part of the services, e.g. copy machine. If he wants to use other services which have higher access requirements, e.g. fax, he should ask others whose role is enough to use the fax (e.g. dean) to give recommendations. The trust model then makes decision based on the recommendations.

The examples above embody many of the key ideas of the research presented in this paper. To maintain system security in such environments, the trust model should dynamically make decision based on the role of the service requester. Context information is used to dynamically assign the role to the service requesters as well as the recommenders. And the trust decision made by each service provider should also base on its own acceptance level to the whole environment.

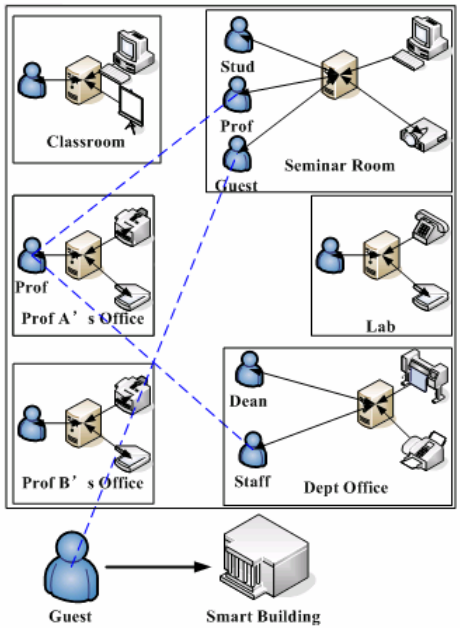
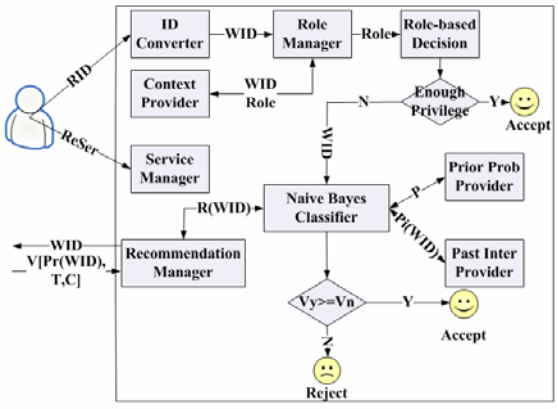


Fig.2. Smart Building Application

3. Our Proposed Trust Model

Our trust model uses the architecture shown in Fig.3 to make dynamical trust decision. The idea for the solution is that: the context information is used to decide each service requester's role. Every role has the privilege to use a set of services. For the guest to the building, a guest's role is assigned to him. If one role wants to use the service that is not allowed for him, he should ask others whose roles are permitted to use the service to give recommendations. Based on the recommendations and the role of the service requester, the service provider makes trust decision whether to provider the service. For each service provider, its thresholds for trust decision making should be dynamically updated according to its own judging standard. There are two steps included in the architecture to make trust decision. The following subsections are used to describe these steps separately.



RID: service requester's real world ID
WID: service requester's work ID in this ubiquitous environment
ReSer: requested service's ID
P: prior probability

Pi: past interaction history
T: time of last interaction
C: confidence to the service requester
Pr: recommendation value from one certain recommender
R: final recommendation value for the service requester from all recommenders

Fig.3. Trust Decision Making

3.1 Role Based Decision

When a service provider gets a request, Role Based Decision module as shown in Fig.3 is first used to make trust decision. Assume that each mobile device which acts as service requester or recommender in the ubiquitous environment is bounded to a unique real world ID (e.g. in Korea, every cell phone is bounded with user's ID card). When one uses his mobile devices to request services or give recommendations to others, its unique ID is recorded. In this way, our model is able to keep the track of different devices used in the ubiquitous environment to prevent the malicious users. When making decision, our trust model first convert the service requester's real world ID in to work ID. Work ID is the unique ID used in the ubiquitous environment and we assume that it is impossible to guess the real world ID from work ID. The use of this anonymous work ID is to prevent the recommenders to give unfair recommendations.

The Context Provider module provides the role according to the current surrounding environments of one entity. Context Provider is based on CAMUS. CAMUS envision a comprehensive middleware solution that not only focuses on providing context composition at the software level but also facilitates dynamic features retrieval at the hardware level by masking the inherent heterogeneity of environment sensors. Complexity is handled by providing "separation of concerns" between environment features extraction, contextual data composition and context interpretation. The entities and contextual information provided/utilized by them must have invariant meanings in order to have a common understanding among them. This results in sharing of information with common semantics, at different times and at different places and provides testability of formalized knowledge, emerging as a pool of consistent contextual knowledge available to different context-aware systems.

The role used here has a hierarchical structure. Role hierarchical structure helps manage role complexity to exploit commonality among roles. Fig.4 gives an example of role hierarchical structure in the smart building. By convention, more powerful roles are shown toward the top of the diagram and less powerful roles toward the bottom. Besides the services provided in his role, the service requester is permitted to use the services available for the roles whose layer is lower than him. For example, the Professor role is allowed to use all the services available for Team Leader role.

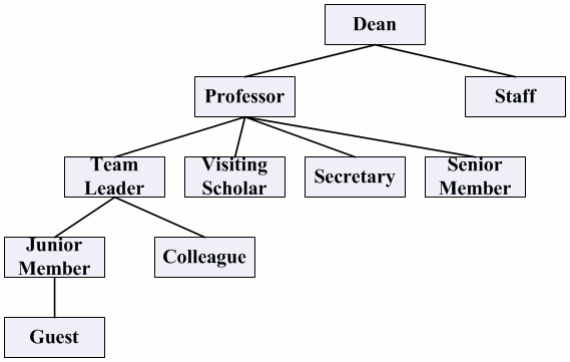


Fig.4. Role hierarchical Structure Example

3.2 Naive Bayes Classifier Based Decision

Different service providers have different acceptance levels to the ubiquitous environments, at the same time this acceptance level for one service provider can also change due to different reasons. Hence it is essential to make dynamic trust decision for different service providers at different situations. Therefore Naive Bayes classifier is utilized in our method to make dynamic trust decision. Naive Bayes is a technique for estimating probabilities of individual variable values, given a class, from training data and then to allow the use of these probabilities for classify new entities. As shown in Fig.3, Naive Bayes Classifier based trust decision is used when the service requesters do not have enough privilege to use certain services only based the permissions of their roles. There are four factors used in the Naive Bayes Classifier to make the trust decision in our model.

1. Prior Probability

Prior probability reflects the acceptance level of a service provider. It corresponds to the service provider's trusting beliefs for the whole ubiquitous environment. The prior probability keeps on adjusting since the acceptance levels of a service provider keeps on changing.

Definition 1: $P_{Sp_j}(y)$ and $P_{Sp_j}(n)$ are used to denote service provider Sp_j's prior probability of acceptance and rejection respectively. Here m is the size of training sample; k is the size of acceptance sample.

$$P_{Sp_j}(y) = \begin{cases} \frac{k}{m} & m \neq 0, \\ 0 & m = 0, \end{cases} \text{ where } j, m, k \in N, k \le m,$$

$$P_{Sp_j}(n) = 1 - P_{Sp_j}(y).$$

In case $P_{Sp_i}(y) > P_{Sp_j}(y)$, service provider Sp_i is more likely to provide the service. This situation is similar to the one we have in our social society, Sp_i is easier to believe others comparing with Sp_j.

2. Past Interaction History

Past interaction history is an entity's prior knowledge (this entity can be a recommender or service provider in our model) of acceptance to certain service requester. Past interaction history is different from prior probability since it corresponds to one entity's trusting belief to certain service requester while prior probability corresponds to service provider's trusting beliefs to the whole environment.

Definition 3: $Pi(S_i, S_j)$ is used to denote the past interaction history between entities S_i and S_j. Entity S_i and S_j can be service requester, service provider or recommender.

$$Pi(S_i, S_j) = \begin{cases} \dfrac{n-(m-n)}{m} & m \neq 0, \\ 0 & m = 0, \end{cases}$$

where $i, j, m, n \in N$, $i \neq j$, $n \leq m$. Here m is the total communication times between entity S_i and S_j. And n is the successful communication times between entity S_i and S_j. $Pi(S_i, S_j) \in [-1,1]$.

If S_i never communicate with S_j before, then $Pi(S_i, S_j) = 0$. If S_i and S_j have unpleasant interaction history, our model set $Pi(S_i, S_j) \in [-1, 0)$, which is convenient to differentiate the unknown entity from malicious entity.

3. Time Based Evaluation

Intuitively, very old experiences of peers should have less effect in recommendation over new ones. Thus we take into account the time based evaluation.

Definition 4: $T(R_k, Sr_i)$ is the time based operator for recommender R_k to service requester Sr_i. Suppose we choose a time window $[t_m, t_n]$, $\Delta\tau_0 = t_m - t_n$.

$$T(R_k, Sr_i) = \eta \frac{t_{R_k, Sr_i} - t_m}{\Delta\tau_0},$$

where t_{R_k, Sr_i} denotes the time when last communication between R_k and Sr_i happened. And η is time adapting operator.

4. Peer Recommendation

Apparently if recommender R_k had more interactions with service requester Sr_i, the recommendation given by R_k should be more importance for decision making, which introduces the notion of confidence.

Definition 5: $C(R_k, Sr_i)$ is used to denote recommender R_k's confidence to service requester Sr_i.

$$C(R_k, Sr_i) = \frac{1}{\sqrt{2\pi} * std(M)} \exp\left(-\frac{(m_k - mean(M))^2}{2 * std(M)^2}\right),$$

where $i, k \in N$, $M[k] = m_k$, $k = 1, 2, ..., n$. Here m_k is the communication times between R_k and Sr_i. We suppose that m has Gaussian distribution characterized by a mean and standard deviation. Here $mean(M)$ is the mean of $M[k]$. And $std(M)$ is the standard deviation of $M[k]$.

Definition 6: $\Pr(R_k, Sr_i)$ is used to denote the peer recommendations from recommender R_k to service requester Sr_i, $k, i \in N$. Peer recommendations for certain service requester from different recommenders are independent of each other.

$$\Pr(R_k, Sr_i) = C(R_k, Sr_i) * \frac{n_k}{m_k} * \frac{T(R_k, Sr_i)}{\Delta\tau_0},$$

where m_k and n_k are the total communication times and successful communication times between R_k and Sr_i respectively.

The final recommendation is the aggregate of the peer recommendations.

Definition 7: $R(Sr_i)$ is used to denote the aggregate of recommendations for Sr_i from all the recommenders in the ubiquitous computing environment.

$$R(Sr_i) = \frac{\sum_{k=1}^{n} \Pr(R_k, Sr_i)}{n},$$

where $k, n, i \in N$. And n is the number of the recommenders.

Using the above factors, our trust model uses Naive Bayes classifier to make the trust decision based on each service provider's acceptance level.

When service requester Sr_i gives a request to service provider Sp_j, $h(Sr_i, Sp_j)$ is used to denote Sp_j's trust decision. Accept=1; Reject=0.

$$h(Sr_i, Sp_j) = \begin{cases} 1 & V_{NB|y} \geq V_{NB|n} \\ 0 & V_{NB|y} < V_{NB|n} \end{cases},$$

$V_{NB|y} / V_{NB|n}$: the acceptance/rejection value.

Using Naive Bayes classier:

$$V_{NB} = \arg\max_{v_m \in V} P_q(v_m) \prod_n P(a_n|v_m)$$

$$= \arg\max_{v_m \in \{yes,no\}} P_q(v_m) \prod_n P(a_n|v_m)$$

$$V_{NB|y} = P_{Sp_j}(y) f_{\mu_y,\sigma_y}(R_{Sr_i}) f_{\mu_{y'},\sigma_{y'}}(Pi(Sr_i, Sp_j))$$

$$V_{NB|n} = P_{Sp_j}(n) f_{\mu_n,\sigma_n}(R_{Sr_i}) f_{\mu_{n'},\sigma_{n'}}(Pi(Sr_i, Sp_j))$$

$f_{\mu_y,\sigma_y}(R(Sr_i)) / f_{\mu_n,\sigma_n}(R(Sr_i))$: the probability of $R(Sr_i)$ when given acceptance/ rejection.

$f_{\mu'_y,\sigma'_y}(Pi(Sr_i, Sp_j)) / f_{\mu'_n,\sigma'_n}(Pi(Sr_i, Sp_j))$: the probability of $Pi(Sr_i, Sp_j)$ when given acceptance/ rejection.

$$f_{\mu_y,\sigma_y}(R(Sr_i)) = \frac{1}{\sqrt{2\pi}\sigma_y} \exp(-\frac{(R(Sr_i)-\mu_y)^2}{2\sigma_y^2})$$

$$f_{\mu_n,\sigma_n}(R(Sr_i)) = \frac{1}{\sqrt{2\pi}\sigma_n} \exp(-\frac{(R(Sr_i)-\mu_n)^2}{2\sigma_n^2})$$

μ_y / μ_n (μ'_y / μ'_n): Mean of $R(Sr_i)$ ($Pi(Sr_i, Sp_j)$) when given accepted/rejected.

σ_y / σ_n (σ'_y / σ'_n): Standard deviation of $R(Sr_i)$ ($Pi(Sr_i, Sp_j)$) when given accepted/ rejected.

As shown above, when making trust decision using Naive Bayes Classifier, our trust model compares the value of $V_{NB|y}$ and $V_{NB|n}$. The calculation of $V_{NB|y}$ and $V_{NB|n}$ involves different factors as well as the prior probability, which makes our trust decision dynamically changes according to the acceptance level of each service provider.

4. RELATED WORK

Since mid '90s the research on the key role of trust management models has been outlined in [4] [5] [6] to develop complex and dependable computer systems. In the field of ubiquitous computing, researchers have paid much more attention to build autonomous trust management as fundamental building block to design the future security framework, such as [7][8][9][10] [11].

A general concept of dynamic trust model in ubiquitous computing environments had been given in [12]. In [13], the authors explained basic scenarios in ubiquitous computing and modeling requirements of trust. A solution to evaluate trust from the past experience was given in [14]. In [15], the authors proposed a role-based trust model in ubiquitous environment, where recommendations were used to make decision. Trust level (a measure of one's belief in the honesty, competence and dependability to a certain entity) was used to make decision in [16]. The trust was divided into 6 levels and operators such as time and distance were used to evaluate the trust levels. In [17], the authors involved the concept of confidence, which reflects the communication frequency between two entities, in the trust evaluation. Trust values and confidence values were used to made the finally decision together. In [18], the authors proposed a novel Cloud-Based trust model to solve uncertain problem. These works involved great efforts to evaluate the trust values, however, when it came to decision making based on these trust values, they did not consider the context based role of each service requester and their trust decision can not dynamically change due to the altering of the service provider's acceptance level to the environment.

5. CONCLUSION AND FUTURE WORK

Our trust decision making reduces the computing complexity of the trust model in large scale ubiquitous environments by introducing a role-based architecture. The role is dynamically assigned to the service requesters based on the context. When the service requester's role is not enough to access the service, the recommendations are need for the service providers to make the trust decision. We use Naive Bayes classifier to make trust decisions based on the recommendations given by recommenders. Our trust evaluation is based on each entity's own prior knowledge in stead of using common evaluation and pre-defined weight values, which effectively reduce the subjectivity by human opinions compare with the other trust models.

We will add risk analysis in the coming work, since trust and risk always tightly coupled with each other. Other works like how to choose reliable recommenders to avoid unfair recommendations in ubiquitous trust model will also be involved in the coming work. We also propose to implement our trust model to be used in CAMUS in the future work.

ACKNOWLEDGMENT. This research was supported by the MIC (Ministry of Information and Communications), Korea under the ITRC (Information Technology Research Center) support program supervised by the IITA (Institute of Information Technology Assessment) in collaboration with Sun-Moon University.

6. REFERENCES

[1] Kagal, Finin, Joshi, "Trust-based Security in Pervasive Computing Environments". IEEE Computer, December 2001

[2] Steffen Staab, Bharat Bhargava, Leszek Lilien, Arnon Rosenthal et al. The pudding of trust. IEEE Intelligent System, Volume 19, Issue 5 (September 2004) 74-88.

[3] Hung Q. Ngo, Anjum Shehzad, Saad Liaquat Kiani, Maria Riaz, Kim Anh Ngoc , Sungyong Lee.: Developing Context-aware Ubiquitous Computing Systems with a Unified Middleware FrameWork. The 2004 Internation Conference on Embedded & Ubiquitous Computing(EUC2004),

[4] M. Blaze, J. Feigenbaum, and J,Lacy.: Decentralized trust management. In proceedings of the 1996 IEEE Symposium on Security and Privacy. (1996) 164-173

[5] A.Josang.: The right type of trust of distributed systems. In New security paradigms workshop, Lake Arrowhead (CA, USA). (1996) 119-131
[6] Colin English, Paddy Nixon.: Dynamic Trust Models for Ubiquitous Computing Environments. First Workshop on Security in Ubiquitous Computing at the Fourth Annual Conference on Ubiquitous Computing (Ubicomp2002).
[7] Kumar Ranganathan.: Trustworthy Pervasive Computing : The Hard Security Problems. Second IEEE Annual Conference on Pervasive Computing and Communications Workshops. (2004) 117
[8] Brian Shand, Nathan Dimmock, and Jean Bacon.: Trust for Ubiquitous, Transparent Collaboration. ACM: Special issue: Pervasive computing and communications. (2004)711- 721
[9] Zhaoyu Liu, Anthony W.Jo y, and Robert A. Thompson.: A A Dynamic Trust Model for Mobile Ad Hoc NetWorks. 10th IEEE International Workshop on Future Trends of Distributed Computing Systems (FTDCS'04). (2004) 80-85
[10] George Theodorakopoulos John S. Baras.: Trust Evaluation in Ad-Hoc Networks. Proceedings of the 2004 ACM workshop on Wireless security. USA. (2004) 1-10
[11] Rui He, Jianwei Niu, Man Yuan, Jiangping Hu.: A Novel Cloud-Based Trust Model for Pervasive Computing. The Fourth International Conference on Computer and Information Technology (CIT'04). 693-670
[12] S. P. Marsh. Formalising Trust as a Computational Concept. Ph.D. Thesis, University of Stirling, (1994)
[13] Pradip Lamsal.: Requirements for modeling trust in ubiquitous computing and ad hoc networks. Ad Hoc Mobile Wireless Networks - Research Seminar on Telecommunications Software, Autumn 2002.
[14] Marco D. Aime, Antonio Lioy: Incremental trust: building trust from past experience. WoWMoM 2005: IEEE International Symposium on a World of Wireless Mobile and Multimedia Networks, Taormina (Italy). (2005) 603-608
[15] R.Guha, R.Kumar, P. Raghavan, and A. Tomkins: Propagation of trust and distrust. In International Conference on World Wide Wed. USA.(2004) 403-412
[16] P. Michiardi and R. Molva.: A collaborative reputation mechanism to enforce node cooperation in mobile ad-hoc networks. In IFIP Communication and Multimedia Security Conference, Portoroz(Slovenia). (2002) 107-121
[17] S. Ganeriwal and M. B. Srivastava.: Reputation-based framework for high integrity sensor networks. In ACM Workshop on Security of ad-hoc and sensor networks. Washington(DC, USA. (2004) 66-77
[18] Cristiano Castelfranchi, Rino Falcone, and Giovanni Pezzulo.: Trust in Information Sources as a source for Trust: A Fuzzy Approach. In Proceedings of the second international joint conference on Autonomous agents and multiagent systems. (2003) 89-96