Proceedings of the International Multiconference on
Computer Science and Information Technology, pp. 253 – 261
ISSN 1896-7094

Simple Reputation Metrics for Mobile Agent in Open Environment

Mieczyslaw A. Klopotek[1,2], Michal Wolski[1]

[1] Institute of Computer Science, University of Podlasie,
ul. Sienkiewicza 51, 08-110 Siedlce, Poland
michal@iis.ap.siedlce.pl
[2] Institute of Computer Science, Polish Academy of Sciences,
ul. J.K. Ordona 21, 01-237 Warszawa, Poland
klopotek@ipipan.waw.pl

Abstract. Trust metrics are useful as a mechanism to build reputation of virtual communities. In this paper we describe and compare four reputation metrics used for mobile agents. Comparison is based on simulation of open environment, characterized by unknown network topology. In our experiments when agents begin to inspect network they don't have any information about nodes. We present our simulation environment (MARS) and "path mechanism" that is very helpful to recognize evil nodes. In last part of this paper we want to check which of well known algorithms can better recognize reputation of node.

1 Introduction

Nowdays information systems have to trust each another. By trust (or symmetrically, distrust) we mean "…a particular level of the subjective probability with which an agent will perform a particular action, both before he can monitor such action (or independently of his capacity to monitor it) and in a context in which it affects his own action." [3]

Exchange of piece of data need authentication process. When we want to explore data for Internet network or other medium we want to have trust elements which we want to collaborate. Modern Internet exploration is based on mobile agents, that work in open environment. This kind of agents can interact with one another using the particular mechanisms and protocols.

Expansion of mobile agents software is due to the business requirements that need software which will cooperate with each other without early coordination. That situation forced that agent has to have a trust to other agents before he begins transaction.

During his journey the mobile agent can interact with each of nodes and learn their behaviour over a number of encounters. Knowledge about node behavior will have to be stored in a common repository.

This situation generates some problems, which the most important are:

- unknown network topology,
- evil (hostile) nodes which can destroy agents even though sometimes evil nodes could have attractive information for agent,
- need for communication between agents member family on selected trusted nodes,
- necessity to store large resources of agents activities.

To counteract a malicious behavior of nodes, agents have to build families, so that the "death" of an individual agent will not completely erase its experience. The families need to use a common repository, where

trust information will be stored. By an agent family we understand a group of agents that have (1) a common: producer, (2) a base node where they can communicate each other, and (3) common goal for work.

In this paper we describe our experiments devoted to exploration of unknown network by mobile agents. When agents begin to inspect open environment, they don't know any information about network topology, the node hostility or friendliness. In last part of this paper we want to check which of well known algorithms can better recognize reputation of nodes under such assumptions.

2 Research environment

We can separate two different types of reputation valuations of agents to node :global one and local one. In global reputation system, there is only a single reputation value per node that is stored in common repository. Local reputation systems provide different reputation values for a node depending on the current position of agent(s) in the network.

The main subject of our research is finding solution to the above problems, and to adapt a local methods appropriately. We try to find solution by building MARS software (Mobile Agents Reputation Simulator). In this study we simulate four algorithms: eBay[7], Average , EigenTrust[1] and BetaSystem[6].

2.1 Structure of network

We have tested each of the presented algorithms on the same network, created in a random way, with topological features similar to the Internet. We investigated networks of three sizes: of 100 ,1000, and 10000 nodes.

Within a network, four groups of nodes may be encountered:

- good family nodes, composed of nodes, which have attractive information for agents,
- neutral family nodes, composed of nodes, which have nothing interesting for agents,
- bad (evil, hostile) family nodes, composed of nodes, which destroy agent in case interaction between agent and nodes[4],
- random family nodes, composed of a mixture of nodes described in the previous three groups.
- In our tests we use the node proportion structure that is presented in Table 1. Presented value describe how many nodes of particular family of nodes was used in experiment.

Table 1 . Structure of network

Kind of node	Good family nodes (25%)	Neutral family nodes (25%)	Evil family nodes (25%)	Random family nodes (25%)	Amount (all nodes) (100%)
good	100 %	0 %	0 %	48 %	36 %
neutral	0 %	100 %	0 %	44 %	37 %
evil	0 %	0 %	100 %	8 %	27 %

2.2 Simulator

MARS is Java-based simulator which allows to compare different trust algorithms. His main target, at this moment, is calculating trust value for nodes and nodes family in unknown network. Figures 1 describe MARS behavior.

MARS simulation algorithm has three important features:

- it can simulate evil nodes that can destroy agents,

- it allow to use “path algorithm”[5], that is useful to inform agents about evil nodes,
- agents have common repository to store reputation value of nodes

First property represents such situation when node is not friendly for our agent or we have a power-off failure on the network node, or agent never comes back. In the simulation when agent goes to node and node will behave unfriendly for the agent, the agent will be destroyed by the node.

Second property based on “path algorithm” is a “cure” for the first problem. Path algorithm is a sequence of actions, that allow to store information about destination of mobile agent move. When an agent wants to go to node he stores idnode and sets status to “2”. When the agent comes back or leaves message about transaction, node status is changed to “1”. If another agent wants go to the same node he checks node status. If node status is not set to “2” he goes to node. In another case he doesn't do it anything and selects other node.

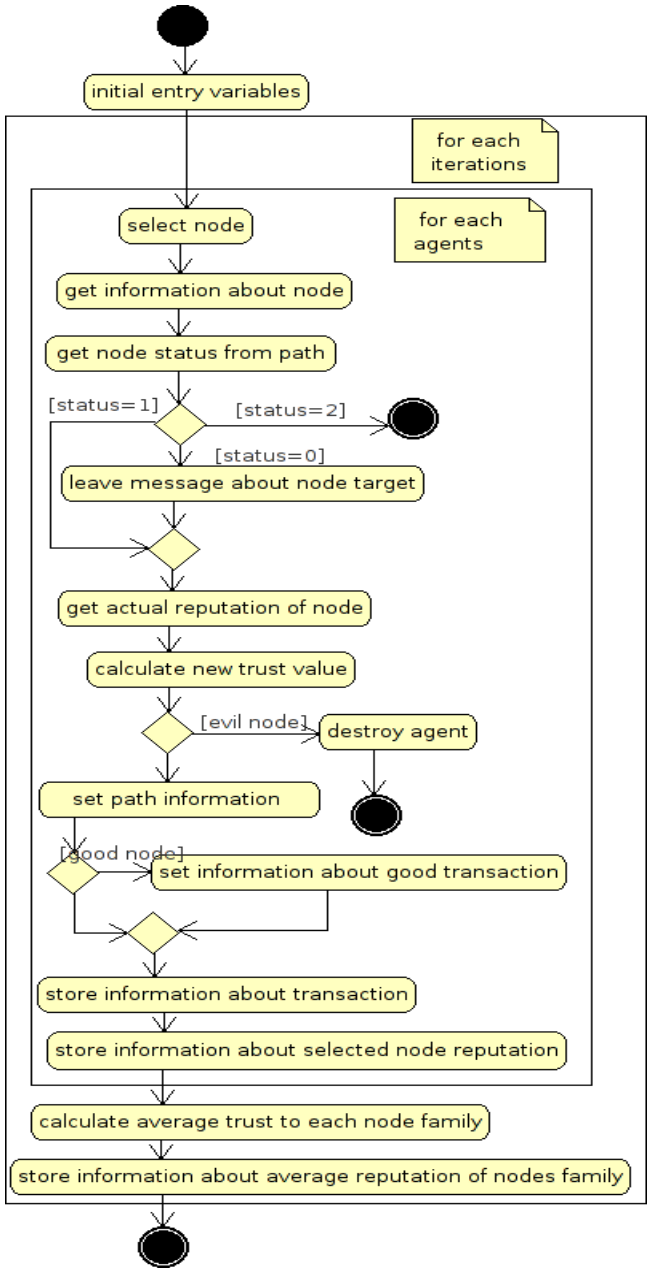

Fig. 1. MARS behaviour algorithm

Third property of MARS is realized by a common table when each agent (truster) can leave information about transaction, which happened with selected node[2], which is trustee. Information in common repository is useful to calculate the reputation value for nodes family.

It's very important to store information about nodes and their reputation, because in open environment described common repository is sole sources of information about nodes and their behavior.

3 Results

3.1 Metrics

First trust metrics, that we investigated, is well known eBay algorithm. Fig. 2 presents how trust value is growing up to 1 for family which consist only from good nodes.

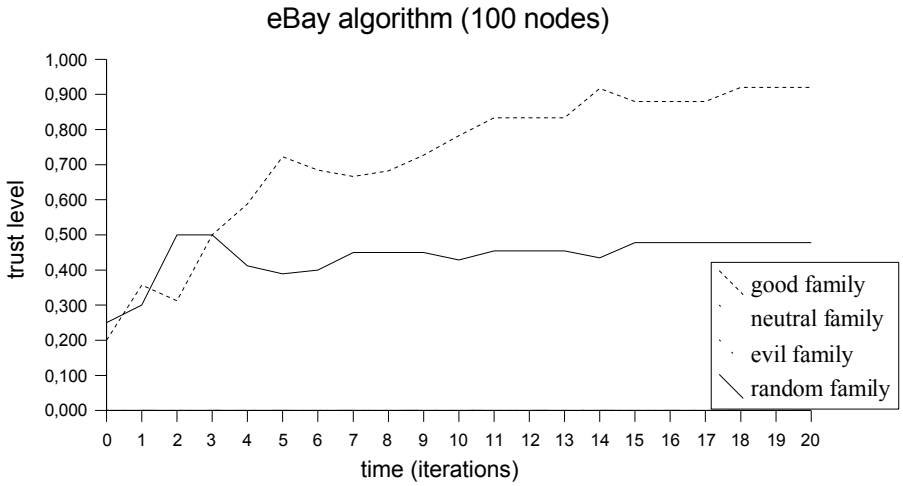

Fig. 2. Reputation of eBay algorithm for 100 nodes

Second line is trust value for random family. Reputation for this group of nodes lies near 0,5 because this family contains a lot of evil and neutral nodes. Fig. 2 do not represent evil family and nothing family because in this algorithm reputations is calculated from equation 1.

$$R_{eBay} = \frac{g}{a} \tag{1}$$

where g – good transactions

a – all transactions

Second metrics is Average algorithm that is represent by equation 2. Reputation value calculated by Average Algorithm for each family nodes except evil family is presented on Fig. 3.

This reputation metrics isn't very good for open network environment because as iterations is growing up as changes in trust are lower. Second negative value of presented Average algorithm is that each of family nodes have similar reputations.

$$R_{Average} = \frac{(t_n * a_n + t_{n+1})}{a_{n+1}} \tag{2}$$

where t - trust

a - all transactions

n - iteration

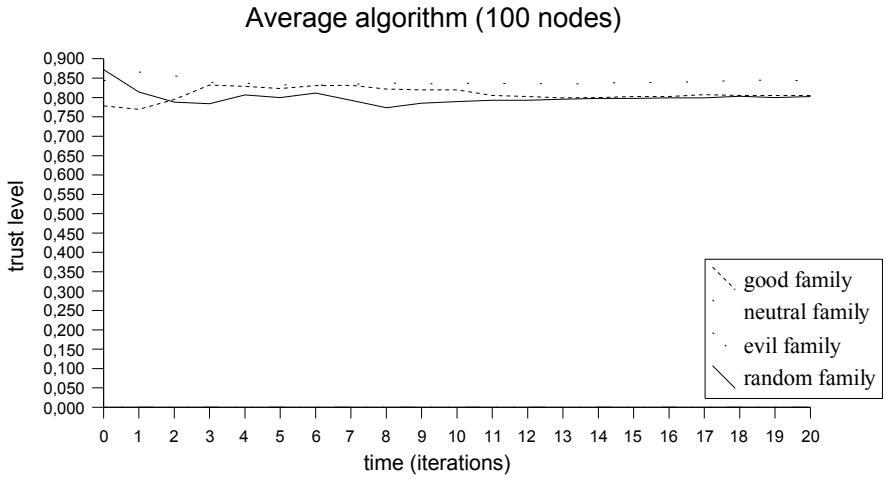

Fig. 3. Reputation of Average algorithm for 100 nodes

Following algorithm is Bayesian Reputation System called BetaSystem. BetaSystem is very easy to calculate what was shown on equation 3.

$$R_{BetaSystem} = \frac{(g+1)}{(n+g+2)} \tag{3}$$

where g – good transactions

n negative transactions

BetaSystem is based on the rule that each agent is allowed to rate node positive or negative. In our simulations positive ratting is given to good nodes, and negative ratting is obtained by neutral and bad nodes. Each of rate was stored in special table. When we want to calculate trust for particular node we have to add good and negative transactions from all transactions of selected node.

Figure 4 represents composite trust value for every explored family. Characteristic for BetaSystem algorithm is that line starts from value 0,5 and when agent collaborates with good node line is growing up, else line is going down.

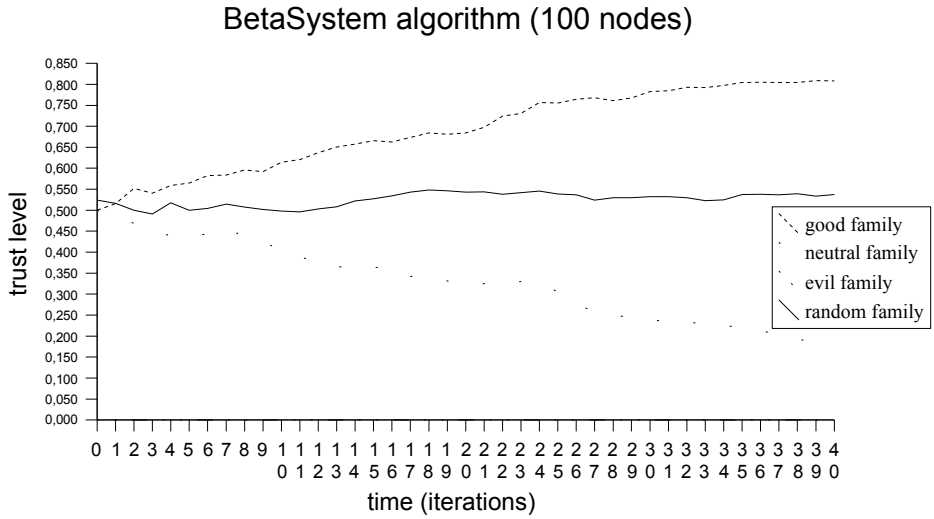

Fig. 4. Reputation of BetaSystem algorithm for 100 nodes

The last one to present is EigenTrust algorithm, that is similar to PageRank and is dedicated especially to P2P networks.

This system combines the local reputation values of each node iteratively to a global reputation. This is done by modifying a target node's reputation values stored locally at one agent *a* by the opinions of surrounding agents. These opinions are weighed according to the local reputation values an agent has about its neighbors. During this process the individual reputations are iteratively accumulated to the one single global reputation for each node. Fig. 5 represents average trust for EigenTrust algorithm in early phase of simulation. In this case reputation for node communities is not high because agent family work in accordance with EigenTrust. Family permanently collects trust for each node (see equation 4)

$$
\begin{aligned}
&\vec{t}^{(0)} = \vec{p} \\
&\textit{repeat} \\
&\qquad \vec{t}^{(k+1)} = C^T \vec{t}^{(k)}; \\
&\qquad \vec{t}^{(k+1)} = (1-a)\, t^{k+1} + a\, \vec{p}; \\
&\qquad \textit{delta} = \| t^{k+1} - t^{k} \|; \\
&\textit{until } \delta < \epsilon;
\end{aligned}
\tag{4}
$$

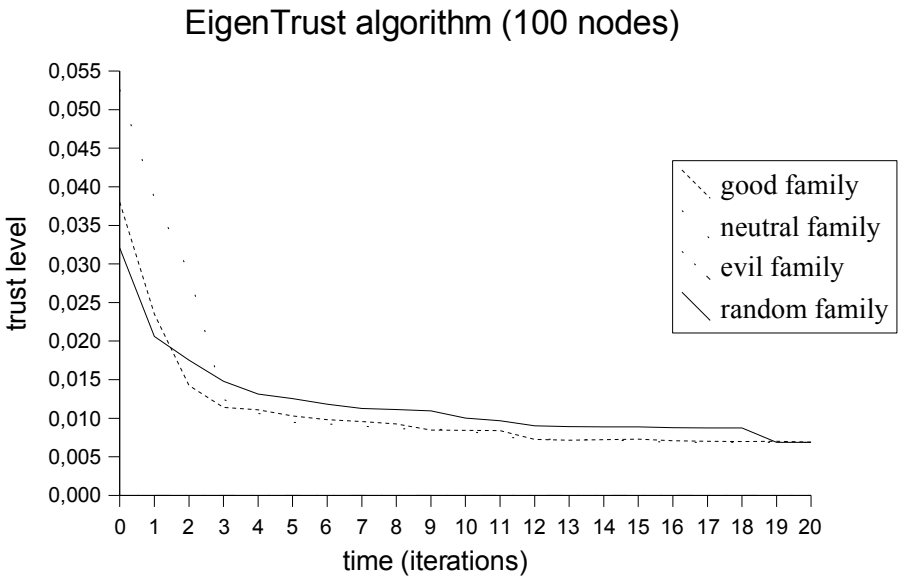

Fig. 5. Reputation ofEigenTrust algorithm for 100 nodes

Last figure (Fig. 6) shows how, during simulations, agents were destroyed by evil nodes. Because we use the same structure of the network we observe similar forms of lines.

Very important fact is that we must to have more agents than prospective evil nodes, because during learning process agents are destroyed.

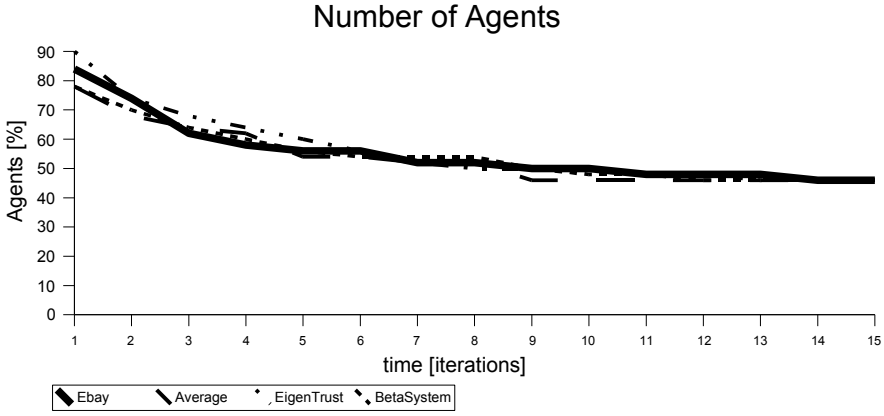

Fig. 6. Numer of agents in first 15 iterations (for 100 nodes)

3.2. Additional observations

In each experiment, irrespective of explored family nodes, agents were destroyed almost in 50% of their populations. This value directs our research to check which of explored agent family can learn network structure faster than others. When agents can learn faster, they can determine good nodes faster too.

To solve this problem we have to explore two areas. The first one is how many transactions have to be done by agents family to find all evil nodes. The second problem is, how many transactions have to be done for agents to determine true reputations for particular family nodes.

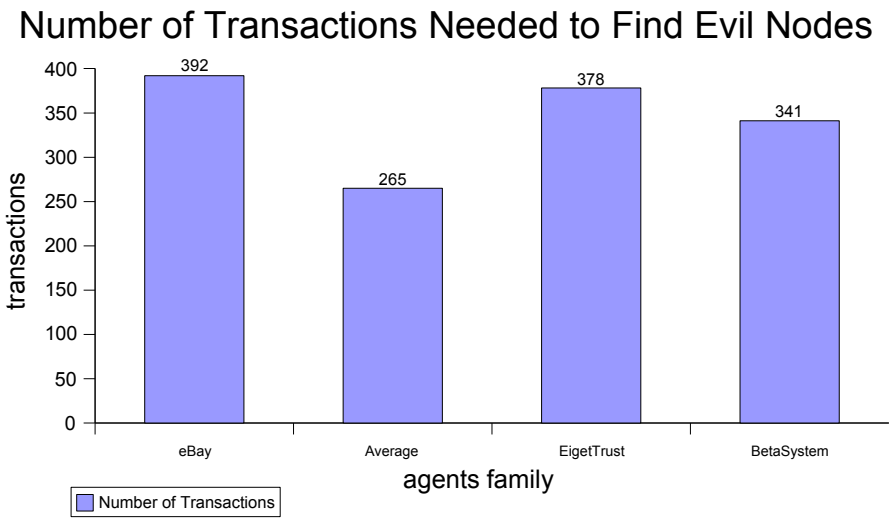

Fig. 7. Number of transactions needed to find evil nodes

Figure 7 presents how many transactions have to be done by agents to find evil nodes. From this point of view the best algorithm is Average, that need ~40 % less transactions than other algorithms.

However Fig. 8 presents how many transactions take agents to identify truly reputations of each family nodes. f we look only at this picture we can deduce that, Average algorithm will be the best of all. However when we look at previous figures we can see that Average algorithm isn't stable and we can't discern different nodes family.

In our opinion during simulations the best algorithms (in order) are BetaSystem and eBay. Our conclusion is based on the fact, that only these two trust metrics allow to discern different family nodes and set them correct reputation value. Advantages of eBay algorithm is velocity, but this metrics doesn't describe neutral family nodes. However BetaSystem algorithm is better than the others because it can discern each of family nodes. Disadvantage for BetaSystem is long time, which is needed to learn unknown network.

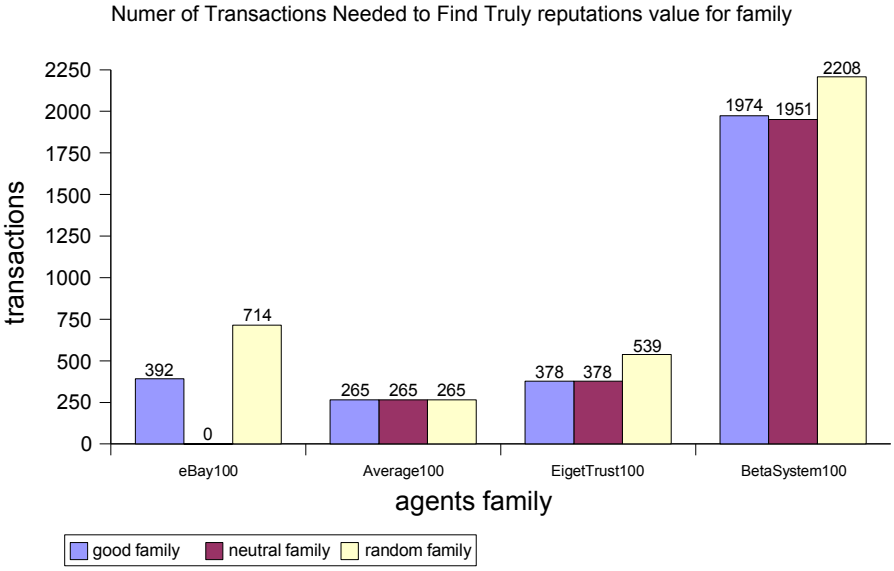

Fig. 8. Numer of transactions needed to find truly reputations of family

4 Conclusions

This article reports on a first stage of our research in open agent trust systems. In this paper we carry out a comparative simulation study of four reputation metrics that represent different trust metrics. We present the dynamics of agent families driven by those metrics, in particular investigating the convergence of estimated trust values with respect to true ones. While considering the efficiency of algorithms, we find out that not always the fastest algorithm is the best in open environment network. We recognize, that BetaSystem algorithm, among explored metrics, is the best in unknown network environment.

In the future, we plan to extend our model and the simulation framework to include simulation models of further reputation systems and more complex contexts. We want to find reputation algorithm, which allows to set faster the correct value of trust for particular family of nodes.

References

1. Kamvar S. D., Schlosser M. T., Garcia-Molina H.: *The Eigentrust Algorithm for Reputation Management in P2P Networks,* http://dbpubs.stanford.edu:8090/pub/2002-56
2. Kłopotek M. A., Wolski M.: *Comparative Study of Trust Algorithms for Mobile Agent in Open Environment,* Proceedings of Artificial Intelligence Studies Vol.**3**, (26)/2006, pp 213-220.
3. Misztal B.: *Trust in Modern Societies,* Polity Press, Cambridge, Mass., (1996).
4. Ramchurn S. D., Jennings N. R.: *Trust in agent-based software ,* Cyber Trust & Crime Prevention Project, 2004.
5. Wolski M. Kłopotek M.A.: *A Concept of Reputation for Mobile Agents Environments,* Chapter on Polish Journal of Environmental Studies Vol. **15**, No. 4C, 2006, pp 207-211, ISSN 1230-1485, Świnoujście 2006.
6. Zacharia G., Maes P.: *Trust Management through Reputation Mechanisms,* Applied Artificial Intelligence, 2000.
7. http://www.eBay.com