Modelling Suspicion as a Game Mechanism for Designing a Computer-Played Investigation Character

Nahum Alvarez[1] and Federico Peinado[2]

[1] Production Department,
Gameloft Tokyo
151-0061 Tokyo, Japan
nahum.alvarezayerza@gameloft.com
[2] Departamento de Ingeniería del Software e Inteligencia Artificial,
Facultad de Informática, Universidad Complutense de Madrid
28040 Madrid, Spain
email@federicopeinado.com

Abstract. Nowadays creating believable characters is a top trend in the video game industry. Recent projects present new design concepts and improvements in artificial intelligence that are oriented to this goal. The expected behaviour of computer-played characters becomes increasingly demanding: proactivity, autonomous decision-making, social interaction, natural communication, reasoning and knowledge management, etc. Our research project explores the possibility for one of these characters to investigate, solving an enigma while dealing with incomplete information and the lies of other characters. In this paper we propose how to manage trust issues when modelling suspicion in the artificial mind of a Columbo-like detective, considering that our desired gameplay is a murder mystery game in which the player plays the role of the culprit.

Keywords. Interactive Digital Storytelling, Video Game Design, Artificial Intelligence, Believable Characters, Epistemic Modal Logic, Trust Systems

1 Introduction

Computer simulations allow a high grade of realism; however, computer-controlled characters in virtual worlds still feel like mere automatons: the user gives an input and receives a response. Advances in intelligent behaviour are changing the user experience, improving these characters' performance and expanding the scope of their potential uses. One aspect that refrain us from believing such characters are "alive" is that they seem completely "naive": they will believe without doubts all the information received from other sources. Of course, in interactive storytelling this issue is mainly confronted with a script that the character will follow: if it has to suspect, it will suspect, if it has to lie, it will lie, etc. but there is no real intelligence behind those decisions. A system capable of managing suspicion would improve greatly the degree of realism of the characters, but such feature has not been sufficiently explored.

In this work we focus in the question of trust, intrinsic to human nature, and introduce a model for a computer-played character that suspect about others and try to detect when they are covering their actions or telling lies. In this model, the autonomous "investigator" works with incomplete information, managing uncertain knowledge and potentially false pieces of evidence. We also propose a videogame application designed to test our model. The application consists of an interactive version of the popular TV series from the NBC: *Columbo*, where the player controls the murderer instead of the main character of the story, Lieutenant Columbo, who is the homicide detective. The culprit is obvious to the player since the very beginning of the game, and the goal of this character is to deceive the detective for not getting caught (as in the original episodes, the format of the typical "whodunit" mystery is reversed to the "howcatchthem" paradigm or "inverted detective story").

In order to frame our proposal, in Section 2 we review other models of deception and trust. Section 3 presents our model of suspicion for computer-controlled characters and Section 4 describes an example scenario of how our model would work using a murder mystery game as an application. Finally, in Section 5 we present and discuss our conclusions, foreseen the next steps of this project.

2 Related Work

Modelling *deception* has been object of research from long ago. Jameson presented IMP in 1983 [11], a system that simulates a real estate agent who tries to deceive the user. Nissan and Rousseau described how to model the mental state of agents in a more playable scenario: a crime investigation interactive fiction [19]. In later works, Nissan showed a model for representing complex stories involving deceptions and *suspicion* [18]. This proposal is a theoretical model, but it established a basis to construct working systems using trust and partial knowledge. Such formal models have been presented since then, showing an integrated frame for uncertainty and deception in game theory [16].

Sometimes the character may work not only with incomplete information, but also with information that is considered as "uncertain" (depending on the sincerity of the characters that reveal it). In order to manage these features, intelligent characters have to decide which information to trust. Research on *trust* and deception itself is a broad concept with multiple aspects that have been explored in the literature, such as self-deception and how it affects decision-making [10], deceptive strategies in auction style interactions [5] or deceptive information in decision-making processes for games [14]. However, these theoretic approaches were mostly oriented to a very limited type of interactions.

Certainly, trust is a crucial aspect to deal when requesting services or information from third parties [15]. It has been extensively treated in literature but still it does not have a standard formal definition [22]. In a general classification proposed in [17], four general types of trust are defined: basic trust, a priori generalized trust, inner dialogicality and context-specific trust. The last type is defined for applications that require trust management, which is the case in a murder mystery game; for instance,

we trust a doctor about health issues when he is treating us, but not about the death of his wife when he is the main suspect in her murder. This type of trust contains context-specific rules, so we can decide if trusting or not using different methods. For example, Griffin and Moore [8] present a model that assigns different truth values to the concepts managed by autonomous agents, allowing them to choose if they trust their sources or not. These agents simulate how the situation would evolve if they take certain decisions, trying to maximize their outcome in a minimax fashion. The system of Wang et al. [24] uses fuzzy values, implying the information of a source is not definitely true or false, but it has some grade of truth, and we can also find systems using a vector of parameters for different aspects of trust [25]. Another model of trust and commitment [12] proposes that positive and negatives experiences impact in the trustee, treating all of them equally (although probably in real life negative ones have a greater impact).

Trusting a source of information does not only rely in judging over the data received from that source, but we also have to take in account all the related sources. Moreover, we also should take in account third parties' information: for example, in [3] trust is built without direct interaction between the parts using a set of relationships and trust parameters in their multi-agent model, allowing to build trust on the information of a third party agent. This transitivity is well defined in the theory of Human Plausible Reasoning [7], a frame designed to simulate the way a human reason about truth values. This frame establishes an ontology-based model that uses a fuzzy parameter for measuring the degree of trust of a statement or a source. This has been used as a basis for developing complex trust systems, showing good results.

ScubAA [1] is another example consisting of a generic framework for managing trust in multi-agent systems. A central trust manager is in charge of sending the user's request to the most trusted agent for the needed task. The trust degree of each agent is calculated using not only the information it has but also the information of the agents related to them, so an agent can evaluate agents directly unknown for it. This transitivity is useful for building a trust network where information of third parties is considered, affecting the trust of the agents, even if they are not present or currently known. We also use this approach in our proposal, but in our system, instead of designing a centralized system, each agent can manage its own trust network. Also, it is advisable to take into account the trust history of the other parties [13], in order to trust in sources that were "trustful" in the past for us or for our partners [26]. Anyway, agents have to take this information carefully, because they only receive partial information from others. This also presents an interesting feature and a powerful possibility: if a malicious agent manages to tarnish your trust level for other parties, they will not trust you in future interactions.

In our model, allowing the character to build its own trust values, makes the information they hold to be incomplete and different for each character. This feature is beneficial from our point of view, because it allows introducing deceptive information in the system without being "caught", testing the investigator's ability to find the truth.

Finally, in order to generate trust in other parties or deceive them with our lies, we have to relay in *argumentation*. In literature, different analysis about epistemology and argumentation has been examined and described thoroughly, establishing directions that allow modelling critical reasoning [2]. In short, we can trust a source only

after building argumentation backing it [23]. Especially interesting for our goal is how to build critical questions in order to attack other's argumentation, a process that will help us to discover the "holes" in an agent's argumentation, finding not only lies but also important information previously unknown [9].

3 A Model of Suspicion

Our knowledge model is designed to allow computer-controlled characters to decide if they should believe (trust in) the information received from a third party. The decision-making or judgement about a concrete piece of information will be based on the previous data they have about it, and about the source from they received that information. This paper is focused on introducing the model, so we have deliberately postponed the analysis of the more technical aspects (i.e. the reasoning engine) for further research. In order to test our model, the characters will operate in a detective scenario, a great domain example where trust and deception are key features to have into account. In this scenario, the user plays the role of a murderer who has to mislead an important non-player character: the detective. The player's goal will be to prevent the detective to discover the murderer's identity (his/her own identity). The scenario contains also other suspicious characters, each one with its own information about the incident. The detective will obtain new information (from now on, *Facts*) about the scenario, talking with other characters and finding evidences, storing all this information in a knowledge base.

In order to decide what information should trust and how, the first challenge the detective confronts is determining the truth value of Facts. As we saw in the previous section, other models are based on lists of trust parameters, fuzzy truth values, or statistical probabilities. We decided to use a discrete system using a ternary value ("true", "false" and "unknown") for each Fact. The first time the detective gets informed about a Fact, he will create an entry in his knowledge base about it, and he will update its value depending of the additional information he finds about it. *Not knowing* certain information is not the same than knowing it is true or false (i.e. open world assumption), so we use the value "unknown" in order to mark those doubtful Facts and subsequently trying to discover their real value with abductive reasoning .

The next question is how to "quantify" the truth value of a Fact. We only have found simple methods in the literature for this decision-making process: usually it is enough to use a fixed probability number or comparing the number of agents supporting a fact with the number of agents denying it. However, if the system is that simple we may lose certain desirable characteristics as taking into account evidences that support the truth or falsehood about a fact, or remembering if a character previously lied to us before evaluating her statements, especially if those statements are related with the one about she lied.

Considering these requirements, our detective will have a list of truth values about a Fact where they will store each evidence they obtain about it. Having a bigger number of supporting evidence for a Fact ("true" values) will make him to believe the statement, and on the other hand, having more contradictory evidence ("false" values)

will lead the agent to consider the statement as a lie. This is similar to the system used in [6] where the positive evidences are cancelled by negative ones. Whenever we confirm or discard a fact from a source, we will record whether he lied to us or not. If we receive information from one source, the truth value of that information would be set initially to "true", assuming the source did not lie before (so we trust in him). We would set the information as "false" if the source lied about a related topic, or "unknown" if we don't have previous information about the source, or if the source lied but about another topic.

Once we have a model describing how the information will be stored and judged, we need a reasoning method in order to figure out the truth value of unknown Facts. Although it is not the main focus of this paper to present a reasoning implementation, it is necessary to describe such method in order to show how to deal with its characteristic features: explicitly unknown values and non-trustable sources. In our model, the automatic reasoner has the goal to clear "unknown" values by giving them "false" or "true" values by identifying if assigning those values generates a contradiction or a possible outcome. We propose using two joint techniques that resemble a detective work to achieve our goal.

The first one would be using an "established knowledge" database. Facts we will receive are checked using a common sense's rules database: if some Fact contradicts a rule in that database we will know that the source is lying; for example, stating that a character was outside of a building while it was raining, but seeing that his clothes are dry. Also, we can do this Fact checking with our first person evidences: if we know a Fact for sure, any information that contradicts it would be false.

The second technique consists in trying to clear the "unknown" actively. If we want to know who the murderer is, but we have limited information including false evidence from the culprit, it's very likely that once we obtain enough information from other characters we will have some "relative contradictions", represented by having "true" and "false" values in the list of truth values obtained from different sources about the same fact. Upon finding a contradiction, the character will run two simulations, respectively binding the truth value for that fact to "true" and "false", and propagating further consequences. If the resulting knowledge base of the simulation has a contradiction with another Fact we know for sure is true (values in our "established knowledge" database), we can discard that value.

We are currently analysing reasoning tools in order to apply the most suitable one to work with our suspicion model. In order to model sophisticated scenarios, the reasoner should have planning capabilities, and be able to work with hypothetic or potential facts. For example, since our knowledge structure works with undetermined truth values, a model based in Intuitionistic Logic would fit well. Using this technique, a suitable framework would be the Hypothetical Logic of Proof [21]. This work is derived from Natural Deduction [20], which is designed to be similar to intuitive, informal reasoning, like the one we can see in detective's stories. Next steps in this research will explore the different techniques that work well with our model, presenting a working reasoning prototype. This prototype will consist on as a detective game like the one we have mentioned, which is detailed in the next section.

4 Application in a Murder Mystery Game

In order to test our model we propose a game around the concept of trust and deception. Previous researchers have also used games as well for simulating trust relations, like [6] where an investment game (a modification of Berg's game [4]) is used for analysing how users react when trust and deception is introduced to the game (showing that users trust more in a system with advisor, even if he can deceive them).

Our scenario will be an inverted detective story based in the Columbo TV episodes, where the player has the role of the culprit in a murder case, and his goal is to deceive a computer-controlled detective (a sort of virtual Columbo) and lead him to arrest another suspect instead of the player.

The simulation will play as a series of rounds in which the detective will question the player and the other suspects. Obviously, the player knows the details of the murder and he will have the opportunity to hear what other characters say in order to gain additional information from them to be used for deceiving the detective. The player can, for example, create an alibi for himself ("I was with that person at the moment of the crime"), or blame a third party ("I saw that person entering in the victim room just before the crime"). If the player manages to plant enough false evidence, the detective will arrest the wrong character and the player will win the game. However, if the detective find contradictions in the player's alibi or manages to extract the true testimony from the other suspects, he will soon discover the player as the real culprit, making him lose the game.

In order to illustrate these ideas, the next figures show how the knowledge of the detective evolves during a game session. The example is inspired in the first episode of the original Columbo series, "Murder by the Book" in which Jim Ferris, a famous writer, is killed by his colleague, Ken Franklin. As part of his alibi, Ken tries to incriminate the Mafia, dropping a false document in the crime scene and lying about the next book Jim was working on. The player plays the role of Ken, explaining his alibi to the detective, who will try to find something that contradicts Ken's suggestions. The next paragraphs explain just a small part of the complete plot of the episode, enough to illustrate the basic ideas behind our model.

Firstly, the crime is introduced (Fig. 1) with a clear fact (represented by a white box), and some basic deductions (black arrows) performed by the automatic reasoner. This puts the detective in his way to find the murderer, that could be someone close to the victim, or not (red arrows with a white head represents disjoint Facts).

Fig. 1. The crime is introduced and basic deductions are performed

Then, the interrogation starts (Fig. 2) and the detective asks Joanna, Jim's wife, about the enemies of his husband. If Joanna tells the truth, which initially is "believable" (discontinuous line) because there are no contradictions, Jim has no enemies. That means the murderer could be Joanna herself or Ken, his closest friend.

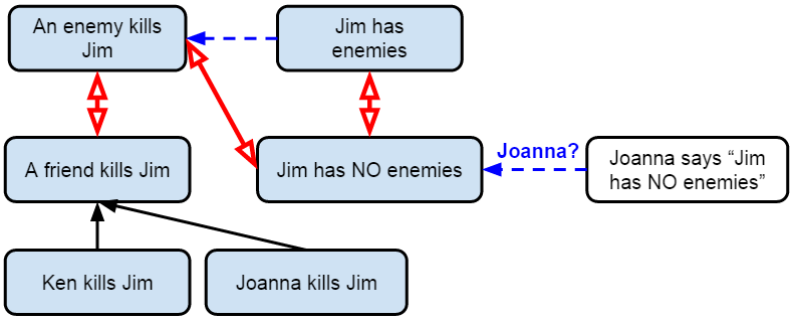

Fig. 2. Joanna is interrogated, adding facts and alternatives about the murderer

Considering the complete case, there are still many unknown Facts, so the interrogation continues (Fig. 3), this time asking Ken about his friend. As the real culprit, the player may try to deceive the detective lying about the existence of a secret project: Jim was writing a book about the Mafia. He could probably reinforce the idea with more Facts, explicitly lying about Jim and his "enemies". This contradiction makes "Jim has enemies" an unknown Fact that the detective will investigate.

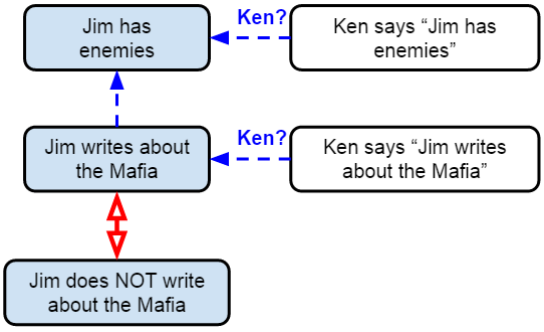

Fig. 3. Ken is interrogated, adding lies and contradictory facts

Later on the game session, a false document is found in the crime scene (Fig. 4). As the person who put that document there, Ken has been waiting for this moment and its "dramatic effect". A list of names of dangerous Mafia members is found in the desk of the victim. The reasoner of the computer-controlled detective finds plausible that the writer created that list, so it seems evident that Jim was really working on that book.

But at the end of the game session, looking for more clues to solve other parts of the mystery, more evidence appears: the detective notices that the document has been

folded as if someone stored it in his pocket (Fig. 5). It does not make sense that Jim created that list and folded it before putting on his desk, so other person should have done it, possibly trying to incriminate the Mafia instead of the real murderer. So Ken' story about the Mafia was a lie, and probably previous Facts coming from him should be considered "false". Now the detective believes Ken is a liar and one of the main suspects: a good approach to the solution of the whole case.

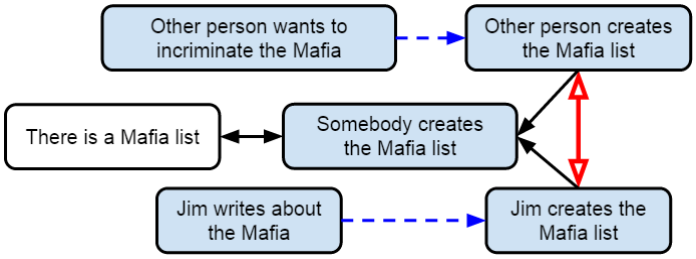

Fig. 4. Evidence is found in the crime scene. Ken's information seems to be true

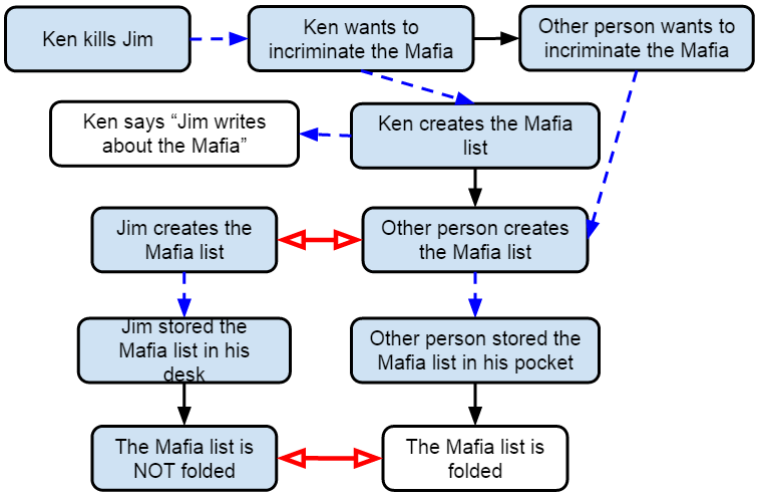

Fig. 5. Another piece of evidence. The detective will not trust Ken anymore

5 Conclusions

In this paper we surveyed previous works on computational trust and deception and proposed a model for computer-controlled characters that suspects from others, not trusting any information received from third parties. Previous work on trust models relies on sources that generally do not lie, but in our case that assumption cannot be accepted. Our model is designed for situations with incomplete information and a

constant need of judge what is likely to be true or false. Deciding who to trust is a key process in real life that can support and enrich a very wide range of games.

We used a ternary depiction for the truth value of facts and an automatic reasoner for inferring potential outcomes starting from the known facts. The model manages a list of potential truth values for each Fact, even coming from information given by a third party, selecting the value that appears the most from trustful sources. It also marks the other parties as trustful or not whenever it discards contradictory facts or accepts valid ones coming from them. If the model cannot decide the truth value of something (hence, it has the same number of supporting and refuting evidence), it will mark it as "unknown". We described superficially how a reasoning engine could work using this model, trying to clear "unknown" values by reasoning over the potential outcomes that any of the truth values could generate. Such engine needs to be explored more in depth after a comparison of potential techniques, and will be the focus of our next steps. We also designed a game mechanism that is illustrated with an example scenario based in the first episode of the Columbo TV series for developing our model. In this gameplay, the player takes the role of the culprit in a murder case and a computer-controlled detective tries to discover who did it by looking for evidence and questioning suspects, including the player.

As the next steps of our research, we are currently working on a computational model that implements this example scenario as a simple text-based dialogue, allowing the user to play the culprit role with the computer acting as the detective. Also, we plan to test the scenario, containing the complete murder mystery, with real users taking the roles of the culprit and the detective in a "Wizard of Oz experiment" where we can analyse how they act and what kind of questions are asked in order to evade suspicion or discovering lies. With the results of this experiment, we will establish a behavioural baseline for comparing our model with the one someone in the role of a detective would have. Then we will re-enact the experiment, but this time using our application where the users only play the culprit role, and we will compare the behaviours of the human detective and the computational one.

Further research opens interesting paths: we want to analyse the nature of deception as well, so we want to expand our model allowing the rest of the non-player characters to decide when they should lie in order to cover their alibi, or hide private information. We think that creating a working model for trust and deception would be a useful contribution to interactive storytelling and video games, and to the Artificial Intelligence community as well, because it is a feature not fully explored yet.

References

1. Abedinzadeh, S., Sadahoui S.: A trust-based service suggestion system using human plausible reasoning. Applied Intelligence 41, 1 (2014): 55-75.
2. Amgoud, L., Cayrol, C.: A reasoning model based on the production of acceptable arguments. Annals of Mathematics and Artificial Intelligence 34, 1-3 (2002): 197-215.
3. Barber, K.S., Fullan, K., Kim, J.: Challenges for trust, fraud and deception research in multi-agent systems. Trust, Reputation, and Security: Theories and Practice. Springer Berlin Heidelberg (2003): 8-14.

4. Berg, J., Dickhaut, J., Mccabe, K.: Trust, reciprocity, and social history. Games and Economic Behavior 10, 1 (1995): 122-142.
5. Broin, P.Ó., O'Riordan C.: An evolutionary approach to deception in multi-agent systems. Artificial Intelligence Review 27, 4 (2007): 257-271.
6. Buntain, C., Azaria A., Kraus S.: Leveraging fee-based, imperfect advisors in human-agent games of trust. AAAI Conference on Artificial Intelligence (2014).
7. Collins, A., Michalski, R.: The logic of plausible reasoning: A core theory. Cognitive Science 13, 1 (1989): 1-49.
8. Griffin, C., Moore, K.: A framework for modeling decision making and deception with semantic information. Security and Privacy Workshops, IEEE Symposium (2012).
9. Godden, D.J., Walton, D.: Advances in the theory of argumentation schemes and critical questions. Informal Logic 27, 3 (2007): 267-292.
10. Ito, J. Y., Pynadath, D. V., Marsella, S. C. Modeling self-deception within a decision-theoretic framework. Autonomous Agents and Multi-Agent Systems 20, 1 (2010): 3-13.
11. Jameson, A.: Impression monitoring in evaluation-oriented dialog: The role of the listener's assumed expectations and values in the generation of informative statements. International Joint Conference on Artificial intelligence 2 (1983): 616-620.
12. Kalia, A.K.: The semantic interpretation of trust in multiagent interactions. AAAI Conference on Artificial Intelligence (2014).
13. Lewicki, R.J. and Bunker, B.B.: Trust in relationships: A model of development and decline. Jossey-Bass (1995).
14. Li, D. , Cruz, J.B.: Information, decision-making and deception in games. Decision Support Systems 47, 4 (2009): 518-527.
15. Li, L., Wang, Y.: The roadmap of trust and trust evaluation in web applications and web services. Advanced Web Services. Springer New York (2014): 75-99.
16. Ma, Z.S.: Towards an extended evolutionary game theory with survival analysis and agreement algorithms for modeling uncertainty, vulnerability, and deception. Artificial Intelligence and Computational Intelligence. Springer Berlin Heidelberg (2009): 608-618.
17. Marková, I., Gillespie, A.: Trust and distrust: Sociocultural perspectives. Information Age Publishing, Inc. (2008).
18. Nissan, E.: Epistemic formulae, argument structures, and a narrative on identity and deception: a formal representation from the AJIT subproject within AURANGZEB. Annals of Mathematics and Artificial Intelligence 54, 4 (2008): 293-362.
19. Nissan, E., Rousseau, D.: Towards AI formalisms for legal evidence. Foundations of Intelligent Systems. Springer Berlin Heidelberg (1997): 328-337.
20. Prawitz, D. Natural Deduction. A Proof-Theoretical Study. Almqvist & Wiksell, Stockholm (1965).
21. Steren, G., Bonelli, E. Intuitionistic hypothetical logic of proofs. Electronic Notes in Theoretical Computer Science, 300 (2014): 89-103.
22. Walterbusch, M., Graüler. M., Teuteberg, F.: How trust is defined: A qualitative and quantitative analysis of scientific literature (2014).
23. Walton, D.: The three bases for the enthymeme: A dialogical theory. Journal of Applied Logic 6, 3 (2008): 361-379.
24. Wang, Y., Lin, K.J., Wong, D.S., Varadharajan, V.: Trust management towards service-oriented applications. Service Oriented Computing and Applications 3, 2 (2009): 129-146.
25. Xiong, L., Liu, L.: PeerTrust: Supporting reputation-based trust for peer-to-peer electronic communities. Knowledge and Data Engineering, IEEE Transactions 16, 7 (2004): 843-857.
26. Zhou, R. Hwang, K.: PowerTrust: A robust and scalable reputation system for trusted peer-to-peer computing. Parallel and Distributed Systems, IEEE Transactions 18, 4 (2007): 460-473.