Rethinking NPC Intelligence - A New Reputation System

John Mooney*, Jan M. Allbeck†
Games and Intelligent Animation Laboratory
George Mason University

Abstract

Creating more believable Non-Player Characters (NPCs) is a significant challenge for video game researchers and industry designers alike. While researchers explore a myriad of solutions, one somewhat forgotten solution area is NPC reputation systems. In this paper, we describe a redefined reputation system for NPC characters that allows for more realistic and dynamic social relationships. Our reputation system focuses on an agent's ability to remember and share observed behavior of other actors in the world. With this knowledge, NPCs can predict behavior of other actors, react according to their own subjective opinion, and exhibit more believable behavior to further immerse the player in the game world.

CR Categories: I.2.1 [Artificial Intelligence]: Applications and Expert Systems—Games K.8.1 [Personal Computing]: General—Games;

Keywords: reputation, intelligent agents, npc behavior

1 Introduction

The quality of a video game is often analyzed through the lens of immersion. How successfully does the game capture a player's attention? Does the player 'lose' himself in the world? There is significant research on how to improve a game's immersion factor, particularly relating to Non-Player Characters. For NPCs, immersion stems from believability; if the player's experience matches his expectations, than believability is achieved. [Jennett et al. 2008] The goal of creating more believable NPCs has spurred research across numerous areas. The Belief-Desire-Intention (BDI) model describes agents that act to achieve an explicitly defined set of goals according to a set of beliefs. [Woolridge 2003] Emotion models are also a growing area of research. [Ennis et al. 2013; Rumbell et al. 2012] These models increase believability by altering an NPC's behavior or appearance to better mimic human emotion. Personality models have also been created to give unique qualities to NPCs. [Egges et al. 2004] Lastly, there is work to more effectively model the social relationships that NPCs create, particularly with the player. [Ochs et al. 2009; Dias and Paiva 2013] The research on more believable agents has had much success, however one particular area that is largely ignored by both academic and industry patrons is reputation systems.

A reputation system is the method through which an actor, primarily the player, is generally 'seen' or represented across the NPCs in a game world. Reputation is the collective opinion of an actor within a community. In video games, particularly role-playing games (RPGs), this is often described as 'faction', 'favor', or 'reputation.' Video games often implement a global-scalar view of reputation, where a community collectively shares some positive or negative number describing the favor of a character. Consequently, the games industry has recognized some fallbacks with this model. Otello is a reputation system that aims to overcome some fallbacks of the global-scalar model. [Sellers 2008]

The work presented in this paper is inspired by all the previous efforts to create more believable Non-Player Characters in games. Particularly, we are motivated to create a reputation system that complements current academic work and builds upon reputation systems developed in the games industry. We begin our discussion with an overview of current state-of-the-art, particularly emotion, personality, and social relationship models in academia and reputation models in industry. From these solutions, we propose a reputation system that allows agents to construct and share subjective knowledge of actors in the world. We highlight the ability for an NPC to predict and react to behaviors of other entities, demonstrating more human-like behaviors. Reputation and trust are redefined to give a more generalized representation, and we describe an extended view of relationships that includes both an agent's subjective opinion of an actor and his memory of the actions an agent has taken. Finally, we highlight the key contributions of our design and discuss its drawbacks and potential improvements for later iterations.

2 Related Work

Over the past few years, researchers have made significant strides in creating more believable NPC agents for game worlds. Acting as a foundation in artificial intelligence is the Belief-Desire-Intention model. [Rao et al. 1995; Woolridge 2003] The BDI model simulates three key aspects of human reasoning to model natural decision making. Much work on NPC artificial intelligence stems from the BDI design. Researchers have also developed more accurate emotion models to increase believability in NPC agents. Some researchers, such as Ennis and Egges, focus on an agent's portrayal of emotion. [Ennis et al. 2013] Others, such as Rumbell et. all. analyze how an agent's emotions can improve action selection and behavior. [Rumbell et al. 2012] Additionally, NPC emotion models are often developed alongside specialized personality or social models. For instance, Egges, Kshirsagar, and Thalmann combine all three mechanisms and describe a generic model for updating conversational agents' emotions and personalities. [Egges et al. 2004] Dias and Paiva propose a method for establishing and strengthening social relationships between agents according to the BDI model. Their agent's also express some notion of emotional intelligence in their relationship building. [Dias and Paiva 2013] Ochs et. al. also explore NPC emotion models, particularly how an NPC's personality, emotion, and social relationships effect his behavior. [Ochs et al. 2009] These works only reflect a small subset of the work on emotion, personality, and social relationships of NPCs and agents. Our work is inspired by and intends to complement much of the work in emotionally aware agents.

Another relevant area of research is multi-agent systems, particularly reputation and trust between cooperating agents. Panait and Luke provide an overview of reputation and trust from the coopera-

*e-mail:jmooney3@gmu.edu
†jallbeck@gmu.edu

tive multi-agent perspective. [Panait and Luke 2005] They highlight the importance of reputation and trust in multi-agent systems, particularly in overcoming many challenges across a broad spectrum of applications. Panait identifies a focus on challenges of security and optimization for these systems; two challenges which do not hold significant merit when developing an NPC reputation system. Similarly to Panait and Luke, Pinyol and Sabater-Mir give a review of reputation and trust models for multiagent systems. [Pinyol and Sabater-Mir 2013] Despite describing more recent trends in the multi-agent systems community, much of the work is still specific to security and robotics domains. Despite differences between the communities, our reputation system is able to draw from research in the multi-agent systems community. For our work, we are motivated by high-level solution designs, particularly machine learning approaches, that are overviewed within Pinait and Pinyols' works.

Lastly, our work is largely motivated by state-of-the-art reputation systems found in the industry. Reputation systems are very common among role playing games. A player usually develops his reputation across various communities of the game world, and often his reputation will affect gameplay mechanics. For instance, a player with high reputation may unlock more items or quests within a specific community, or a player with large negative favor may be attacked on sight. The industry generally represents reputation according to the single-value design. That is a community of NPCs all share the same 'likeness' value for the player. For instance, the massively popular game *World of Warcraft* by Blizzard employs this system. The player has a reputation in each town that represents how members of that town feel about the player, either positively or negatively. Similar systems are at work within other largely successful RPGs: Square Enix's *Final Fantasy XIV* and Bethesda Softwork's *The Elder Scrolls V: Skyrim* [BlizzardEntertainment 2004; SquareEnix 2013; BethesdaSoftworks 2011] The limitations of the single-value system has motivated our work in this paper. Michael Sellers of Online Alchemy also recognizes some of the issues with the single-scalar representation. [Sellers 2008] The Otello system recognizes the importance of bias between participants in the community; each individual must form his own opinion. The system constructs a social graph to disseminate information between participants, and users can place differing values of trust in those around them. While Otello does improve upon the single-value system of modern industry games, information is still disseminated immediately and without physical bias. Reputation is still limited to be either a positive or negative value. These are motivations to create a system with a more general definition of reputation and trust.

3 Redefining Reputation

In improving upon the industry standard, we have identified three key contributions of a reinvented reputation system. A reputation system should:

1. Allow for realistic information sharing between agents
2. Represent the subjective opinions of community members
3. Incorporate a broader definition of reputation and trust

We first give our definition of trust and reputation in accordance with our system. An actor's **reputation** is *a prediction of future behavior or actions based on a memory of recorded actions and events.* **Trust** is defined as *the confidence an entity holds in the truthfulness of information.*

These definitions allow us to represent more rich relationships between actors within our world. For instance, consider an RPG game where the player has recently stolen from the local shopkeeper. This particular player has a history of illegal behavior, and has shown no mercy to those who threaten his laissez-faire lifestyle. In a single-value system, the shopkeeper will know only to dislike the player. He may choose to avoid or express his discontentment, getting himself killed in the process. In our system, the shopkeeper has a memory of the player's nefarious past, and he can use this information to make more educated decisions. He understands that the player is likely to kill him if confronted negatively, however the shopkeeper may also take advantage of the player's particular strengths to solve some other problems. While the shopkeeper may despise the player for stealing his wares, he may also have high trust in the player's ability to kill the bandits that have been harassing his family. The result is a personal bounty-hunting quest for the player. Our definitions of reputation and trust allow for a broader view that more accurately resembles real human reasoning. They allow us to capture complex social situations that were previously unrepresented in reputation systems.

An overview of our agent architecture design is shown in Figure 1. The components in green represent core reputation functionality and will be the focus for the remainder of this paper. The components in green act to predict a participating agent's next actions given a memory of previous behavior, and the best-guess actions are passed along to the planner to determine an appropriate response.

4 Information Representation

All reputation-based information in our game world is represented as an Resource Description Framework (RDF) [Tauberer 2006] triple and confidence value pair. We call this tuple a *percept.* The RDF triple component represents relationships between entities within the world; it can be viewed as a string containing a subject actor, relationship or action verb, and direct object. "The Player stole from The Shopkeeper" is a simple example. Here, 'the player' is the subject, 'stole from' is the predicate or relationship, and 'the shopkeeper' is the direct object. Attached to this RDF triple is a confidence value between 0 and 1 that represents how much trust the agent has in the truthfulness of this information.

- ("The Player stole from The Shopkeeper", .8)

An NPC gathers percepts from his environment through his perception system. Whenever an action is performed within the world, a corresponding percept is created to encode such information. A visual percept is 'seen' when it enters the unoccluded view frustum of an NPC within the world, and an audial percept is 'heard' when an NPC moves within some proximity of the information. Figure 2 displays a graphical representation of a visual and audial percept respectively.

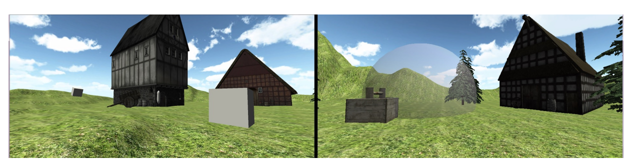

Figure 2: *Graphical Representation of Percepts - Left: Two boxes represent visible percepts - Right: A sphere represents an audible percept*

4.1 Retrieving Information

The trust value associated with each percept is a necessary condition in the case of gossip. Any raw percept, that is any percept directly created by the environment, is received with 100% confidence. NPCs have the ability to share percepts with one another,

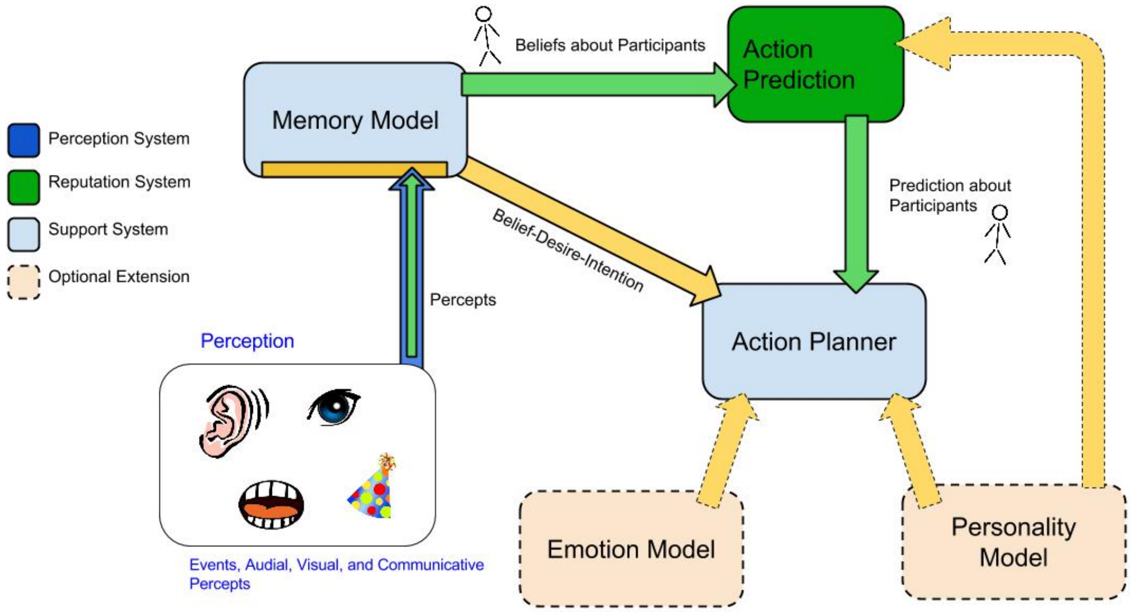

Figure 1: *NPC Architecture*

and when this occurs, a listening NPC reduces his trust in this information accordingly. As in Otello, this is used to prevent infinite dissemination of information, provides bias between agents, and provides an easy mechanism for resolving conflicting information; if one NPC has low trust in the truthfulness of another NPC, he may dramatically lower the trust of any received percepts from that source. As information passes from agent to agent, the truthfulness of an information steadily decreases until it is fundamentally 0 and is no longer relevant.

Using this scheme, we are able to represent communication between agents, a key component of our redesigned reputation system. When two agents wish to communicate, they share information via percepts in their environment. We have created a medieval town resembling Bethesda Softworks' *The Elder Scrolls Skyrim,* RPG. The characters in our world do not live with the luxury of the cell phone, so they must speak with one another directly to share their gossip. When two NPCs meet to talk about the town, they share percepts with one another by creating communication-audial percept zones, which are specially marked to distinguish from environmental percepts. This creates an important distinction from the single-value reputation systems of Skyrim or the social graph system of Otello because for an NPC to receive information about the player, he must witness it directly or overhear it from gossip within the town. If a witnessing agent is unable or chooses not to share his information with the members of the town, then the player's reputation will not be affected. Additionally, an NPC may choose with whom the information is shared, so one could anticipate cliques forming within the social space of the town distinguishing one group of trusted NPCs from another. This scheme allows our reputation system to model realistic information sharing between agents.

4.2 Understanding Information - The Ontology

In order for the NPCs to understand what actions are being performed within the world, we have designed an ontology that establishes relationships between different actions. The ontology categorizes actions and RDF predicates into groups, where each group shares some similarities. For instance, the action 'attack' is categorized under ('Action', 'Directed', 'Physical', 'Harmful.') The action 'kill' is categorized the same, while the action 'talk' is ('Action', 'Directed', 'Social.') This representation allows our NPCs to draw extended conclusions about percepts they receive and is described in further detail in section 5.1.3.

It is important to note that our action ontology is still a work in progress. In our current implementation, the ontology is defined by hand according to our intuitions. We recognize the need for further research into this area, particularly in drawing from natural language resources to better understand relationships between actions. While not the target of this paper, we look toward to future work in this area. A promising avenue we may explore includes automatic ontology generation [Alani et al. 2003].

4.3 Storing Information - Memory Models

An integral component to any human-like agent system is a realistic memory model. For our reputation system, the memory model plays a key role in forwarding relevant information to the prediction module as well as limiting the complexity of perceptual information. As an agent senses his environment, percepts are passed to the memory model before being processed by the action planner. This allows the memory system to truncate any irrelevant or repetitive percepts. Additionally, the memory system is responsible for efficiently identifying the known history of an actor and forwarding this information to the prediction module.

In our current implementation, we utilize a very simple memory that stores an agent's percepts as textual RDFs in a mapping structure. When stimuli is forwarded to the memory, we identify the key actors involved in the perceived event and index to retrieve relevant information. At the moment, our memory model serves only as an interface between the environment and the prediction module. In our future work, we hope to implement a more realistic memory system such as [Li et al. 2013; Kope et al. 2013] to more accurately simulate real humans.

5 The Prediction Module

Another important component of our reputation system is the prediction module. The prediction module is responsible for estimating another agent's future actions given a subjective history of past actions; we achieve this functionality using a Bayesian Network. Given an actor A's history, our NPCs learn a bayesian network that calculates the likelihood that actor A will perform some action. After relevant probabilities have been calculated, the module forwards a list of agent A's most probable actions to the planning module.

Data: Dictionary P - Probability of each action a in A, Integer n
Result: A list of n-maximum probability actions for some actor
Sort(P); //According to Descending Probabilities
for *(int i = 0, i ¡ n, i++)* **do**
  Add(List L, P[i]);
end
return List L;

Algorithm 1: Forwarding most probable actions

Our system utilizes a bayesian network for this method for a few specific reasons. Firstly, bayesian networks are capable of computing large quantities of independent probabilities efficiently. [Cozman et al. 2000]. Additionally, our problem space can be represented as a series of dependent random variables, and prior probabilities are initially zero; an agent has no prior knowledge of other actors. Lastly, the bayesian network's relatively simple approach to machine learning fits our application domain perfectly. The idea is straightforward and provides logical and consistent results for our agents. The high-level design of the prediction module is displayed in green in Figure 1.

5.1 Bayesian Network Design

When designing our bayesian network, we identified three key features that must be taken into account: The history of actions an actor A has performed, the environment in which actor A is present, and the set of actions that A has not yet, but could at some point act out. Each of these is discussed in greater detail below. In addition to considering these factors when evaluating agent A's likely next move, we have also included functionality for personality to drive the prediction process. If NPC B is predicting actor A's next move, B's personality may influence the decision he arrives at. For instance, if B is an optimistic person, he may believe A will act positively on his next move when a pessimistic individual would disagree. This functionality has not been implemented in our system, but we are excited for this avenue of future work.

5.1.1 Action History

An actor A's action history is the largest indicator of his/her next intentions. This information is passed to the bayesian network directly from NPC B's memory model, and it directly modifies the prior probabilities for each considered action. The probability that actor A performs action a is related to the number of times A has done a and the total number of actions A has completed. Equation 1 likely gives a more understandable definition. It should be noted that we apply an m-estimate to avoid margin errors.

$$P(a|Hist(A)) = \frac{\#a + 1}{\|Hist(A)\| + 2} \tag{1}$$

This definition alone is enough for a simple bayesian network and prediction module. However, we can construct more educated guesses with some additional information.

5.1.2 Acting Environment

The acting environment is a broad definition for the environment variables that may have an impact on an actor's decisions. This environment includes a vast number of possibilities such as actor A's current emotional state, the objects at A's current disposal, and if there are any other agents whom could skew A's decision making. The list continues and is an interesting area of future research, but for our purposes we simplify the acting environment solely to the direct object which is being acted upon. Referring back to our definition of the percept and its RDF triple, we note that every information unit may have an associated direct object. The intuition behind this is that an actor may act on certain objects in some specific way that is consistent. For example, consider the case where we observe an actor A in a room with a broomstick. Our memory of A might suggest he goes to sleep, because that is the action we most frequently observe him perform. However, we also observe the broomstick and remember that whenever we see A and a broomstick nearby, A has swept the floor. The idea of the acting environment has countless extensions and plays a significant role in the accuracy of our predictions. The modified equation is given below.

$$P(a|Hist(A), Env(A)) = \frac{\#a + 1}{\|Hist(A) \cup Env(A)\| + 2} \tag{2}$$

5.1.3 Action Ontology

Lastly, one can observe that actions are generally related to one another. For instance, 'attacking' and 'killing' share many similarities, and one may argue that a history of attacking may suggest a similarly violent future. To capture this intuition, we refer to the action ontology previously described. As all actions are categorized according to an explicit ontology, we can generalize the actions of a character to expect similar behavior. This idea proves a significant challenge however. How does one determine how influential these similarities should be? Will different NPC observers have dissenting opinions on how significant similar actions should be? For our implementation, we have concluded that this generalization should influence the decision, however it should not drastically change the outcome of the expected behavior. Our solution is to strictly limit the potency of this observation. The goal is to recognize that when an actor has a history of attacking, but has never committed murder, the probability of murder is likely non-zero. Our equation is listed below and can be applied after the posterior probabilities have been established. It is important to note that this equation can be applied in O(n) efficiency using dynamic programming algorithms.

$$\begin{aligned} &P(a|Similar(a,b)) = \\ &Max(\frac{1}{5} * P(b|Hist(A), Env(A)), P(a|Hist(A), Env(A))) \end{aligned} \tag{3}$$

In our future work, we will explore new ways of confronting this relationship. Particularly, we hope to focus on how actions relate to one another and how that influences human decision making.

5.2 Bayesian Network Structure

Given these considerations, the final task is constructing the physical structure of the bayesian network. Figure 3 shows the two-layer architecture of our bayes nets.

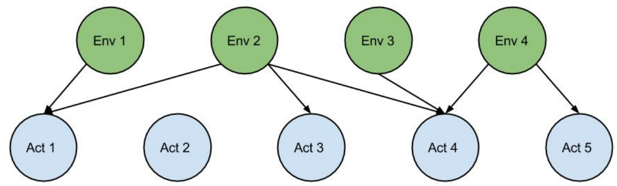

Figure 3: *A two-level prediction net*

The prediction network is implemented as a two-layer bayesian network, where the acting environment variables parent the actions an actor may take. We populate the prior probability tables according to the equations listed in section 5.1. From here, we can apply a variable elimination algorithm [2000] to efficiently compute the probability that actor A performs each action, with and without an acting environment.

When constructing the bayesian network, we also utilize a simple presence hueristic to significantly reduce computation costs. Each bayesian network begins with zero nodes, and is built up as our NPC learns about other agents in the world. When an NPC performs an action in a certain environment, we create the corresponding nodes in the bayesian network. If the nodes already exist, we update the probability tables to reflect the new information. In this manner, no unnecessary nodes will complicate the calculation. A significant challenge however is the efficient memory management of these bayesian networks. We recognize the complexity of this solution and provide a discussion in our concluding remarks.

6 Action Planning

The action planning module is responsible for determining the NPC's next action to perform given information about the world. For a traditional Belief-Desire-Intention system, this module analyzes the agent's current intentions to determine his next course of events, but things are a bit more complicated when our reputation system is included. In addition to analyzing the agent's belief, desires, and intentions, the planning module must also consider the predicted actions of his neighbors. This extra factor challenges the already-difficult problem of agent planning with a new consideration, and there may be situations when the next best action has conflicting interests. Consider again the shopkeeper and the nefarious player. The shopkeeper must decide if he should confront the thief or avoid a confrontation; while the shopkeeper is aware of the player's murderous reputation, he is also driven by his own desires and intentions to maintain profits in his shop to feed his family. These conundrums are difficult for real humans, and are now more evident when the reputation system is included.

Our solution to this problem remains a work in progress. There is significant room for improvement for more robust action planners on NPC agents, and our current research efforts are focused on developing a synergy between an agent's desires and the new form of knowledge that is reputation. While not an intended contribution of this paper, we expect to provide more details regarding the action planning module of our reputation system in the near future. Our path to solving this problem lies in action planning research, particularly on the complexities of the BDI model.

7 Conclusions and Future Work

Creating more believable NPCs is a significant challenge for game researches and designers alike. While there have been significant improvements across emotion, personality, and relationship models, reputation systems have not advanced as rapidly. In this paper, we have outlined some key benefits of a redesigned reputation model. We describe a system that:

1. Allows for realistic information sharing between agents
2. Represents the subjective opinions of community members
3. Incorporates a broader definition of reputation and trust

Our design provides a unique solution to the player-shopkeeper scenario. A player enters a new town in medieval times, so his reputation is unknown among the community. After a frenzy of attacks on a group of nearby merchants, the player quickly makes a name for himself. The player's 'attack' action creates visible percepts that some bystanders take note-of, and soon audial gossip percepts are popping up all over town. Importantly, because each NPC's knowledge is limited by their perception system, members of the town come to different subjective conclusions about the player. A few NPCs have not heard of the player's vicious attacks, many who have are fearful and flee, some are impressed by the players prowess and confront him with new job offers. The player's reputation is not constrained by a single global value, so more complex relationships occur. For instance, the shopkeeper has knowledge of the player's many attacks. He is able to predict that the player will likely attack again, and though he is fearful, the shopkeeper desires that his business competitor be 'taken care of'. He understands there is high-likelihood of the player attacking successfully, so he confronts the player with a bounty for the death of his neighbor. Of course, the shopkeeper and player must be careful to discuss privately. They wouldn't want word of their nefarious deal to spread throughout town.

In regard to the contributions in this paper, there remains significant room for improvement in tackling the problem of NPC believability. For our system, our foremost concern is integrating reputation into an agent's planning processes. We hope for our NPCs to determine their next actions based upon a combination of their desires as well as predicted behavior of their neighbors. Another avenue of future work is in developing an assessment of our reputation system. It is our ultimate goal to conduct user studies, where participants highlight an increase in immersion and higher believability for our NPC agents. However, we believe that the agent planning processes must first be finished before a user can adequately interact with our reputation system. Our reputation system also raises questions about performance for large scale games and applications, and while we have provided heuristics that significantly reduce computation costs, future work will require a metric overview of the system's complexity as the number of agents and actions increase. We believe our contributions in this paper highlight a relatively ignored solution area for the NPC believability problem. We describe a system that rethinks how NPCs gather and share reputation information and provides new decision making tools for the creation of believable Non-Player Characters.

References

ALANI, H., KIM, S., MILLARD, D., WEAL, M., HALL, W., LEWIS, P., AND SHADBOLT, N. 2003. Automatic ontology-based knowledge extraction from web documents. *Intelligent Systems, IEEE 18*, 1 (Jan), 14–21.

BETHESDASOFTWORKS, 2011. The elder scrolls v: Skyrim. [CD-ROM].

BLIZZARDENTERTAINMENT, 2004. World of warcraft. [CD-ROM].

COZMAN, F. G., ET AL. 2000. Generalizing variable elimination in bayesian networks. In *Workshop on Prob. Reasoning in Bayesian Networks at SBIA/Iberamia*, 21–26.

DIAS, J. A., AND PAIVA, A. 2013. I want to be your friend: Establishing relations with emotionally intelligent agents. In *Proceedings of the 2013 International Conference on Autonomous Agents and Multi-agent Systems*, International Foundation for Autonomous Agents and Multiagent Systems, Richland, SC, AAMAS '13, 777–784.

EGGES, A., KSHIRSAGAR, S., AND MAGNENAT-THALMANN, N. 2004. Generic personality and emotion simulation for conversational agents. *Computer Animation and Virtual Worlds 15*, 1, 1–13.

ENNIS, C., HOYET, L., EGGES, A., AND MCDONNELL, R. 2013. Emotion capture: Emotionally expressive characters for games. In *Proceedings of Motion on Games*, ACM, New York, NY, USA, MIG '13, 31:53–31:60.

HUYNH, T., JENNINGS, N., AND SHADBOLT, N. 2006. An integrated trust and reputation model for open multi-agent systems. *Autonomous Agents and Multi-Agent Systems 13*, 2, 119–154.

JENNETT, C., COX, A. L., CAIRNS, P., DHOPAREE, S., EPPS, A., TIJS, T., AND WALTON, A. 2008. Measuring and defining the experience of immersion in games. *Int. J. Hum.-Comput. Stud. 66*, 9 (Sept.), 641–661.

KOPE, A., ROSE, C., AND KATCHABAW, M. 2013. Modeling autobiographical memory for believable agents. In *AIIDE*.

LI, W., BALINT, T., AND ALLBECK, J. 2013. Using a parameterized memory model to modulate npc ai. In *Intelligent Virtual Agents*, R. Aylett, B. Krenn, C. Pelachaud, and H. Shimodaira, Eds., vol. 8108 of *Lecture Notes in Computer Science*. Springer Berlin Heidelberg, 1–14.

OCHS, M., SABOURET, N., AND CORRUBLE, V. 2009. Simulation of the dynamics of nonplayer characters' emotions and social relations in games. *Computational Intelligence and AI in Games, IEEE Transactions on 1*, 4 (Dec), 281–297.

PANAIT, L., AND LUKE, S. 2005. Cooperative multi-agent learning: The state of the art. *Autonomous Agents and Multi-Agent Systems 11*, 3 (Nov.), 387–434.

PINYOL, I., AND SABATER-MIR, J. 2013. Computational trust and reputation models for open multi-agent systems: a review. *Artificial Intelligence Review 40*, 1, 1–25.

RAO, A. S., GEORGEFF, M. P., ET AL. 1995. Bdi agents: From theory to practice. In *ICMAS*, vol. 95, 312–319.

RUMBELL, T., BARNDEN, J., DENHAM, S., AND WENNEKERS, T. 2012. Emotions in autonomous agents: Comparative analysis of mechanisms and functions. *Autonomous Agents and Multi-Agent Systems 25*, 1 (July), 1–45.

SELLERS, M. 2008. Otello: A next-generation reputation system for humans and npcs. In *Proceedings of Artificial Intelligence and Interactive Digital Entertainment*, AAAI, Menlo Park, California, USA, AIIDE '08, 149–154.

SQUAREENIX, 2013. Final fantasy xiv: A realm reborn. [CD-ROM].

TAUBERER, J., 2006. What is rdf? `http://www.xml.com/pub/a/2001/01/24/rdf.html`.

WOOLRIDGE, M. 2003. *Reasoning about Rational Agents*. The MIT Press.