Player Modeling in Civilization IV

Pieter Spronck and **Freek den Teuling**
Tilburg center for Creative Computing
Tilburg University, The Netherlands

Abstract

This research aims at building a preference-based player model of *Civilization IV* players. Our model incorporates attributes which are defined for AI players. We use a sequential minimal optimization (SMO) classifier to build the player model based on a training set with observations of a large number of games between six AI players. The model was validated on a test set of games between the same six AI players. While it did not seem to generalize well to the preferences of different AI players, it did manage to accurately predict some of the preferences for a veteran human player. Further tests showed that AI players with the same play styles but different preference values were often confused by the model. We conclude that for a complex game such as *Civilization IV* a model that attempts to accurately predict specific preference values is hard to construct. A model that focusses on play styles might succeed better.

Introduction

A player model is a representation of specific attributes of a player of a game. In practice, when player modeling is implemented in game artificial intelligence (AI), it is usually in the form of opponent modeling. The AI creates a model of its (human) opponent, to allow it to be more effective in defeating this human player. There are many games, however, where the goal of the game AI is not to defeat the human. Instead, the goal of such game AI might be to entertain the human player, or to assist him in acquiring certain skills. One could argue that to achieve such goals, it is highly important for the AI to gain an insight into the player's particular traits. Hence, it is beneficial for the AI to have access to an appropriate model of the player. As such a model does not treat the human (only) as an opponent, we prefer the term 'player model' over the term 'opponent model.'

In practice, player/opponent models usually are action models. I.e., they store the action that the human player is most likely to take in a given situation. An action model can be beneficial when the human player is treated as an opponent which must be defeated, as the AI can use it to predict the opponent's next move. However, for other goals an action model is less suitable. If the AI strives to entertain or train a human, a player model should encompass the human's preferences and skills, i.e., what the player desires to accomplish or experience in the game, and to what extent he is able to do that.

A player model that encompasses preferences and skills (which we will refer to as a 'preference model') has the potential to be much more useful than an action model, as it has explanatory power and is better able to generalize over actions. Still, in practice preference models are seldom used. The reason is that it is a much harder task to capture preferences in a model than it is to capture actions, as actions are clearly measurable, while preferences are not (to a lesser degree, the same holds for skills). In the present paper, we investigate to what extent it is possible to build a preference model of a player of the highly complex strategy game *Civilization IV*. We employ the preferences that are encoded into the AI players of *Civilization IV*, build models for several of them based on game observations, and use these models to predict preferences of different AI players and of human players. Our goal is to be able to predict accurately the preferences of a *Civilization IV* player by observing a game state in which this player is involved.

Background

Most research in game AI is directed at deterministic two-player board games with perfect information. The standard approach for such game AI is tree search.In 1993, two research groups started investigating the incorporation of explicit opponent models in tree search techniques. Carmel and Markovitch (1993) focussed on learning of opponent models, while Iida et al (1993) focussed on potential opponent model applications in tree search. In follow-up research, Uiterwijk and Van den Herik (1994) investigated search techniques that concentrated on fallibility of an opponent, and Donkers (2003) defined probabilistic opponent models that took into account uncertainty about the opponent's strategy.

While opponent modeling may be useful for deterministic, two-player games of perfect information, they are seldom employed in practice because the consequence of using a faulty opponent model may be that the AI plays significantly worse than it would without taking the opponent model into account (Donkers 2003). In imperfect-information games, however, the inclusion of an explicit opponent model is often a necessity for strong gameplay. See for example the work by Egnor (2000)on *Roshambo*, and the work by Billings (2006) on *Texas Hold'm Poker*.

In video games, opponent modeling is increasing in importance (Fürnkranz 2007). The main reason is that the purpose of AI in video games is usually 'entertaining the human player,' rather than 'defeating the human player' (van den Herik, Donkers, and Spronck 2005). Entertainment should not only be evoked, but also be maintained (van Lankveld et al. 2010). A game that succeeds in doing that for considerable time is appreciated more by players than a game that does not. Therefore maintaining entertainment in a game is beneficial for the developers' future business. Furthermore, in 'serious games' where training is the purpose of the game, a good player model may assist the game in achieving its goals in an efficient and effective manner.

Player modeling in video games is a challenging task for at least three reasons: (1) the environment is often highly complex, (2) there is little time for making observations, and (3) the environment is often only partially observable (Houlette 2004). It is rarely attempted in video games. When it is, it is usually in the form of action modeling (Donkers and Spronck 2006). Action models are usually built by simply counting actions or by pattern recognition methods, such as *N*-grams (Laramée 2002; Millington 2006). For most games action models are of limited use, as they can only be applied in the specific situations for which they stored actions. Preference models, in contrast, encompass preferences and skills of a player, such as predicting in which situation a player will behave aggressively. These models have explanatory power and the ability to generalize to novel situations (Donkers and Spronck 2006).

In recent years several implementations of preference models have been attempted. Bauckhage et al (2007) used a Markov Decision Process to model a player's strategies in *Quake II*. Rohs (2007) built rudimentary preference models for AI in the game *Civilization IV*. Yannakakis and Hallam (2007) investigated the modeling of game players with the specific purpose to increase player satisfaction. Thue et al (2008) researched how player models can be used to create stories that fit the player's interests. Sharma et al (2007) implemented a player model that represented a player's likes and dislikes for particular story elements in a game. Van Lankveld et al (2010) demonstrated how a single-valued player model can be used to maintain entertainment in a simple arcade game. Van Lankveld et al (2009) also did preliminary work on building a psychologically verified model of a player's extraversion by observing the player's behavior in a role-playing game. Finally, Bakkes (2010) investigated how player models can be automatically discovered to indicate high-level preferences for strategies in the RTS game *Spring*. None of this research has been applied in published games yet.

The work discussed in the present paper distinguishes itself from previous research by attempting to ramp up the complexity of the environment and the level of detail of the models. It also attempts to apply the models to predict the preferences of human players.

Experimental Setup

The goal of our experiments is to create a viable player preference model for *Civilization IV* players. We build this model based on a training set of observations of gameplay of *Civilization IV* AI players. We test the accuracy of the models we obtain against four test sets. The first test set consists of observations on the same AI players as used to build the training set. The second test set consists of observations on different AI players. The third and fourth test sets consist of observations on two human players, one casual player and one veteran player.

Civilization IV

Sid Meier's Civilization IV (from hereon: *Civilization IV*) is a highly complex turn-based strategy game in which the player controls a civilization. One or more rival civilizations populate the game world. The rivals are controlled either by AI, or by other human players. The players move in turn: on his turn, a player gets to command all his units and micromanage his societal, political, economical, military, and scientific structures. He can also engage in diplomatic contacts with his rivals, to make deals about trades and to come to political agreements. In its normal setup, the game lasts up to 460 turns. During this time, each of the civilizations attempts to grow and develop, outperforming rivals, to be the first to fulfill a victory condition. There are six different victory conditions, some focussing on military conquest, and others on scientific or cultural domination. In contrast with other strategy games, most of the turns of *Civilization IV* are spent in peaceful (though often tense) relationships with neighboring rivals. For the present research, we used version 1.61 of the base game of *Civilization IV*.

Player Model

In *Civilization IV* each civilization is represented by a leader. For AI players each possible leader has different preferences and a different playing style. For instance, Alexander the Great is an opportunistic, aggressive, military-oriented player, while Hatshepsut is a mildly-aggressive, peaceful, culturally-oriented leader. Because of the different ways leaders behave in the game, we decided to base our player models for *Civilization IV* on the behavior of these leaders.

The behavior of leaders is for a large part determined by their preferences.The developers of *Civilization IV* specified a list of parameters for the leaders which they call 'flavors.' These flavors are represented by a name and a value in the range $\{0, 2, 5, 10\}$. There are two possibilities for each leader: (1) either the values for all flavors are zero, except for one which has value 10, (2) or the values for all flavors are zero, except for one which has value 2, and one which has value 5. The higher the value for a flavor, the more a leader 'prefers' that flavor. Besides the flavors, several other preferences are specified for each leader. One of those is the preference for 'aggression,' which is represented by a value in the range $\{1, 2, 3, 4, 5\}$. The higher the value for 'aggression,' the more aggressive a leader will behave.

We decided to let our player model incorporate a player's preference for 'aggression,' and six of the flavors. In our

Table 1: Player model.

Preference	Range	Interpretation
Aggression	$1, 3, 4, 5$	Tendency to expand and declare war
Culture	$0, 5$	Tendency to invest in culture
Gold	$0, 2, 5$	Tendency to invest in economy
Growth	$0, 2$	Tendency to invest in food production
Military	$0, 2, 5$	Tendency to invest in military
Religion	$0, 2$	Tendency to invest in religion
Science	$0, 5$	Tendency to invest in research

experiments we focussed on a subset of the leaders. This lead to a player model consisting of seven parameters, each with a limited range of values (those which occurred in the training set). The complete player model is given in Table 1.

Observations

Our player models are built on observations of *Civilization IV* games. The games we used were duels between two players, on a 'tiny' map, in which both players started on the same continent. An observation is an abstraction of a *Civilization IV* game state. As *Civilization IV* is a highly complex game, for which the game state is only partially observable, such an abstraction will necessarily lose a great deal of information.

We decided to include in the observations all information which was easily accessible by the game AI, with two requirements: (1) it should be accessible by human players too, and (2) it should be accessible by a human player in a game against another human player. We settled on 25 basic features, including number of cities, number of units, population size, land size, wealth, score, war declarations, and values for economy, industry, agriculture, culture, and military power. The features are from the perspective of one of the players, and provide values for that player's civilization (e.g., 'cities' is the number of cities under the player's control). We extended the set of basic features with over 100 composite features, which are comparisons of features with corresponding features of the opponent (e.g., the difference in number of cities), or of features with the same features in earlier turns (e.g., the increase in number of cities). The composite features demonstrate differences, trends, and derivations. A full list is provided by Den Teuling (2010).

Training Set

The models were built on a training set of observations. To collect these observations, we let AIs, in the form of leaders, play *Civilization IV* in duels. These duels were run automatically, with both leaders controlled by an AI. This is not standard functionality in *Civilization IV*, but we could accomplish it by using a modification called *AiAutoPlay* (easily found by an internet search), which we adapted for our purposes. We used the 'Noble' difficulty setting, in which neither player has an advantage over the others.

We started with six leaders, which are listed, with their preferences, in Table 2. Each leader played against each other leader a total of 8 times, for a total of 40 games per leader. An observation is recorded at the end of each game turn. Each game provided between 240 and 460 observations. However, since the early turns of a *Civilization IV* game progress more or less the same, regardless of the leader, we decided to remove the observations for the first 100 turns of every game. This lead to a training set with a total of almost 55,000 observations, roughly 9,000 for each leader.

Table 2: Leaders in the training and Alexander test set.

Leader	Agg	Cul	Gold	Gro	Mil	Rel	Sci
Alexander	5	0	0	2	5	0	0
Hatshepsut	3	5	0	0	0	2	0
Louis XIV	3	5	0	0	2	0	0
Mansa Musa	1	0	5	0	0	2	0
Napoleon	4	0	2	0	5	0	0
Tokugawa	4	0	0	0	2	0	5

Table 3: Leaders used in the Cyrus test set.

Leader	Agg	Cul	Gold	Gro	Mil	Rel	Sci
Cyrus	4	0	0	2	5	0	0
Montezuma	5	0	2	0	5	0	0
Peter	4	0	0	2	0	0	5
Saladin	3	0	0	0	5	2	0
Victoria	3	0	5	2	0	0	0
Washington	3	0	0	2	5	0	0

Test Sets

We used four different test sets. For each of the games in the test sets, observations on the first 100 turns were removed, just as in the training set. The first test set is the *Alexander test set*, which contains observations on duels fought between the same six leaders that are incorporated in the training set. These were 10 new games per leader, for a total of 17,939 observations. The second is the *Cyrus test set*, which contains observations on duels fought between six different leaders. The six leaders included in the Cyrus test set are listed in Table 3. These leaders were chosen because their preference values met the same limitations we found in the training set. Ten games were played per leader, for a total of 16,330 observations in the Cyrus test set.

The third is the *casual test set*, which contains observations on two games of a human player who was a novice to the *Civilization IV* game. One game was played against Julius Caesar (who was neither in the training set, nor in the Cyrus test set), and one against Mansa Musa. This test set contained 656 observations. The fourth is the *veteran test set*, which contains observations on four games of a human player who was reasonably experienced with the *Civilization IV* game. He played games against Louis XIV, Tokugawa, Napoleon, and Mansa Musa. This test set contained 509 observations. Naturally, we do not have values for the preferences of the two human players. Therefore we asked them to comment their games, indicating their focus and preferences, and if they decided to change their focus in-game, list the new focus and the game turn when they changed. We interpreted their reports and assigned preference values accordingly. We deliberately asked the veteran player to stick to one focus for the game, just like an AI player would.

Modeling by Classification

Our preference model is a collection of seven classification models, one for each preference. To create these classification models, we applied the Sequential Minimal Optimization (SMO) classifier (Platt 1998) to the training set, using the Weka 3.6.1 environment (Witten and Frank 2005). We chose SMO, with the default Weka parameter settings, because in preliminary experiments we compared it to four other classifiers, and found it to be the most suitable for our purposes (den Teuling 2010).

Results

We performed three experiments. In the *validation* experiment, we created our player preference model and validated it on the Alexander test set. In the *generalization* experiment, we tested whether the model generalized to accurately predict the preferences of different AI leaders, by applying it to the Cyrus test set. In the *human players* experiment, we tested whether the model was able to accurately predict the preferences of two human players, by applying it to the casual and veteran test sets. The results of these experiments are discussed below.

Validation

We built a player model with the seven preference values by applying the SMO classifier to the training set for each of the preferences. The validation experiment tests the accuracy of the player model by applying it to the Alexander test set. To be considered a useful model, it should outperform at least a frequency baseline for the preferences. A frequency baseline is the percentage of the test set that would be predicted accurately, if the prediction was that all observations belonged to the preference that occurs most in the test set. For instance, the value 0 for the preference 'culture' occurs in 11828 of the 17939 observations in the Alexander test set, therefore the frequency baseline for 'culture' is 65.93%.

The results of the validation of the player model on the Alexander test set are graphically presented in Figure 1. From these results we can conclude that the accuracy of our player model is validated on the Alexander test set. For none of the preferences the model scores worse than the frequency baseline, and for the preferences 'aggression,' 'culture,' 'gold,' 'military,' and 'religion' it scores substantially higher. For the preferences 'growth' and 'science' improvements over the baseline are only marginal.

Generalization

To test to what extent our validated player model is able to generalize and accurately predict the preferences of other *Civilization IV* leaders, we applied the model to the Cyrus test set. A frequency baseline was calculated for all preferences in the Cyrus test set. A comparison of the frequency baselines and the predictions of the applied player model is graphically presented in Figure 2.

From these results we must conclude that the player model acquired from the training set does not generalize well. For all preferences, the player model's predictions are worse than the frequency baselines. These results can be labeled

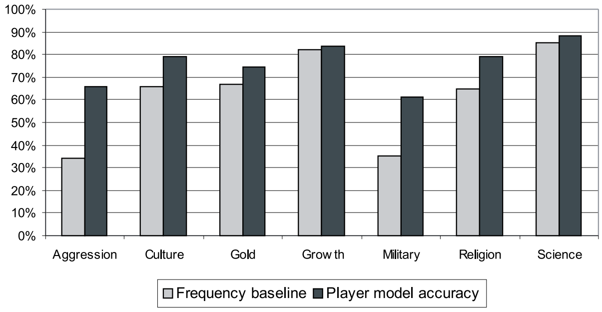

Figure 1: Results of the validation experiment.

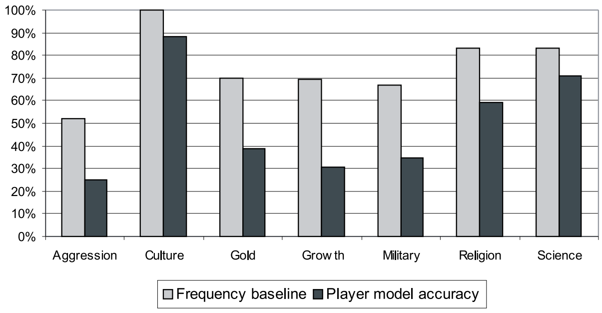

Figure 2: Results of the generalization experiment.

as 'somewhat disappointing.' Possible reasons for why the results are as they are, and potential avenues for improving them, are discussed below.

Human Players

To test to what extent our validated player model is able to accurately predict the preferences of human *Civilization IV* players, we applied the model to the casual test set and the veteran test set. For the casual test set, the accuracy of the player on all preferences scores lower than the frequency baseline. For the veteran test set, the player model scores substantially higher than the frequency baseline on the preferences 'culture' (92% by a frequency baseline of 51%) and 'gold' (81% by a frequency baseline of 56%), and performs about equal to the frequency baseline on the preferences 'growth,' 'military,' and 'religion.' Therefore, somewhat surprisingly considering the results on the Cyrus test set, the player model is actually reasonably successful in predicting the preferences of the veteran player. The results on the casual and veteran test sets are hard to interpret, as the preference values were assigned by us on the basis of the player reports. They are further discussed below.

Discussion

In this section we discuss the implications of the results achieved with the validation experiment, the generalization experiment, and the human players experiment.

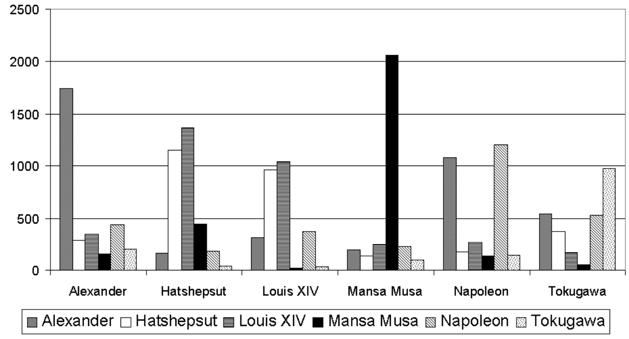

Figure 3: Indication of as whom each leader was classified.

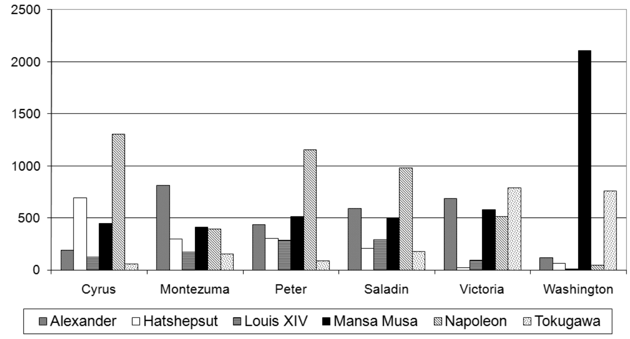

Figure 4: Test set leaders classified as training set leaders.

Discussion of the Validation Experiment

The results of the validation experiment showed that the preferences in the Alexander test set could be predicted substantially better than the frequency baselines. The predictions for 'growth' and 'science,' however, are not much better than the frequency baseline. One possible explanation for the lack of success on the preference 'growth' is that in our training set and test sets, the only values occurring for 'growth' are 0 and 2, i.e., 'no interest in growth' and 'a little bit of interest in growth' respectively. And as all civilizations must invest in growth to some extent, no clear indications for growth preference can be found in our observations. One possible explanation for the lack of success on the preference 'science' is that there was only one leader in our training set who is strongly interested in science, namely Tokugawa, who tends to be unsuccessful in scientific explorations as he refuses to make deals with other players. It should be noted that from a gameplay perspective Tokugawa *needs* a high value for the science preference, as without it he would fall way behind in scientific discoveries because of his isolationist nature.

Discussion of the Generalization Experiment

The results of the generalization experiment showed that the preferences in the Cyrus test set could not be predicted more accurately than the frequency baselines. In fact, in general the player model scored much lower than the frequency baselines. Our first impression was that the player model was overfitting the leaders which were included in the training set and Alexander test set. If that was the case, then we should be able to create a classification that predicts the actual leaders from the training set. To investigate this, we performed a new experiment in which we created, using SMO, a classification model that distinguishes the leaders. The results, which are presented in Figure 3, were quite illuminating. On the horizontal axis the figure displays the six leaders in the Alexander test set, and the bars indicate, for each leader, the number of observations for which they were classified as the leader represented by the bar.

In Figure 3 we see that Alexander and Mansa Musa are recognized well. We also see that Hatshepsut and Louis XIV are often confused with each other, while Napoleon is often identified as Alexander, and Tokugawa is often identified as Alexander or Napoleon. Alexander and Napoleon are both highly-aggressive, military-oriented leaders, while Hatshepsut and Louis XIV are both mildly aggressive, culturally-oriented leaders. Amongst these leaders Mansa Musa is in a league of his own, as the sole economically-oriented leader. Tokugawa seems hard to classify, but when he is misclassified it is mostly as an aggressive, military leader, which fits him well. Therefore we see that, while our approach seems to be unable to recognize all leaders, it seems to be particularly effective at recognizing leader play styles.

In Figure 4 we used the leader classification model on the Cyrus test set. We see that most of the leaders are hard to label. A strong exception is Washington, who is recognized as Mansa Musa. Both these leaders have a peaceful, economically driven play style. We also see that Cyrus, Peter, and Saladin seem to have similarities with Napoleon. Indeed, these are the three aggressive leaders of the Cyrus test set.

Play styles translate to preferences only to a limited extent. Tokugawa is a case in point: he has a very recognizable (isolationist) play style which the player model cannot represent, but which has a high impact on the observations. Play styles of the leaders in *Civilization IV* are mostly determined by special-purpose code. For the leaders we picked for our training set, interest in 'culture,' 'gold,' and 'military plus high aggression' are easy to recognize, and consequently the corresponding preferences can be predicted reasonably well. However, in the Cyrus test set, no cultural or clearly economical leaders can be found, and high military is combined with both high and low aggression. In short, the play styles of the leaders in the Cyrus test set do not match the training set. This seems to be the main reason that our model fails on the Cyrus test set. For future work, we therefore think that for such complex environments as found in *Civilization IV*, a player model should encompass play styles rather than specific parameter values. We did note, however, that it seemed easier to predict preferences for which high values were incorporated in the training set. Therefore, using our approach with more 'extreme' leaders might also lead to results that generalize better.

Discussion of the Human Players Experiment

The main problem with the experiment with human players, is that we needed to manually translate the written reports of the players to preference values. When we examined the results on the human player test sets for misclassifications of preferences with more than two possible values, we found that in 60-80% of the cases a misclassification concerned a 'one-off error,' i.e., a preference being classified as a neighbor value of the value that we distilled from the reports.

Conclusion

In this paper we discussed the building of preference-based player models for *Civilization IV*. We used a training set of observations on games played by six *Civilization IV* leaders, with different play styles, to construct a player model encompassing seven preferences. The model was validated on a test set containing games played by the same six leaders. However, we found that the model did not generalize well to different AI leaders. Further investigations showed that with our training set we can build a player model that recognizes play styles, but that the observation features in the training set seem to be insufficient to derive precise preference values. Experiments with human players show that for them the player model can predict some of the preference values well, if the human follows a focused play style.

The results provide indications for three directions of future research. First, we might achieve results that generalize better if we train on leaders with 'extreme' preference values. Second, modeling play style might generalize better than modeling preference values. Third, in a simpler environment with less variety of play, our approach might lead to better results. We intend to follow all three lines of research in future investigations.

References

Bakkes, S. 2010. *Rapid Adaptation of Video Game AI*. TiCC Ph.D. series 11. Tilburg, The Netherlands: Tilburg centre for Creative Computing, Tilburg University.

Bauckhage, C.; Gorman, B.; Thurau, C.; and Humphrys, M. 2007. Learning human behavior from analyzing activities in virtual environments. *MMI–Interaktiv* (12).

Billings, D. 2006. *Algorithms and Assessment in Computer Poker*. Ph.D. thesis. Edmonton, Alberta, Canada: Department of Computing Science, University of Alberta.

Carmel, D., and Markovitch, S. 1993. Learning models of opponent's strategy in game playing. In *Proceedings of AAAI Fall Symposium on Games: Planning and Learning*, 140–147.

den Teuling, F. 2010. *Preference-Based Player Modelling by Classification in Civilization IV*. M.A. thesis. Tilburg, The Netherlands: Tilburg University.

Donkers, H., and Spronck, P. 2006. Preference-based player modeling. In Rabin, S., ed., *AI Game Programming Wisdom 3*, 647–659. Hingham, MA: Charles River Media, Inc.

Donkers, H. 2003. *Nosce Hostem: Searching with Opponent Models*. Ph.D. thesis. Maastricht, The Netherlands: Universitaire Pers Maastricht.

Egnor, D. 2000. Iocaine powder. *ICGA Journal* 23(1):33–35.

Fürnkranz, J. 2007. Recent advances in machine learning and game playing. *ÖGAI-Journal* 26(6).

Houlette, R. 2004. Player modeling for adaptive games. In Rabin, S., ed., *AI Game Programming Wisdom 2*, 557–566. Hingham, MA: Charles River Media, Inc.

Iida, H.; Uiterwijk, J.; van den Herik, H.; and Herschberg, I. 1993. Potential applications of opponent-model search. part 1: the domain of applicability. *ICCA Journal* 16(4):201–208.

Laramée, F. 2002. Using N-gram statistical models to predict player behavior. In Rabin, S., ed., *AI Game Programming Wisdom*, 596–601. Hingham, MA: Charles River Media, Inc.

Millington, I. 2006. *Artificial Intelligence for Games*. San Francisco, CA: Morgan Kaufmann.

Platt, J. 1998. Fast training of support vector machines using sequential minimal optimization. In Schölkopf, B.; Burges, C.; and Smola, A., eds., *Advances in Kernel Methods Support Vector Machine*, 185–208. Cambridge, MA: MIT Press.

Rohs, M. 2007. *Preference-based Player Modelling for Civilization IV*. B.Sc. thesis. Maastricht, The Netherlands: Faculty of Humanities and Sciences, Maastricht University.

Sharma, M.; Ontañon, S.; Strong, C.; Mehta, M.; and Ram, A. 2007. Towards player preference modeling for drama management in interactive stories. In Wilson, D., and Sutcliffe, G., eds., *Proceedings of the Twentieth International FLAIRS Conference*, 571–576. AAAI Press.

Thue, D.; Bulitko, V.; and Spetch, M. 2008. Player modeling for interactive storytelling: A practical approach. In Rabin, S., ed., *AI Game Programming Wisdom 4*, 633–636. Hingham, MA: Charles River Media, Inc.

Uiterwijk, J., and van den Herik, H. 1994. Speculative play in computer chess. In van den Herik, H.; Herschberg, I.; and Uiterwijk, J., eds., *Advances in Computer Chess 7*, 79–90. Maastricht, The Netherlands: Maastricht University.

van den Herik, H.; Donkers, H.; and Spronck, P. 2005. Opponent modelling and commercial games. In Kendall, G., and Lucas, S., eds., *Proceedings of the IEEE 2005 Symposium on Computational Intelligence and Games (CIG'05)*, 15–25.

van Lankveld, G.; Spronck, P.; Rauterberg, M.; and van den Herik, H. 2010. Incongruity-based adaptive game balancing. In van den Herik, H., and Spronck, P., eds., *Proceedings of the 12th Advances in Computer Games Conference*, 208–219. Berlin, Germany: Springer-Verlag.

van Lankveld, G.; Schreurs, S.; and Spronck, P. 2009. Psychologically verified player modelling. In Breitlauch, L., ed., *10th International Conference on Intelligent Games and Simulation GAME-ON 2009*, 12–19. Gent, Belgium: EUROSIS.

Witten, I., and Frank, E. 2005. *Data Mining: Practical Machine Learning Tools and Techniques (Second Edition)*. San Francisco, CA: Morgan Kaufmann.

Yannakakis, G., and Hallam, J. 2007. Towards optimizing entertainment in computer games. *Applied Artificial Intelligence* 21(10):933–971.