Opponent Modelling In World Of Warcraft

A.J.J. Valkenberg

19th June 2007

Abstract

In tactical commercial games, knowledge of an opponent's location is advantageous when designing a tactic. This paper proposes using a modified Reliable Adaptive Game Intelligence model to predict the position of opponents in the Massive Multiplayer Online game World of Warcraft. This model uses a case base to create explicit opponent models. We discuss a method to convert the imperfect information that is generated in these games to perfect information. Experiments show that in a simulated environment the opponent models that make use of this conversion can represent the positioning of the opponents. We conclude that this model and the information-conversion method can successfully be used to predict the location of opponents in commercial games.

1 Introduction

In the last decade, the commercial gaming industry has grown to become the major industry it is today. Among the different games published, one popular game-type is the Massive Multiplayer Online (MMO) game. In these games thousands of users simultaneously play together. A specific genre among these games is the MMO Role Playing Game (MMORPG), where humans play a specific character with special abilities and powers. A human player can interact with all the other players in the area and vice-versa, as well as with many Non-Playable Characters (NPCs).

Among the most popular of MMORPGs is the game World of Warcraft (WoW) [2]. The number of subscribers to this game, which the developer Blizzard recently reported to be over 8.5 million worldwide [1], gives a good indication of the scale of this game and the industry. The setting of WoW is a fantasy world, based on the Warcraft games-series, where a player can choose to make a character that has its allegiance to one of two factions. These factions are at war with each other and thus provide a basis for the Player-versus-Player (PvP) aspect of the game. PvP means that two or more human-controlled characters fight with each other. This PvP action can occur throughout the game-world, but there are also special places where mini-games can be played, called BattleGrounds (BGs). At the moment WoW contains four different BGs. These BGs are objective-oriented games, in which one team consisting of players from one faction competes against another team from the other faction.

The focus of this paper will be on the Eye of the Storm battleground, hereafter referred to as 'the Eye'. In this battleground the goal of each team is to defeat the other team by having a better strategy then they do. Knowledge the positions of the opponents can be advantageous, since it indicates weak spots in their strategy. To acquire this knowledge we use techniques from adaptive game AI, a case base, and opponent modelling, and with them attempt to predict the opposing players' locations. The problem statement will therefore be: *To what extent can a case base and opponent modelling be used to predict the position of human opponents in 'the Eye of the Storm', a resource gathering PvP game within* World of Warcraft

The outline of this paper is as follows. In Section 2 related research and the rules and details of the Eye of the Storm are presented. Section 3 gives an overview of the experimental set-up involving the modified Reliable Adaptive Game Intelligence (RAGI) model. Section 4 reflects on the results obtained by the experiments. In Section 5 we discuss whether the simplified RAGI model is suitable for predicting player locations, and conclude the paper.

2 Background

This section explains the rules of the Eye and discusses the related research that this paper uses as a foundation.

2.1 Game setting

The Eye of the Storm BG is a game that is played in a constrained game world, illustrated in Figure 1. Two teams of maximally 15 players fight against each other. The two teams start opposite from each other, represented by an A for the alliance and an H for the horde in Figure 1. The winning objective for each team is to gather 'resources'. These resources can be gained from

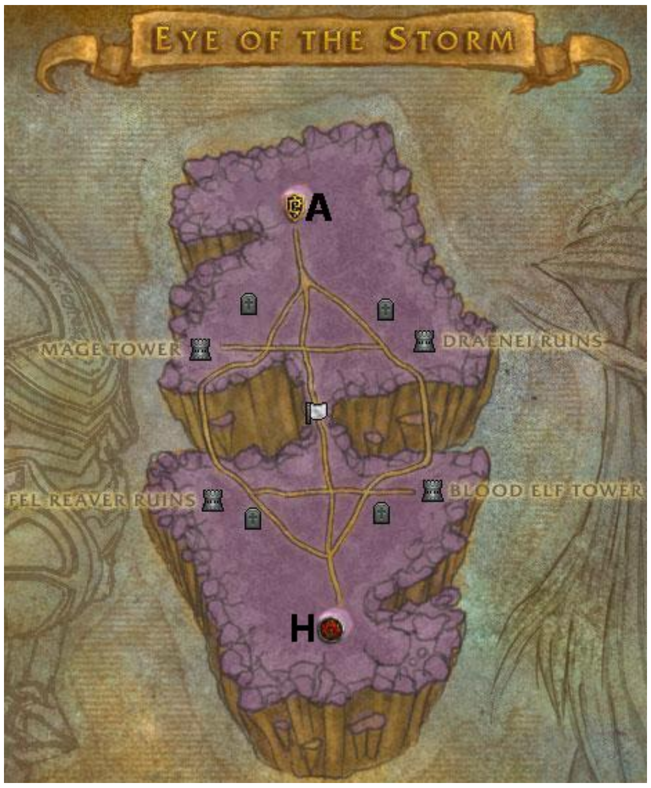

Figure 1: The Eye of the Storm BattleGround.

controlling any of the four towers in the Eye, or by capturing the flag. The towers must be claimed by one of the two teams in order to start producing resources for that team. The resource production of a tower for one of the teams increases when more towers are controlled by that team. When a tower is claimed by one of the teams, the other team can assault it and try to claim it for their own. To claim a tower a team must be present at the tower with more players than the other team. After a while, the tower will either go from a claimed state to a neutral state or from a neutral state to a claimed state. The flag works as a bonus item, which can be claimed and then moved to one of the towers that the claiming team controls to gain extra resources. After capture, the flag will automatically re-appear in its starting location. Since the flag has the same states as the towers, we will treat it as a tower for the remainder of this paper. When a player is killed in combat, he[1] will come back into the game at one of the four graveyards, represented in Figure 1 as gravestones. It is clear that in order to win this mini-game, a team needs to control at least two towers and capture the flag more times than the opposite team.

The challenge is that the strategy of the opposing team is unknown at the start of the game. As in most games players in a team are brought together more or less randomly, each player just acts out of his own interests and there is no joint-strategy. When a player knows where his opponents are located it is clear that an advantage can be gained. Knowing the location of the opposing players means that the player can know which tower is undefended or defended by the least number of opposing players. However, the position of the opponents can only be observed by direct contact with them. In other words, not all events are observable all the time, which makes this a game with imperfect information.

[1]Throughout this paper we mean to refer to 'he/she' and 'him/her' when saying 'he' and 'him' respectively.

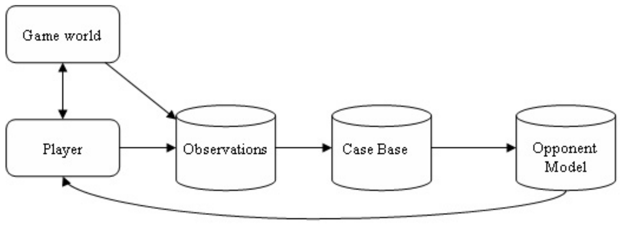

Figure 2: The simplified RAGI model.

2.2 Related Research

Spronck [4] states that the RAGI model can be used for the implementation of reliable adaptive game AI meeting eight requirements: speed, effectiveness, robustness, efficiency, clarity, variety, consistency, and scalability. This paper proposes using a simplified version of the RAGI model to predict the positioning of a human opponent in the Eye.

The simplified RAGI model (see Figure 2) implements a feedback loop collecting observations from the game world and the player (in contrast to an agent in the RAGI model). From these observations a case base is constructed similar to one described by Spronck [4]. From the case base an explicit opponent model is built. Spronck [4] defines an explicit opponent model as: "An opponent model is explicit in game AI when a specification of the opponent's attributes exists separately from the decision-making process." Because this paper focuses on location prediction of human players the term 'game AI' in the definition is interchangeable with 'location prediction'. Carmel and Markovitch [3] have shown that such an explicit model gives more effective AI than an implicit one.

3 Experimental set-up

This section explains the simplified RAGI model in detail. Section 3.1 discusses the observations that can be made in the Eye. Section 3.2 describes how the case base is built. Section 3.3 shows how an Opponent Model can be constructed from the case base and Section 3.4 discusses the set-up for the experiments and tests.

3.1 Observations

In order to predict the position of the opposing team we need to predict where each individual opponent is. Since it is not possible to query the game server for the coordinates of the enemy players or observe events taking place some distance away from the player, only the location of opponents within the field of view of the player can be known.

In order to record and parse information on opponent locations, we developed a special client-side program, called add-on, for WoW. This add-on can receive certain events from the game server. An event-system that is ideal to generate the needed observations is the combat system in WoW. This system generates all kinds of different messages while combat is going on between players. Another system of events that was used is the system that the BG uses to broadcast game-state changes to the players. These game-state changes include the state changes of the towers, like a tower being claimed by one of the teams, which might influence the position of the opponents.

Using these systems, we can determine which opponent the player running the add-on is in combat with. Because we can also use this add-on to track the zone that the player is currently in, we know in which zone encountered opponents were and in what state the tower within that zone was. In this way the player can move to a zone and see which opponents are currently present there. The name of the opponent and the current game-state form the observations that a player with our add-on can make in the Eye.

3.2 Case base

Since we want to predict the location of opponents, we need to have data on each opposing player in the BG. Because a player can only observe events up to a certain (short) distance away, we cannot collect data on every opponent at the same time. We are limited to generating data on the opponents that we meet in the current zone, thus creating imperfect data. However, this imperfect data can be converted to perfect data. We can not only process which opponents were present in the zone, but we can also process which opponents were *not* present. This way we can generate data on all the opponents in the BG, thus creating perfect data.

The case base consists of a table with information for each encountered opponent. A single entry will look like Table 1. These tables contain values that represent the time (in seconds) that the opponent was seen and not seen at a certain tower in a certain state. The values in the case base are defined as follows.

$$t_{soi} \begin{cases} s = \{n, a, h\} \\ o = \{p, \bar{p}\} \\ i = \{1, 2, 3, 4, 5\} \end{cases} \tag{1}$$

In Equation (1), s denotes the state of the tower which can be either n (neutral), a (alliance claimed) or h (horde claimed). o denotes what type of observation was made: p (opponent present) or $\bar{p}$ (opponent not present). i identifies the tower number: 1 (Draenei Ruins), 2 (Blood Elf Tower), 3 (Fel Reaver Ruins), 4 (Mage Tower), and 5 (Flag area).

Combining the case base with the current state of the BG, an Opponent Model can be made for each opponent, which will be discussed next.

3.3 Opponent Model

To be able to predict where an opponent is, we will need to make an Opponent Model (OM). From this OM it should become clear how the location of the opponent, given the current state of the BG, is distributed over the different towers. The OM represents the chance per tower that the opponent is located at that tower. With this information from all the opponents in the current BG, an overview can be constructed of which opponents are likely to be at each tower.

To convert the data in the case base to an OM we need to define some variables.

$$S_p = \sum_{i=1}^{K} t_{spi} \tag{2}$$

$$S_{\bar{p}} = \sum_{i=1}^{K} t_{s\bar{p}i} \tag{3}$$

$$Fr(T_j) = \frac{\left(\frac{t_{spj}}{S_p}\right)}{\left(\frac{t_{spj}+t_{s\bar{p}j}}{S_p+S_{\bar{p}}}\right)} \tag{4}$$

$$P(T_j) = \frac{Fr(T_j)}{\sum_{i=1}^{K} Fr(T_i)} \tag{5}$$

Equation (2) defines the total amount of time that the opponent has been observed as being present, taking into account the current states of the towers. In this equation, s specifies the state of tower i and K specifies the total number of towers, 5 in this case. Equation (3) defines the amount of time that the opponent has been observed as not being present in the same way as Equation (2). Equation (4) defines which fraction of the total observed time the opponent was present at tower j. When this fraction is known for each tower, we can calculate what percentage of time the opponent is positioned at each tower, as in Equation (5). The OM consists of a percentage of chance for each tower that the opponent will be at that tower, taking into account the current state of that tower. Note that the percentages change as soon as the state of at least one of the towers changes, i.e., the OM accurately represents the changing behaviour of a player with respect to changing tower

Tower State	T_1		T_2		T_3		T_4		T_5	
	p	$\bar{p}$	p	$\bar{p}$	p	$\bar{p}$	p	$\bar{p}$	p	$\bar{p}$
neutral	t_{np1}	$t_{n\bar{p}1}$	t_{np2}	$t_{n\bar{p}2}$	t_{np3}	$t_{n\bar{p}3}$	t_{np4}	$t_{n\bar{p}4}$	t_{np5}	$t_{n\bar{p}5}$
alliance	t_{ap1}	$t_{a\bar{p}1}$	t_{ap2}	$t_{a\bar{p}2}$	t_{ap3}	$t_{a\bar{p}3}$	t_{ap4}	$t_{a\bar{p}4}$	t_{ap5}	$t_{a\bar{p}5}$
horde	t_{hp1}	$t_{h\bar{p}1}$	t_{hp2}	$t_{h\bar{p}2}$	t_{hp3}	$t_{h\bar{p}3}$	t_{hp4}	$t_{h\bar{p}4}$	t_{hp5}	$t_{h\bar{p}5}$

Table 1: Example of a case base entry for a single opponent.

states. Next, we discuss how acquiring perfect information from imperfect information using a case base and OM can be tested in a small simulated environment.

3.4 Simulation setup

The goal of this simulation is to convert the imperfect information to perfect information using the case base and OM. To do that, a program was developed in Java. The simulation consists of four towers and five opponents, with each opponent having a different strategy. These strategies represent the chance that the opponent will move to another tower. Note that in the simulation the towers do not have different states. This was not implemented because the states are not relevant to testing if the imperfect information can be converted to perfect information. The opponents in the simulation may move to a new tower every simulated 30 seconds. If an opponent chooses the tower that he is currently at as the tower that he wants to move to, he just remains where he is. The five strategies in the simulation are as follows:

- Defensive. This strategy simulates a defensive player. It has an equal chance to stay at the closest two towers from the starting position, Tower 1 and Tower 2 in the simulation. It will never move to Tower 3 or Tower 4.
- Aggressive. This strategy simulates an aggressive player. It has an equal chance to go to the two towers that are furthest away from the starting position, Tower 3 and Tower 4 in the simulation. It will never move to Tower 1 or Tower 2.
- Stayer. This strategy simulates a player that prefers one tower over the others, but will assist somewhere else if needed. It has a 70 percent chance to stay at Tower 4 and it has a 10 percent chance to move to Tower 1, Tower 2, or Tower 3.
- Random. This strategy simulates a player that just randomly moves from tower to tower, with no tower being visited more often than another. It has a 25 percent chance to visit any of the four towers.
- Clockwise. This strategy simulates a player that always makes the same movements. The distribution of time over the four towers is equal to the Random strategy; 25 percent for each tower. However, this strategy visits the towers in a pre-determined order. It starts at Tower 1 and then moves clockwise to Tower 2, Tower 3, and Tower 4.

See Table 2 for an overview of the different strategies.

Tower Opponent	T_1	T_2	T_3	T_4
Defensive	50%	50%	0%	0%
Aggressive	0%	0%	50%	50%
Stayer	10%	10%	10%	70%
Random	25%	25%	25%	25%
Clockwise	25%	25%	25%	25%

Table 2: An overview of opponent strategies in the simulation.

The simulation will run over a specified number of simulated minutes. To test our way of converting the imperfect information we receive through the add-on program to perfect information, as described in Section 3.2, the simulation will run in two different modes. One mode will be generating perfect information; each tower processes which opponents are present and which are not present. The other mode will use a player, or observer, to move to different towers on a random time interval of 20 to 60 seconds. The observer will only process the opponents that he has encountered at his tower. This last mode represents the actual situation in WoW. Results from tests in both modes are presented in the next section.

4 Results

This section contains the results for the experiment described in section 3.

4.1 Simulation Results

Figure 3 shows the results from the simulation run in perfect-information mode. These results represent the average accuracy of the OM that can be constructed for each opponent with the information from the case base. The case base that was created with this test can be

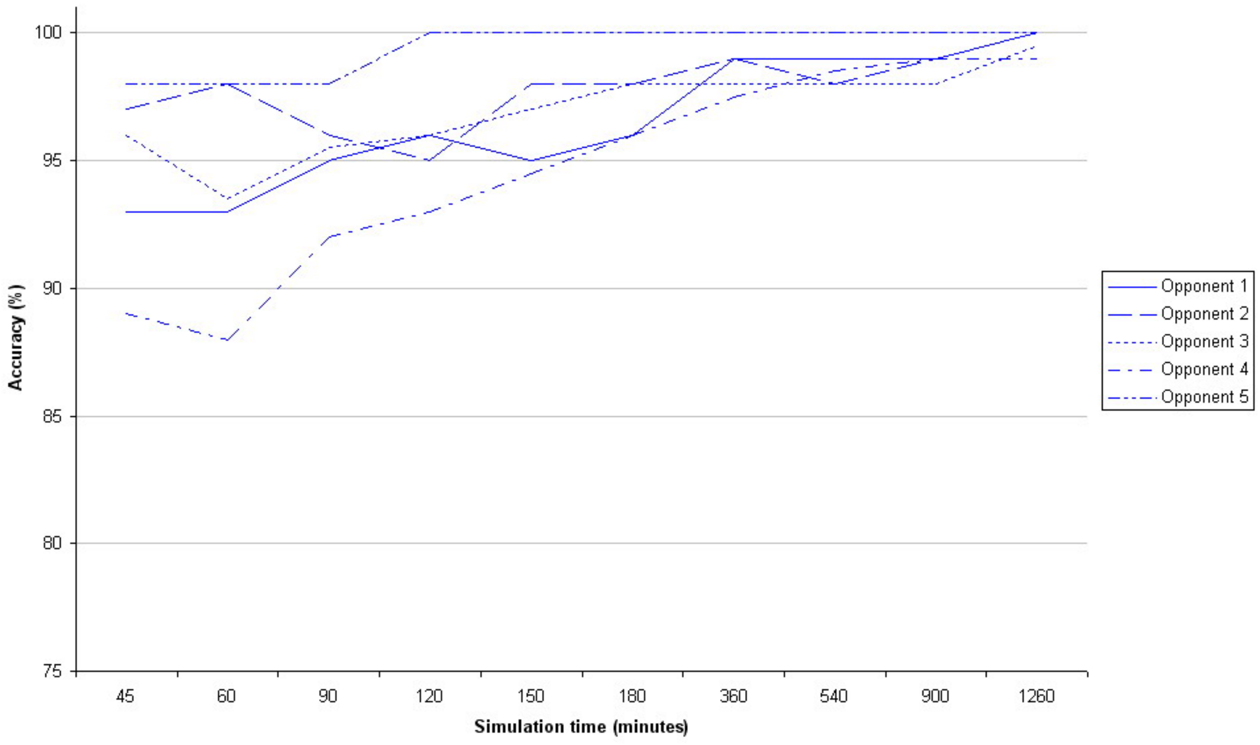

Figure 3: The Accuracy of the perfect-information results.

found in Table 3 (only the case base for the 1260-minutes test is included). We define the error and accuracy that will be used as follows.

$$E(T_i) = Result(T_i) - P(T_i) \tag{6}$$

$$Accuracy = 100 - \frac{\sum_{i=1}^{K} E(T_i)}{K} \tag{7}$$

In (6) the error per opponent per tower is calculated, where $Result(T_i)$ stands for the OM results from the simulation and $P(T_i)$ is the strategy percentage from the simulation (see Table 2). The Accuracy of a simulation per opponent is then defined in (7) as 100 percent minus the sum of the error per tower divided by the total amount of towers.

Note that the total observing time in perfect-information mode will be four times higher than the simulation run time. This is because there are four towers that process information during the run, so each tower processes one whole run time.

In Figure 4 the results from the simulation in observer mode are presented. These are the results from the test that used the imperfect information from the observer and converted it into perfect information using the techniques described in Sections 3.2 and 3.3. Table 4 shows the case base that was created during the simulation. From these tables it becomes clear that even with imperfect information, the strategies of the opponents can be correctly predicted. We can see that after 90 minutes the accuracy for four of the five opponents exceeds 90 percent while the accuracy for the fifth opponent is nearly 80 percent. At this point, the OMs will be accurate enough to start making predictions on the opponents locations. The results for opponent 3 may not seem that good, but it is clear that this opponent favours one tower over the others and that is sufficient to make a prediction. From these OMs we can conclude that, for instance, tower four will likely have two opposing players near it and tower one will likely only have one opponent defending it.

4.2 Practical Results

The add-on program described in Section 3.1 was successfully implemented and tested in the Eye. The results from the experiments were inconclusive due to insufficient data. Because of the amount of players that are present, it takes time to observe opponents more than once. The average game length for an Eye of the Storm game is 15 minutes. In the simulation we noticed that after about 90 minutes the accuracy of the OMs was enough to start making predictions. If these results re-

Tower / Opponent	T_1		T_2		T_3		T_4	
	p	$\bar{p}$	p	$\bar{p}$	p	$\bar{p}$	p	$\bar{p}$
Opponent 1	37950	37650	37650	37950	0	75600	0	75600
Opponent 2	0	75600	0	75600	37710	37890	37890	37710
Opponent 3	7950	67650	7290	68310	7260	68340	53100	22500
Opponent 4	17910	57690	18630	56970	19950	55650	19110	56490
Opponent 5	18930	56670	18930	56670	18870	56730	18870	56730

Table 3: The case base after 1260 minutes of simulating with perfect-information processing.

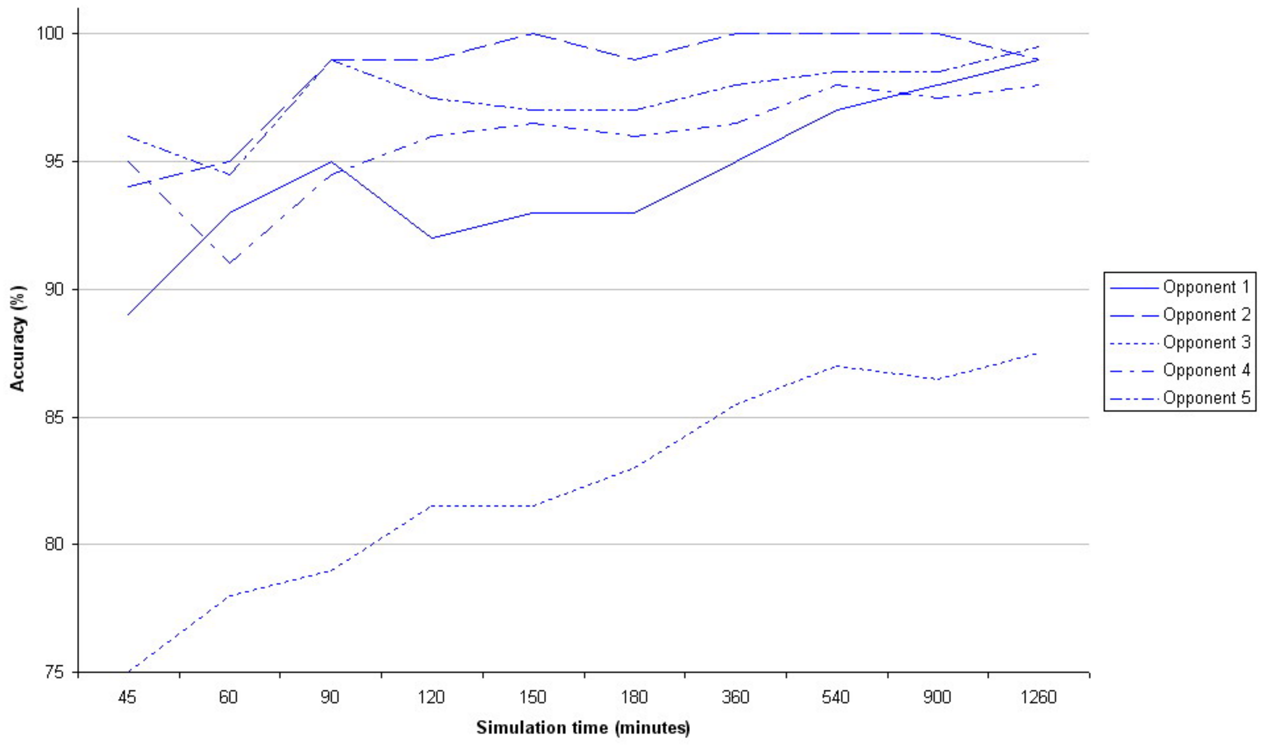

Figure 4: The Accuracy of the imperfect-information results.

main for 'real' WoW, then that means we need to meet an opponent in about 6 games in order to start making reliable predictions. However, we expect that more games are required since human players will be less predictable than the opponents in the simulation. During the practical testing, which were done in the course of 2 weeks, we encountered 509 different opponents. The average observing time for these opponents was about 15 minutes, which is 1 game. Table 5 shows a case base for an opponent that was constructed through the practical tests. The total observed time for this opponent is the second-highest in the data, yet it is 'only' 36 minutes. The opponent with the highest total observed time showed a very elusive case base; he was nearly never observed as being present, so it would be hard to make a statement about him. But when we look at Table 5, we can cautiously classify this opponent as a defensive opponent since he has been observed as present more when a tower was taken by the Horde (his own team). He also was observed more as not being present when a tower was taken by the Alliance (his opposing team). With these results and the simulation results, we can confidently say that when enough data can be gained on opponents, successful location predictions are possible. This will be continued in future work.

5 Discussion

The goal of the research presented in this paper was to find out to what extent a case base and an opponent model could be used to predict player locations in the Eye of the Storm Battleground in the game World of

Tower / Opponent	T_1		T_2		T_3		T_4	
	p	$\bar{p}$	p	$\bar{p}$	p	$\bar{p}$	p	$\bar{p}$
Opponent 1	14351	4416	14517	3570	0	18825	0	19921
Opponent 2	0	18767	0	18087	15141	3684	16515	3406
Opponent 3	4467	14300	4248	13839	4506	14319	18826	1095
Opponent 4	9186	9581	9056	9031	9086	9739	10597	9324
Opponent 5	11115	7652	10173	7914	11142	7683	11851	8070

Table 4: The case base after 1260 minutes of simulating with imperfect-information processing.

Tower / State	T_1		T_2		T_3		T_4		T_5	
	p	$\bar{p}$	p	$\bar{p}$	p	$\bar{p}$	p	$\bar{p}$	p	$\bar{p}$
neutral	0	0	0	12	0	197	0	0	131	150
alliance	127	578	0	135	0	20	0	111	58	48
horde	0	0	80	53	50	183	92	0	33	102

Table 5: The case base for an opponent from the practical tests.

Warcraft. Unfortunately, the case base from the live tests was not sufficient enough to yield conclusive results. We see four reasons for this. The first reason is that because of the large player base that makes up the Battleground system, a lot of games need to be played to encounter an opponent more than one time. A solution for this is to have more than one player in the team run the add-on during play and synchronize their observations. This will provide more observations and thus improve the overall opponent models and prediction. The second reason is that the observations in the case base are never removed. This may lead to sub-optimal results because players tend to adjust their strategies and play styles over time. A solution for this is to adjust the system so that each observation has a time-stamp and is weighed according to that time-stamp when making up the opponent model, or even removed from the case base. The third reason is the balance between exploration and exploitation. At the moment, to be able to generate data that is useful, a player needs to do a lot of exploration before he can use the gathered data to his advantage. By this we mean that the player might not be able to play the game the way he likes to, because that would not generate enough data. This problem can be solved by having more players collecting data, thus allowing them to play the game in a normal way. The fourth reason is the fact that no distinction is made between observations made early in the game and near the end of the game. This could also have an influence on the positions of players, such as desperate attempts to claim extra towers when a team faces defeat in the last stretches of the game.

6 Conclusion

This paper proposes using a modified Reliable Adaptive Game Intelligence model [4] to predict the position of opponents in a battleground within a Massive Multiplayer Online game. We use a case base to store observations made in the game world by a special add-on program and explicit opponent models to represent the strategy of the opponent. The observations are converted from imperfect information to perfect information on all the opponents in the battleground. In a simulated environment the conversion method works; it produces opponent models that are at least 79 percent accurate for the tested types of players when an opponent is observed for 90 minutes, and it produces opponent models that are at least 87.5 percent accurate for the tested types of players when an opponent is observed for 1260 minutes. However, observing an opponent for 90 minutes in the WORLD OF WARCRAFT BattleGround-system in practice is challenging because a game lasts for about 15 minutes and it is difficulty to encounter a specific opponent more than once. This is the reason that there are not yet any significant practical results; there simply is not enough data on each encountered opponent to make a good opponent model. Future work should include synchronizing the observations made my multiple players in the same team using the add-on. This will significantly decrease the amount of games that need to be played before a reliable opponent model can be made and players will not have to worry about having to explore too much if they do not want to. Since the use of such a tool will provide players with a better chance to win the game, we expect that we will not have many problems finding players willing to assist us in these experiments.

Acknowledgements

We would like to thank J.P.M. van Berlo for his unpublished work which formed a basis for this paper (Section 2.2 in particular).

References

[1] Blizzard Entertainment (2007a). Press release. `http://www.blizzard.com/press/070307.shtml`.

[2] Blizzard Entertainment (2007b). World of warcraft. `http://www.worldofwarcraft.com`.

[3] Carmel, D. and Markovitch, S. (1997). Exploration and adaptation in multiagent systems: A model-based approach. *Proceedings of the Fifteenth International Joint Conference on Artificial Intelligence*, pp. 606–611, Morgan Kaufmann, San Francisco, CA.

[4] Spronck, Pieter (2005). A model for reliable adaptive game intelligence. *IJCAI-05 Workshop on Reasoning, Representation, and Learning in Computer Games*, pp. 95–100, Naval Research Laboratory, Navy Center for Applied Research in Artificial Intelligence, Washington, DC.