Entertainment Computing 3 (2012) 71–79

Contents lists available at SciVerse ScienceDirect

Entertainment Computing

journal homepage: ees.elsevier.com/entcom

Player behavioural modelling for video games

Sander C.J. Bakkes [a,*], Pieter H.M. Spronck [b], Giel van Lankveld [b]

[a] Create-IT Applied Research, Amsterdam University of Applied Sciences (HvA), P.O. Box 1025, 1000 BA Amsterdam, The Netherlands
[b] Tilburg Centre for Cognition and Communication (TiCC), Tilburg University, P.O. Box 90153, 5000 LE Tilburg, The Netherlands

ARTICLE INFO

Article history:
Available online 6 January 2012

Keywords:
Video game AI
Player behavioural modelling
Adaptive behaviour

ABSTRACT

Player behavioural modelling has grown from a means to improve the playing strength of computer programs that play classic games (e.g., chess), to a means for impacting the player experience and satisfaction in video games, as well as in cross-domain applications such as interactive storytelling. In this context, player behavioural modelling is concerned with two goals, namely (1) providing an interesting or effective game AI on the basis of player models and (2) creating a basis for game developers to personalise gameplay as a whole, and creating new user-driven game mechanics. In this article, we provide an overview of player behavioural modelling for video games by detailing four distinct approaches, namely (1) modelling player actions, (2) modelling player tactics, (3) modelling player strategies, and (4) player profiling. We conclude the article with an analysis on the applicability of the approaches for the domain of video games.

1. Introduction

Player behavioural modelling is a research area in game playing that is gaining attention from both game researchers and game developers. It concerns generating models of player behaviour and exploiting the models in actual play. Considering the increasing complexity of state-of-the-art video games [1,2], player models are sorely needed for determining accurately, and adapting, the player experience. In general, a player model is an abstracted description of a player in a game environment. Specifically for the context of behavioural modelling, a player model is an abstracted description of a player's behaviour in a game environment. In general it concerns only the behaviour of human players, but player modelling techniques can be applied to the behaviour of computer-controlled players (NPCs) too. In the case that the models concern specifically an opponent player, we speak of 'opponent modelling'. The goal of opponent modelling is to raise the playing strength of a (computer-controlled) player by allowing it to adapt to its (human) opponent and exploit his weaknesses [3–6]. In contrast, the general goal of player behavioural modelling often is to steer the game towards a predictably high player satisfaction [1] on the basis of modelled behaviour of the human player. Moreover, next to being useful for entertainment augmentation, player models are useful (among others) for simulation purposes (e.g., simulating stories or evaluating game maps), for game design purposes (e.g., testing whether the map leads to the gameplay as envisioned by the designers), and for serious game applications such as education (e.g., tailoring the game to a player's model for reaching particular learning objectives) or health (e.g., personalising games for rehabilitation of elderly patients).

Player behavioural modelling is of increasing importance in modern video games [7]. The main reason is that player behavioural modelling is almost a necessity when the purpose of AI is 'entertaining the human player' rather than 'defeating the human player' [1]. A challenge for such player modelling in video games is that models of the player have to be established (1) in game environments that generally are realistic and relatively complex, (2) with typically little time for observation, and (3) often with only partial observability of the environment. The online creation of player models, or the classification of the player into previously established models, is a task that has to be performed real-time, while other computations, such as rendering the game graphics, are performed simultaneously. Researchers estimate that generally only 20% of all computing resources are available to the game AI [8]. Of this per cent, a large portion will be spent on rudimentary AI behaviour, such as manoeuvring game characters within the game environment. This implies that only computationally inexpensive approaches to player modelling are suitable for incorporation in the game AI.

For the domain of modern video games, we deem four approaches applicable to player behavioural modelling, namely (1) modelling actions, (2) modelling tactics, (3) modelling strategies, and (4) profiling a player. In this taxonomy, action models concern game actions that can be observed directly or that can be inferred from other observations. Tactical models concern short-term/local game behaviour as composed of a series of game actions. Strategic

* Corresponding author. Tel.: +31 20 595 1675; fax: +31 20 595 1620.
E-mail addresses: s.c.j.bakkes@hva.nl (S.C.J. Bakkes), p.spronck@uvt.nl (P.H.M. Spronck), g.lankveld@uvt.nl (G. van Lankveld).

1875-9521/$ - see front matter © 2011 International Federation for Information Processing Published by Elsevier B.V. All rights reserved.
doi:10.1016/j.entcom.2011.12.001

models concern long-term/global game behaviour as composed of a series of game tactics, of which the behaviour may span the entire game, several game iterations, and across distinct games. Player profiling acknowledges that employing certain game actions, tactics, and strategies is motivated by the (psychological) profile of the player; distinct motivations and affect may result in distinct strategies, tactics, and actions. An illustration of the adopted taxonomy of player behavioural modelling is given in Fig. 1. Indeed, the defined classes are not mutually exclusive; one can for instance capture player tactics and a player profile in a single model.

In comparison, Sharma et al. [9] proposed a higher-order classification of player modelling, in which as distinction is made between (1) direct-measurement approaches (e.g., that utilise biometric data) and (2) indirect-measurement approaches (e.g., that infer the player's skill level from in-game observations). In the present paper, we focus on player behavioural modelling established via indirect measurements of the human player, by utilising actual in-game observations. Following the terminology of Smith et al. [10], what we investigate can be characterised as "induced reaction models with an individual or class scope". The taxonomy that we propose may be regarded as a refinement of those by Sharma et al. [9] and Smith et al. [10]. That is, Sharma et al. [9] would describe all four of the circles in Fig. 1 as "indirect measures", while Smith et al. [10] would describe each one as an "induced reaction model".

Following an extensive overview of background on the topic of player behavioural modelling, in this paper each of the four approaches to player behavioural modelling is discussed in detail, together with recommendations for applications of the approach, and insight into previously successful implementations. For readability, in the remainder of the paper we will refer to 'player behavioural modelling' as 'player modelling'.

2. Background

Though distinct in goal, the basis of player modelling is largely identical to that of opponent modelling: to improve the capabilities of a computer system (game) by allowing it to adapt to the user (player) and exploit his behavioural characteristics (cf. [3–6]). Even if a game-theoretical optimal solution to a game is known, a computer program that has the capability to model player behaviour may obtain a higher reward. An example that illustrates the importance of player modelling, derived from Fürnkranz [7], is as follows. Consider, the game of ROSHAMBO (also known as ROCK–PAPER–SCISSORS), where if both players play their optimal strategies (i.e., randomly select one of their three moves), either player can expect to win one third of the games (with one third of the games drawn). However, against an opponent player that always plays rock, a player that is able to adapt his strategy to always playing paper can maximise his reward, while a player that sticks with the 'optimal' random strategy will still win only one third of the games.

The concept of modelling the player's behaviour is regarded as important by numerous researchers [11–16]. Moreover, researchers state that player models are sorely needed to deal with the complexities of state-of-the-art video games [1,2]. One of the grand challenges in this line of work are games like POKER, where player modelling is crucial to improve over game-theoretically optimal play [17].

2.1. Player modelling in classic games

In classic games (e.g., chess, Go), player modelling has as its main goal raising the game results of the computer player [1]. The objective is to exploit the opponent player's weaknesses. Better game results are positively correlated with a higher playing strength. Computer programs that play classic games generally incorporate *search techniques* to find possible game actions by the opponent, of which a model can be constructed. As a result, the role of player modelling in classic games (and other games that use similar approaches) is to *guide the search process* towards improved results.

2.1.1. History

In the domain of classic games, player modelling is a research topic that was envisaged decades ago. Van den Herik et al. [1] observed that, for instance, in the 1970s, chess programs incorporated a contempt factor, meaning that against a stronger opponent player a draw was accepted even if the computer player was +0.5 ahead, and a draw was declined against a weaker opponent even when the computer player had a negative score.

The first attempt to model players in classic games was taken by Slagle and Dixon [18], who incorporated rudimentary knowledge of the opponent player in the search process. In general, such knowledge may concern assumptions on the fallibility of an opponent [19] (e.g., game AI can consider the chance that the opponent performs a non-rational game action). In related work, Jansen [20,21] investigated using knowledge about the opponent in game-tree search.

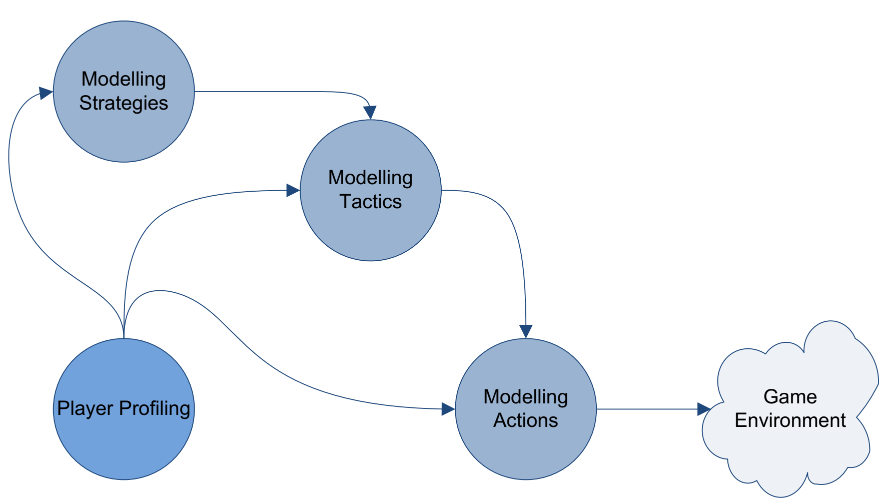

Fig. 1. Taxonomy of player behavioural modelling by means of indirect measurements of the human player (i.e., utilising actual in-game observations to generate player models).

Research specifically focussed on the topic of opponent-modelling search started in 1993. In that year, two research groups, one in Haifa, Israel and one in the Maastricht, Netherlands, simultaneously invented a search method that took knowledge of the opponent player into account. They both called it: opponent-model search. In Israel, Carmel and Markovitch [3] investigated in depth the learning of models of opponent strategies. In The Netherlands, Iida et al. [4] investigated potential applications of opponent-model search. An extensive description of the history of opponent modelling is given by Donkers [6].

In the year 1994, Uiterwijk and Van den Herik [22] invented a search technique to speculate on the fallibility of the opponent player. In the 2000s, Donkers et al. [5] and Donkers [6] defined probabilistic opponent models, that attempted to avoid the pitfalls of opponent modelling by incorporating the computer player's uncertainty about the opponent's strategy.

2.1.2. State of affairs

In the domain of classic games, the realisation of most ideas concerning player modelling is still in its infancy. There are a few successful instances of actual implementation, viz. (1) ROSHAMBO [23], (2) the iterated prisoner's dilemma [24], and (3) POKER [25]. Still, there is a wealth of techniques that are waiting for implementation in other games [1].

2.2. Player modelling in video games

Player modelling is of increasing importance in modern video games [7]. In video games, player modelling has as its main goal raising the entertainment factor (instead of raising the playing strength) [1]. For video-game environments, player modelling has distinct roles and challenges.

2.2.1. Role of player modelling

In order to raise the entertainment factor of a video game, game AI that incorporates player modelling may fulfil three roles: (1) as a companion, (2) as a coach, or (3) as an opponent. Each role entails distinct requirements for the game AI. A description of the three roles is given next. The description is largely derived from a review article by Van den Herik et al. [1], to which we refer the reader for more information on the topic.

2.2.1.1. Companion role. In the companion role, the game AI must behave according to the expectations of the human player. For instance, when the human player prefers an inconspicuous approach to dealing with hostile in-game characters (e.g., by attempting to remain undetected), he will not be pleased when the computer-controlled companions immediately attack every hostile character that is near. If the companions fail to predict with a high degree of success what the human player desires, they will likely annoy the human player, which, in turn, is detrimental for the entertainment value of the game.

2.2.1.2. Coaching role. In the coaching role, the game AI monitors closely the behaviour of the human player, and dependent on the goal of the game redirects the player's focus, or encourages a certain course of action. This is particularly the case for so-called 'serious games', where training is typically the purpose of the game, and personalised coaching often an inherent requirement. To this end, a good player model may assist the game in achieving its goals in an efficient and effective manner. Note that the coach is not necessarily supporting the player in his approach to the game; a drama manager such as used by Sharma et al. [26] would fall in the same category.

2.2.1.3. Opponent role. In the opponent role, the game AI must be able to match the playing skills of the human player, and respond adequately to the player's playing style. This is a task that is difficult to balance. Research shows that when the opponent characters play too weak a game against the human player, the human player loses interest in the game [27]. In contrast, when the opponent characters play too strong a game against the human player, the human player gets frustrated with the game and will quit playing too [28,29].

2.2.2. State of affairs

In recent years there have been several successful implementations of player modelling. For instance, Rohs [30] and Spronck and den Teuling [31] were able to model accurately several preferences of players in the game CIVILIZATION IV. Yannakakis [32] investigated the modelling of opponent players, for the purpose of augmenting player satisfaction, and Sailer et al. [33] incorporated player modelling to enhance simulation-based planning in RTS games. Laviers et al. [34,35] investigated improving AI performance through opponent modelling in the RUSH 2008 football simulator.

Van der Heijden et al. [36] applied player modelling to increase the effectiveness of strategies in a tactical game mode of the ORTS game. In the card game MACHIAVELLI, which shares numerous characteristics with modern video games, Bergsma [37] was successful in establishing and effectively utilising models of the opponent player. In the game GHOSTS, Aiolli and Palazzi [38,39] were able to enhance the game AI by allowing it to learn the opponent's playing style. In the game of GUESS IT, Lockett et al. [40] showed that opponent modelling could be used to learn effective game strategies. In the complex SPRING game, Schadd et al. [41] were able to generate automatically accurate models of the opponent player.

In addition, researchers are incorporating techniques to predict sequences of user actions (cf., e.g., Davison and Hirsh [42]), such as the position of opponent players in first-person shooters [43,44], and in the game WORLD OF WARCRAFT [45]. Wong et al. [46] investigated player modelling for a simple 2D shooter game. In the domain of interactive storytelling, Thue et al. [47] investigated how models of the opponent player can be applied to create stories that can be adapted to fit individual players.

Houlette [48], Charles and Black [49], Charles et al. [50], and Bohil and Biocca [51] discussed the challenges of player modelling in video-game environments, and suggested possible implementations of player modelling. As discussed in Section 1, only computationally inexpensive approaches to player modelling are suitable for incorporation in video game AI. In the sections that follow, we discuss in detail four such approaches.

3. Modelling player actions

A straightforward way to implement player modelling is by modelling the actions that a player executes. In its most general form, such an action model consists of a list of game states, each combined with one or more player actions, and a likelihood value that the player will undertake that action in the state. A perfect action model predicts exactly one action for each possible game state with 100% accuracy. In practice, action models are not necessarily implemented as a list of game states with associated actions, as the number of distinguishable game states is often very high. They may be implemented, for instance, as a list of rules that test the current game state and generate an appropriate action, or as a function that evaluates possible actions with respect to the current game state.

Action models originate in classical board game research. Player models for classical board games are almost exclusively used to enhance game tree search, i.e., to predict which moves the opponent will make in answer to the computer's moves. The player model is expressed as an evaluation function, which, in essence, determines a value for each possible opponent move and thus a likelihood that a move will be selected by the opponent [3,4,6]. As such, most player models in classical board games can be considered action models. Note that actually all tree-search techniques use player models, as by default they use the computer's own evaluation function to predict opponent moves; therefore, the model used is actually the computer itself.

ROBOSOCCER as a research environment is comparable to video games such as sports games and first-person shooters. The first explorations of the kind of player models that can be used in video games was performed in ROBOSOCCER. These models were predominantly action models, which specifically predict what kind of actions the opponent bots are going to take. For example, Ledezma et al. [52,53] use classification techniques to build action models of members of the champion team of the 2001 ROBOCUP edition.

A simple technique that has been proposed for building action models for video games is sequential prediction [54], specifically by the use of N-grams [55]. N-grams are sequences of choices, i.e., moves or actions. It is assumed that action sequences that have been observed in the past can be used to predict a future action. For instance, if it has been observed that when action A_1 is executed twice in a row, it is followed 75% of the time by action A_2, the prediction would be that there is a 75% likelihood of the next action being A_2 if the previous two observed actions were both A_1. In general, the more actions in the past are observed, the better the N-grams will function. The big problem with N-grams is, however, that they are *only* based on action sequences, while disregarding other state parameters. Therefore they mainly work for games in which the prediction of move sequences is key to gameplay, such as fighting games.

In games more complex than the previously mentioned, it might be hard to predict low-level actions as there are so many to choose from. However, actions might be predicted on a slightly higher level where the number of possible actions is manageable. For instance, Baker et al. [56] use probabilistic A^* path analysis to predict which of three targets an opponent had selected. Their player model consists of a simple probability distribution over all possible moves that the player might make. While it might be hard to predict the exact next move of the player, their model can determine the selected target with high accuracy. Similar work in a more complex game by Butler and Demiris [57] uses an approach inspired by the Theory of Mind, in which they predict the selection of a target of a team of units in an RTS game, by mapping the team's movement to A^* paths which lead to the respective targets.

For many games a limited number of player types can be distinguished, each with a predisposition for specific action choices. An action model of a particular player can then be defined as a series of weights for each of the possible player types, and the predicted choice of action can be determined as a weighted voting by all the types. This is the basis behind the strongest player models for TEXAS HOLD'EM POKER [25,58], but is also used for other games, such as GUESS IT [40].

The big advantage of action models is that they are easy to employ by an AI. If it is known which action the opponent is going to take, it is easy to block the action or avoid confrontation, if desired.

There are two disadvantages, however. The first is that states in video games typically encompass an enormous number of parameters, and the number of different actions is usually also large. This leads to an unmanageable state–action space. Added to that is the fact that in most games state information is incomplete. The consequence is that for action models to be learned efficiently, state information must be restricted to a few simplified features, which are usually insufficient for building an acceptable action model except for very simple games.

The second disadvantage is that it is hard to make an action model generalise directly to new situations (i.e., without requiring additional learning trials). Virtually all action models used in practice are based on direct observations of behaviour, and couple observed actions to observed states. Consequently, they do not know the reasons for actions and thus may have problems being effective in different circumstances. For example, suppose that a human player controls a fighter character in a role-playing game, and an action model is determined for his behaviour. When later the human player controls a wizard character, with a different list of possible actions, he might use aggressive spells in situations where his fighter character used melee attacks. A equivalent play style leads to different actions, so without understanding the play style, the model of the player controlling a fighter does not transfer to controlling a wizard.[1]

We conclude that action models might be of use for the relatively simple task of responding to specific low-level player actions in specific games, but that it is hard to make them generalise directly to different games or even to different situation in the same game.

4. Modelling player tactics

Here we look into the topic of modelling player tactics (4.1), highlight a previously successful implementation (4.2), and discuss the benefits of modelling player tactics in video games (4.3).

4.1. Description of the approach

A tactic, in the military sense, is defined as the 'art of organising an army; the techniques for using weapons or military units in combination for engaging and defeating an enemy in battle' [59]. In the context of video games, we define tactics analogously, being a relatively low planning level that involves one to several units, organised in such a way as to achieve a particular local goal. Tactics, therefore, can be interpreted as the motives for player actions.

Modelling player tactics, by extension, can be defined as creating player models by means of automatically distiling the organisation and particular goal of certain in-game characters, as exemplified by the observed character actions. For instance, in video games, game characters may manoeuvre together in a so-called squad, enabling certain vulnerable characters to be protected by strong characters, and enabling the strong characters to benefit from distinct other qualities of the vulnerable characters. Indeed, the composition of a formation of characters may constitute a tactic by itself. As another example, in numerous video games a tactic may concern precisely *how* characters manoeuvre in a squad. That is, the internal arrangement of characters may have particular tactical advantages, say in numerous first-person shooter (FPS) games. To provide additional examples: in a platform game, such as SUPER MARIO BROS., a tactic may constitute a series of game actions leading to the local goal of defeating a mini-boss. In a casual racing game, such as SUPER MARIO KART, a tactic may be to purposely wait with using a particular power-up, and in an adventure game, such as THE SECRET OF MONKEY ISLAND, a tactic may be to randomly apply all gathered objects with all other in-game objects (arguably, a generally inefficient tactic).

[1] Indeed, one may correctly note that with additional learning trials, generalisation may nevertheless be obtained. For instance, using techniques such as inverse reinforcement learning (IRL), it would be possible to learn from the fighter character, and perfectly generalise to the wizard character. This is because, in IRL, we infer a reward function from the observed low-level user actions, and then learn a policy that optimises that reward function (mimicking the human). Since we have the reward function, changing the action set (moving to a wizard), only requires learning a different optimal policy that optimises the learnt reward function. Techniques such as IRL work at the action level, can predict human actions, can reason about motivations, and do generalise. The penalty, of course, is computational complexity and scalability.

Modelling player tactics is achieved generally by incorporating learning techniques, or applying methods for case-based reasoning over in-game observations. For instance, Auslander et al. [60] uses case-based reasoning to allow reinforcement learning to respond more quickly to tactically-changing circumstances in the UNREAL TOURNAMENT domination game. Laviers et al. [34,35] demonstrate how the exploitation of tactical models may increase game AI effectiveness in the game of football.

4.2. Highlighted implementation

Following our definition of modelling player tactics (i.e., creating player models by means of automatically distiling the organisation and particular goal of certain in-game characters, as exemplified by the observed character actions), we highlight an implementation by Van der Heijden et al. [36,61]. It models tactical *formations* of game characters. A formation is an arrangement or disposition of units [62], is typically applied for a tactical purpose, and has already been found in tribal societies such as the Maori [63]. Commonly seen formations, such as a shield wall, a phalanx or a wedge, have historical significance and are still used in modern military operations [64].

An important factor that influences the effectiveness of a formation, is the formation employed by the opponent player. To make predictions about the behaviour of the opponent, an AI player first needs to model the opponent player's behaviour. In the present context, therefore, the challenge is to model character behaviour on a local level, specifically, to model the formations that underlie the organisation of game characters. An adequate selection of tactical features is instrumental herein. In the highlighted implementation, explicit opponent models [65] are generated on the basis of three behavioural features: (1) Number of formations employed by the opponent, (2) Unit distribution, and (3) Unit distance.

Opponent models generated on this basis, are used for two purposes: (1) classifying opponent players in actual online gameplay, based on actual game observations and (2) for distinct opponent classifications, applying previously learned successful behaviour online. In experiments, Van der Heijden observes that when competing with known opponent players, applying the previously learned behaviour enables the game AI to be effective from the onset of the game. Moreover, when in competition against a previously unknown player, i.e., an opponent player whose playing features have not been captured in the player models, one can still apply the player models for the purpose of improving game behaviour. Namely, unknown players generally exhibit similarities with regard to their behavioural features. It was therefore observed that behaviour that is successful against one particular player, was to a large extent also successful against similar players.

4.3. Discussion of modelling player tactics

It is safe to state that the advantage of tactics modelling over modelling player actions directly, is that, inherently, the state-space complexity decreases as a result of the higher information abstraction. Also, a modest form of generalisation is incorporated that may suffice for many tactical games. The main disadvantage of tactics modelling is that the information abstraction that is being modelled concerns foremost behaviour on a local scale, while the intricacies of the local interactions with the higher-level, i.e., strategic play of the game is not incorporated in the models. Therefore, solely on the basis of tactical models one cannot generalise over the underlying intentions behind the observed tactics.

Hence, we consider tactical models most useful in (action-based) games that are predominantly tactics-oriented, such as squad-based first-person shooters (FPS). In more complex games such as real-time strategy (RTS) games, however, tactics are not independent of each other, but interrelated and aimed at reaching an overarching goal. In such games, tactical models alone are insufficient for effectively modelling player behaviour.

5. Modelling player strategies

Here we look into the topic of modelling player strategies (5.1), highlight a previously successful implementation (5.2), and discuss the benefits of modelling player strategies in video games (5.3).

5.1. Description of the approach

Strategy, a word of military origin, refers to a plan of action designed to achieve a particular goal. In military usage strategy is distinct from tactics, which are concerned with the conduct of an engagement on a local scale, while strategy is concerned with how different tactical engagements are linked [59]. Strategy concerns generally a method to conflict, and arguably conflict resolution, between at least two sides. These sides interact, and thus a strategy will rarely be successful if it shows no adaptability [66]. Hence, strategy is concerned with the overall means and plan for achieving a long-term outcome. Where in game theory it is common that every player in a non-cooperative game has a set of possible strategies (and must choose one of the choices), in video games it is common that a player needs to envision possible strategies himself from the interplay of employed game tactics.[2]

Modelling player strategies, by extension, can be defined as creating player models by automatically distiling a player's overall means and plan for achieving a long-term outcome, as exemplified by the observed player tactics. Theoretically speaking, modelling player strategies strictly builds upon models of player tactics and well-defined features that represent or imply player tactics. As such, the methods for implementing modelling player strategies have a large degree of overlap with implementing modelling player tactics, in respect that typically a feature abstraction is observed and utilised for tasks such as creating player models and steering game AI behaviour. For example, observing a formation of opponent characters moving into a flanking position may indicate an imminent attack in case the opponent player follows an aggressive strategy. On the other hand, should the opponent player follow a defensive strategy, the flanking characters may indicate the opponent player is merely gathering in-game resources on the particular location, and pose no further threat. As another, more generic example, in numerous video games patterns may be discovered from observed opponent tactics, that form a prelude to future opponent tactics, which are part of an overall strategy. For instance, also in less strategic game genres, game strategies may exist, such as executing a 'speed run' strategy in platform games, and deciding on a one, two, or three pit-stop strategy in advance of playing a Formula 1 racing game.

Research into modelling player strategies in video games is sparse. Ontañón et al. [67] presented a framework for high-level case-based planning on the basis of annotated knowledge drawn from expert demonstrations in the WARGUS game. For strategy prediction, data mining techniques have been applied successfully by Weber and Mateas [68]. Indeed, if a player's strategy can be modelled adequately, one may be able to reason accurately on an effective counter strategy.

[2] We note that, though 'action-tactic-strategy' is a common division of a continuum of abstraction levels, it certainly is not intended as an exhaustive division. Indeed, in any sufficiently complex game, there may be additional abstraction levels between the three proposed categories, at which one could reason in finer detail.

5.2. Highlighted implementation

Following our definition of modelling player strategies (i.e., creating player models by automatically distiling a player's overall means and plan for achieving a long-term outcome, as exemplified by the observed player tactics), we highlight an implementation by Bakkes et al. [69–73]. The approach concerns the online exploitation of a data store of actual game observations, and is validated in the complex real-time strategy (RTS) game SPRING. Here, we focus on how strategic player modelling was utilised for improving the effectiveness of decisions made by an adaptive game AI. In this context, the challenge is to assess which high-level strategy a player adopts in terms of unit preference, in-game technological development, economy, and playing style (e.g., aggressive/defensive), and effectively exploiting this knowledge for steering behaviour expressed by the game AI.

In the highlighted implementation, player models are created automatically. It happens on the basis of game observations gathered in the data store of observations. As a first step to modelling player strategies, features of an opponent's high-level strategic behaviour are defined and selected. Strategic behaviour (e.g., the opponent's preference of unit type, the focus of an opponent's technological development, the strength of his economy, and the aggressiveness of the opponent) can generally be inferred from observing the values of selected features during actual play: (1) Number of observed k-bot units, (2) Number of observed tank units, (3) Number of observed aircraft units, (4) Number of technologically advanced buildings (i.e., level 2 or higher), (5) Number of metal extractors, (6) Number of solar panels, (7) Number of wind turbines, (8) Time of first attack on one of the metal extractors, (9) Time of first attack on one of the solar panels, and (10) Time of first attack on one of the wind turbines.

The first three features express the global strategic preference of an opponent player. The fourth feature expresses the technological development of a player. The fifth, sixth, and seventh feature express the strength of an opponent's economy, and by implication, the strength of the opponent's army. The eighth, ninth, and tenth feature express the aggressiveness of the opponent player.

Player models were generated based on observations gathered from play on three distinct maps. The generated models (Table 1) reveal that opponents observed on the map SmallDivide typically employ a defensive playing style, have a preference for constructing advanced buildings, and have a preference for constructing tank units. Opponents observed on the map TheRing are typically similar to those observed on the map SmallDivide, with the difference that they have a preference for constructing k-bot units, instead of tank units. Opponents observed on the map MetalHeckv2 typically employ an aggressive playing style, do not have a preference for constructing advanced buildings, and have a preference for constructing tank units.

The generated player models are exploited by labelling each game in the data store with the classification of the opponent player against whom the game AI was pitted. The classification of the opponent player is exploited for (A) intelligent initialisation of game AI (i.e., determine which opponent the game AI is likely to be pitted against, and initialise with a predictably effective strategy) and (B) online strategy selection in actual play (i.e., more accurately compare similarities in previous game observations for steering game AI behaviour to a desirable state).

Table 1
Overview of the characteristics of the generated player models.

	Map: SmallDivide	Map: TheRing	Map: MetalHeckv2
# Player models	9	8	9
Typical playing style	Defensive	Defensive	Aggressive
Building preference	Advanced buildings	Advanced buildings	No adv. buildings
Unit preference	Tank units	K-bot units	Tank units

In the performed experiments, it is observed that applying player modelling techniques generally increases the effectiveness of the adaptive game AI. Though the increase in performance was based on effectively utilising observations on opponent players, naturally, the adaptive game AI may still be confronted with an opponent that it has not observed previously. In this situation, the inherent generalisation that is provided by the implemented clustering of player models was observed to already have led to the game AI being initialised with a strategy that is also effective against the previously unobserved opponent. Should the strategy still be ineffective, then it can be adapted during online play. In any case, in the implemented adaptive game AI, the next time that player models are generated, the just-observed game will be included in the data store of observations. As a result, the previously unobserved opponent will be covered in the player models, and accordingly game AI that is predictably more effective will be generated automatically.

5.3. Discussion of modelling player strategies

A disadvantage of modelling player strategies is that it requires a multitude of observations before strategic models may be considered accurate. Indeed, this may also be the case to some degree in modelling player tactics, though we surmise that determining a player's local goal (tactics) generally requires fewer observations than determining a player's higher-order goal (strategy). Also, even when accurate models have been generated, it may not necessarily imply that a counter strategy is directly available for application; it may require an additional learning process. Finally, the classification of a certain strategy may be hampered by a so-called 'fog of war', which renders the game environment only partially visible to the player.

However, the advantage of modelling player strategies concern generalisation of the observed strategies over other repetitions of the game, over games played against distinct other players, and even over distinct other games in the genre (as common strategies exist within game genres). We note that in our demarcation of actions, tactics, and strategies, a strategy is to be considered a higher-order goal entailing a generic fashion in which it can be achieved, i.e., playing aggressively, having a preference for certain tactics, etc. We surmise that in most games such generic, higher-order strategies can be defined or discovered automatically. Indeed, early notice of which strategy which player is employing is of particular interest to other players and game AI. Particularly, it can be utilised to steer the game into a direction of more (or if desired, less) challenge, or may serve as a basis for feedback to the human player on the effectiveness of his exhibited behaviour (e.g., in the context of a training mechanism of a serious game).

6. Player profiling

A fairly recent development with regard to player modelling, is to establish automatically psychologically or sociologically verified player profiles. Such models provide motives or explanations for observed behaviour, regardless whether it concerns strategic behaviour, tactical behaviour, or actions. A solid profile can be used to, for instance, predict a player's affective state while playing a game. In that respect player profiling is closely related to affective player modelling, a research domain that attempts to capture a player's affective state by direct observation [74,75]. Player profiling can be regarded as a form of user modelling. User modelling is usually aimed at capturing an application user's emotions, and has seen considerable research interest in the last decade (cf. [76–79]).

Lankveld et al. [80] states that the major differences between player modelling and player profiling lie in the features that are modelled. That is, player modelling generally attempts to model the player's external behavioural attributes (e.g., tactics and playing style), while player profiling attempts to model internal traits of the player (e.g., personality and preferences). The models produced by player profiling are readily applicable in any situation where conventional personality models can be used. In addition, player profiling is supported by a large body of psychological knowledge.

A leading contributor to this line of work is Yannakakis et al., who in previous work investigated the cognitive modelling of players [32], and focussed particularly on applying the models for the purpose of optimising player satisfaction [81,82]. Bohil and Biocca [51] also investigate cognitive modelling of video game players, particularly with a focus on adapting information interfaces. Recently, Yannakakis and Togelius [83] published a general framework of Player Experience Modelling, of which player profiling can be considered a subset. Van Lankveld et al. [84] determined that in an actual video game (i.e., NEVERWINTER NIGHTS), it is indeed possible to create an adequate personality profile of a player. In his research, all factors of the Five Factor Model of personality (FFM) (cf. [85,86]) could be modelled, consisting of the factors (1) openness to new experience, (2) conscientiousness, (3) extraversion, (4) agreeableness, and (5) neuroticism.

In this regard, player profiling extends the applicability of player modelling techniques to domains where dramatic effect and social adequacy play an important role. For instance, the domain of interactive storytelling has seen numerous advances in terms of chaining together appropriate actor actions, directing scenes toward a dramatic goal (e.g., the work of Mateas and Stern [87]), or planning to achieve a learning objective [88–90]. The evaluation of player models for interactive fiction has been investigated by Sharma et al. [9]. Particularly, player modelling may enrich systems by incorporating psychologically verified knowledge on player satisfaction and experience. A promising system in this regard, is PaSSAGE (still in development), an interactive storytelling system which bases its storytelling decision on an automatically learned model of each player's style of play [47,91]. Also, in the framework of procedural content generation, the importance of including player models has been suggested previously [92], as well as investigated [83,93,94].

One disadvantage of player profiling is that generated profiles are, at least initially, player specific. Generating online a new player profile requires a multitude of player observations. Therefore, well designed applications of player profiling strive for a certain extent of generalisation over models, with regard to player characteristics (e.g., extraversion) or playing style (e.g., aggression). The main advantage of player profiling, on the other hand, is the information richness it may provide on player experience and satisfaction. This may be of particular interest to so-called serious games, that for instance train individuals to certain situations. In addition, and most importantly, an accurate player profile will generalise fully to other games and other domains. That is, the profile covers inherent personal traits and characteristics of a player, which will be of interest for many applications.

7. Conclusions

Following a focus on player behavioural modelling established via indirect measurements of the human player, in this article we distinguished four types of player models: (1) action models, (2) tactical models, (3) strategic models, and (4) player profiles. If we examine these types of models, respectively, we notice that they are increasingly resource-intensive to construct; however, they also increasingly generalise better. When considering the predictive capabilities of these types of models, action models attempt to do what most game developers would like player models to do, namely predict player actions. If exact future actions are known, determining a good response is relatively easy. While action prediction seems an attractive possibility of a model, in practice it is of limited use, unless the games concerned are relatively uncomplicated. The predictions of the other model types become increasingly less specific, but also more generally applicable for direct use (i.e., without requiring additional learning trials).

Tactical and strategic models have a lot of potential, especially when the goal of a game is to provide a strong challenge for the human player (the highlighted implementations are a good example of this). Inherently, tactical and strategic models are capable of more generalisation than is possible on solely an action-state level. Hence, tactical and strategic models provide more means for game developers to personalise and adapt the provided game experience and challenge to the level of individual players.

Player profiling is of a different calibre, though comprised of predominantly ongoing research. By incorporating psychologically-verified knowledge in player models, as well as knowledge on player experience and satisfaction, player profiling may potentially have a substantial (and more directly noticeable) impact on the experience that users have with a gaming system. Indeed, numerous cross-domain applications exist for player modelling approaches, such as in interactive storytelling, or in gaming environments that are generated online, on the basis of a player's behaviour and experience.

Acknowledgements

Part of this research was funded by Agentschap.nl, in the context of the EOS Project Persuasive Agents, and the SIA RAAK Project Smart Systems for Smart Services.

The authors wish to thank the anonymous referees for their constructive comments that helped to improve the article considerably.

References

[1] H.J. Van den Herik, H.H.L.M. Donkers, P.H.M. Spronck, Opponent modelling and commercial games, in: G. Kendall, S. Lucas (Eds.), Proceedings of the IEEE 2005 Symposium on Computational Intelligence and Games (CIG'05), IEEE Press, Piscataway, New York, USA, 2005, pp. 15–25.
[2] S. Rabin, Preface, in: S. Rabin (Ed.), AI Game Programming Wisdom 4, Charles River Media, Inc., Hingham, Massachusetts, USA, 2008, pp. ix–xi.
[3] D. Carmel, S. Markovitch, Learning models of opponent's strategy in game playing, in: Proceedings of AAAI Fall Symposium on Games: Planning and Learning, Raleigh, NC, 1993, pp. 140–147.
[4] H. Iida, J.W.H.M. Uiterwijk, H.J. Van den Herik, I.S. Herschberg, Potential applications of opponent-model search: Part 1. The domain of applicability, International Computer Chess Association (ICCA) Journal 16 (4) (1993) 201–208.
[5] H.H.L.M. Donkers, J.W.H.M. Uiterwijk, H.J. Van den Herik, Probabilistic opponent-model search, Information Sciences 135 (3–4) (2001) 123–149.
[6] H.H.L.M. Donkers, Nosce Hostem - Searching with Opponent Models, Ph.D. Thesis, Faculty of Humanities and Sciences, Maastricht University, Maastricht, The Netherlands, 2003.
[7] J. Fürnkranz, Recent advances in machine learning and game playing, ÖGAI-Journal 26 (2) (2007) 19–28.
[8] I. Millington, Artificial Intelligence for Games, Morgan Kaufmann Publishers, San Francisco, California, USA, 2006.
[9] M. Sharma, M. Mehta, S. Ontanón, A. Ram, Player modeling evaluation for interactive fiction, in: Proceedings of the AIIDE 2007 Workshop on Optimizing Player Satisfaction, 2007, pp. 19–24.
[10] A.M. Smith, C. Lewis, K. Hullet, G. Smith, A. Sullivan, An inclusive view of player modeling, in: Proceedings of the 2011 International Conference on the Foundations of Digital Games (FDG 2011), 2011.
[11] A. Samuel, Some studies in machine learning using the game of checkers: II. Recent progress, IBM Journal 11 (6) (1967) 601–617.

[12] D.E. Knuth, R.W. Moore, An analysis of alpha–beta pruning, Artificial Intelligence 6 (4) (1975) 293–326.
[13] H. Berliner, Search and knowledge, in: Proceedings of the Fifth International Joint Conference on Artificial Intelligence (IJCAI 77), 1977, pp. 975–979.
[14] R. Korf, Generalized game trees, in: Proceedings of the 11th International Joint Conference on Artificial Intelligence (IJCAI 89), 1989, pp. 328–333.
[15] B. Abramson, Expected outcome: a general model of static evaluation, IEEE Transactions on Pattern Analysis and Machine Intelligence 12 (2) (1990) 182–193.
[16] J. Schaeffer, J. Culberson, N. Treloar, B. Knight, P. Lu, D. Szafron, A world championship caliber Checkers program, Artificial Intelligence 53 (2–3) (1992) 273–290.
[17] D. Billings, L. Peña, J. Schaeffer, D. Szafron, The challenge of Poker, Artificial Intelligence 134 (1–2) (2002) 201–240. Special Issue on Games, Computers and Artificial Intelligence.
[18] J.R. Slagle, J.K. Dixon, Experiments with the M & N tree-searching program, Communications of the ACM 13 (3) (1970) 147–154.
[19] A.L. Reibman, B.W. Ballard, Nonminimax search strategies for use against fallible opponents, in: J.M. Tenenbaum (Ed.), Proceedings of the Third National Conference on Artificial Intelligence (AAAI-83), Morgan Kaufmann Publishers, San Francisco, California, USA, 1983, pp. 338–342.
[20] P.J. Jansen, Using Knowledge about the Opponent in Game-Tree Search, Ph.D. Thesis, School of Computer Science, Carnegie Mellon University, Pittsburgh, Pennsylvania, USA, 1992.
[21] P.J. Jansen, KQKR: speculative thwarting a human opponent, International Computer Chess Association (ICCA) Journal 16 (1) (1993) 3–18.
[22] J.W.H.M. Uiterwijk, H.J. Van den Herik, Speculative play in computer chess, in: H.J. Van den Herik, I.S. Herschberg, J.W.H.M. Uiterwijk (Eds.), Advances in Computer Chess, vol. 7, Maastricht University, Maastricht, The Netherlands, 1994, pp. 79–90.
[23] D. Egnor, Iocaine power, International Computer Games Association (ICGA) Journal 23 (1) (2000) 33–35.
[24] G. Kendall, Iterated Prisoner's Dilemma Competition (Online), 2005. Available from: <http://www.prisoners-dilemma.com/>.
[25] D. Billings, Algorithms and Assessment in Computer Poker, Ph.D. Thesis, Department of Computing Science, University of Alberta, Edmonton, Alberta, Canada, 2006.
[26] M. Sharma, S. Ontañón, M. Mehta, A. Ram, Drama management and player modeling for interactive fiction games, Computational Intelligence 26 (2) (2010) 183–211.
[27] B. Scott, The illusion of intelligence, in: S. Rabin (Ed.), AI Game Programming Wisdom, Charles River Media, Inc., Hingham, Massachusetts, USA, 2002, pp. 16–20.
[28] D. Livingstone, D. Charles, Intelligent interfaces for digital games, in: D. Fu, S. Henke, J. Orkin (Eds.), Proceedings of the AAAI-04 Workshop on Challenges in Game Artificial Intelligence, AAAI Press, Menlo Park, California, USA, 2004, pp. 6–10.
[29] G. Van Lankveld, P.H.M. Spronck, H.J. Van den Herik, Incongruity-based adaptive game balancing, in: H.J. Van den Herik, P.H.M. Spronck (Eds.), Proceedings of the 12th Advances in Computer Games Conference (ACG12), Tilburg Centre for Creative Computing (TiCC), Tilburg University, Tilburg, The Netherlands, Springer-Verlag, Heidelberg, Germany, 2009, pp. 208–220.
[30] M. Rohs, Preference-Based Player Modelling for Civilization IV, Bachelor's Thesis, Faculty of Humanities and Sciences, Maastricht University, Maastricht, The Netherlands, 2007.
[31] P.H.M. Spronck, F. den Teuling, Player modeling in civilization IV, in: Proceedings of the Sixth Artificial Intelligence and Interactive Digital Entertainment Conference, AAAI Press, Menlo Park, CA, 2010, pp. 180–185.
[32] G.N. Yannakakis, How to model and augment player satisfaction: a review, in: K. Berkling, D. Giuliani, A. Potamianos (Eds.), Proceedings of the First Workshop on Child, Computer and Interaction (ICMI'08 Post-Conference Workshop), ACM Press, Montreal, Canada (Invited Paper), 2008.
[33] F. Sailer, M. Lanctot, M. Buro, Simulation-based planning in RTS games, in: S. Rabin (Ed.), AI Game Programming Wisdom 4, Charles River Media, Inc., Hingham, Massachusetts, USA, 2008, pp. 405–418.
[34] K. Laviers, G. Sukthankar, M. Klenk, D.W. Aha, M. Molineaux, Opponent modeling and spatial similarity to retrieve and reuse superior plays, in: L. Lamontagne, P.G. Calero (Eds.), Proceedings of the Workshop on Case-Based Reasoning for Computer Games, Eighth International Conference on Case-Based Reasoning (ICCBR 2009), AAAI Press, Menlo Park, California, USA, 2009, pp. 97–106.
[35] K. Laviers, G. Sukthankar, M. Molineaux, D.W. Aha, Improving offensive performance through opponent modeling, in: Proceedings of the Fifth Artificial Intelligence and Interactive Digital Entertainment Conference (AIIDE 2009), 2009, pp. 58–63.
[36] M.J.M. Van der Heijden, S.C.J. Bakkes, P.H.M. Spronck, Dynamic formations in real-time strategy games, in: P. Hingston, L. Barone (Eds.), Proceedings of the IEEE 2008 Symposium on Computational Intelligence and Games (CIG'08), IEEE Press, Piscataway, New York, USA, 2008, pp. 47–54.
[37] M. Bergsma, Opponent Modelling in Machiavelli, Bachelor's Thesis, Faculty of Humanities and Sciences, Maastricht University, Maastricht, The Netherlands, 2005.
[38] F. Aiolli, C.E. Palazzi, New frontiers for entertainment computing, IFIP International Federation for Information Processing, Chap. Enhancing Artificial Intelligence in Games by Learning the Opponent's Playing Style, vol. 279, Springer-Verlag, Heidelberg, Germany, 2008, pp. 1–10.
[39] F. Aiolli, C.E. Palazzi, Enhancing artificial intelligence on a real mobile game, International Journal of Computer Games Technology (2009). article ID 456169.
[40] A.J. Lockett, C.L. Chen, R. Miikkulainen, Evolving explicit opponent models in game playing, in: Proceedings of the Ninth Annual Conference on Genetic and Evolutionary Computation (GECCO-2007), 2007, pp. 2106–2113.
[41] F.C. Schadd, S.C.J. Bakkes, P.H.M. Spronck, Opponent modeling in real-time strategy games, in: M. Roccetti (Ed.), Proceedings of the Eighth International Conference on Intelligent Games and Simulation (GAMEON'2007), EUROSIS-ETI, Ghent University, Ghent, Belgium, 2007, pp. 61–68.
[42] B.D. Davison, H. Hirsh, Predicting sequences of user actions, in: Predicting the Future: AI Approaches to Time-Series Problems, AAAI Press, Menlo Park, California, USA, 1998, pp. 5–12.
[43] C.J. Darken, B.G. Anderegg, Particle filters and simulacra for more realistic opponent tracking, in: S. Rabin (Ed.), AI Game Programming Wisdom 4, Charles River Media, Inc., Hingham, Massachusetts, USA, 2008, pp. 419–428.
[44] S. Hladky, V. Bulitko, An evaluation of models for predicting opponent locations in first-person shooter video games, in: P. Hingston, L. Barone (Eds.), Proceedings of the IEEE 2008 Symposium on Computational Intelligence and Games (CIG'08), IEEE Press, Piscataway, New York, USA, 2008, pp. 39–46.
[45] A.J.J. Valkenberg, Opponent Modelling in World of Warcraft, Bachelor's Thesis, Faculty of Humanities and Sciences, Maastricht University, Maastricht, The Netherlands, 2007.
[46] C. Wong, J. Kim, E. Han, K. Jung, Human-centered modeling for style-based adaptive games, Journal of Zhejiang University – Science A 10 (4) (2009) 530–534.
[47] D.J. Thue, V. Bulitko, M. Spetch, Player modeling for interactive storytelling: a practical approach, in: S. Rabin (Ed.), AI Game Programming Wisdom 4, Charles River Media, Inc., Hingham, Massachusetts, USA, 2008, pp. 633–646.
[48] R. Houlette, Player modeling for adaptive games, AI Game Programming Wisdom 2, Charles River Media, Inc., Hingham, Massachusetts, USA, 2004, pp. 557–566.
[49] D. Charles, M. Black, Dynamic player modeling: a framework for player-centered digital games, in: Q.H. Mehdi, N.E. Gough, D. Al-Dabass (Eds.), Proceedings of Computer Games: Artificial Intelligence, Design and Education (CGAIDE 2004), University of Wolverhampton, Wolverhampton, UK, 2004, pp. 29–35.
[50] D. Charles, M. McNeill, M. McAlister, M. Black, A. Moore, K. Stringer, J. Kucklich, A. Kerr, Player-centered design: player modeling and adaptive digital games, in: Proceedings of the DiGRA Conference: Changing Views – World in Play, 2005, pp. 285–298.
[51] C.J. Bohil, F.A. Biocca, Cognitive Modeling of Video Game Players, Technical Report, Media, Interface, and Network Design Labs (MINDLab), Department of Telecommunication, Information and Media, Michigan State University, East Lansing, Michigan, USA, 2007.
[52] A. Ledezma, R. Aler, A. Sanchis, D. Borrajo, Predicting opponent actions by observation, RoboCup 2004: Robot Soccer World Cup VIII (2005) 286–296.
[53] A. Ledezma, R. Aler, A. Sanchis, D. Borrajo, OMBO: an opponent modeling approach, AI Communications 22 (1) (2009) 21–35. ISSN 0921-7126.
[54] F. Mommersteeg, Pattern recognition with sequential prediction, in: S. Rabin (Ed.), AI Game Programming Wisdom, Charles River Media, Inc., Hingham, Massachusetts, USA, 2002, pp. 586–595.
[55] F.D. Laramée, Using *n*-gram statistical models to predict player behavior, in: S. Rabin (Ed.), AI Game Programming Wisdom, Charles River Media, Inc., Hingham, Massachusetts, USA, 2002, pp. 596–601.
[56] R. Baker, P. Cowling, T. Randall, P. Jiang, Using Bayes Theorem for Path Prediction, in: D. Rigas (Ed.), 9th Informatics Research Workshop for Research Students, University of Bradford, UK, 2008.
[57] S. Butler, Y. Demiris, Using a Cognitive Architecture for Opponent Target Prediction, in: D.M. Romano, D.C. Moffat (Eds.), Proceedings of the Third International Symposium on AI & Games, 2010, pp. 55–62.
[58] D. Dizman, A Survey on Artificial Intelligence in Stochastic Games of Imperfect Information: Poker, June 22, 2010 (Online). Available from: <http://dendiz.com/blog/wp-content/uploads/2010/06/swe578_paper_v2.pdf/>.
[59] C. von Clausewitz, On War, first ed., Princeton University Press, 1989. ISBN-13: 978-0691018546, translated by M. Howard and P. Paret, original title: 'Vom Kriege', 1832.
[60] B. Auslander, S. Lee-Urban, C. Hogg, H. Muñoz-Avila, Recognizing the enemy: combining reinforcement learning with strategy selection using case-based reasoning, in: K.-D. Althoff, R. Bergmann, M. Minor, A. Hanft (Eds.), Proceedings of the Ninth European Conference on Case-Based Reasoning (ECCBR 2008), University of Trier, Trier, Germany, 2008, pp. 59–73.
[61] M.J.M. Van der Heijden, Dynamic Formations in Real-Time Strategy Games, Master's Thesis, Faculty of Humanities and Sciences, Maastricht University, Maastricht, The Netherlands, 2008 (Online). Available from: <http://ticc.uvt.nl/pspronck/theses/Thesis_Heijden.pdf/>.
[62] W. Trumble, L. Brown, Shorter Oxford English Dictionary, fifth ed., Oxford University Press, Oxford, England, 2002.
[63] C.M.G. Gudgeon, Maori wars, Journal of the Polynesian Society 16 (1) (1907) 13–42.
[64] C. Dawson, Formations, in: S. Rabin (Ed.), AI Game Programming Wisdom, Charles River Media, Inc., Hingham, Massachusetts, USA, 2002, pp. 272–281.
[65] D. Carmel, S. Markovitch, Exploration and adaptation in multiagent systems: a model-based approach, in: M.E. Pollack (Ed.), Proceedings of the 15th International Joint Conference for Artificial Intelligence (IJCAI 97), Morgan Kaufmann Publishers, San Francisco, California, USA, 1997, pp. 606–611.

[66] B. Heuser, The Evolution of Strategy: Thinking War from Antiquity to the Present, Cambridge University Press, 2010. ISBN-13: 978-0521155243.
[67] S. Ontañón, K. Mishra, N. Sugandh, A. Ram, Case-based planning and execution for real-time strategy games, in: R.O. Weber, M.M. Richter (Eds.), Proceedings of the Seventh International Conference on Case-Based Reasoning (ICCBR 2007), Springer-Verlag, Heidelberg, Germany, 2007, pp. 164–178.
[68] B. Weber, M. Mateas, A data mining approach to strategy prediction, in: P. Hingston, L. Barone (Eds.), IEEE Symposium on Computational Intelligence and Games, 2009, pp. 140–147.
[69] S.C.J. Bakkes, P.H.M. Spronck, H.J. Van den Herik, Rapid and reliable adaptation of video game AI, IEEE Transactions on Computational Intelligence and AI in Games 1 (2) (2009) 93–104.
[70] S.C.J. Bakkes, P.H.M. Spronck, H.J. Van den Herik, Opponent modelling for case-based adaptive game AI, Entertainment Computing 1 (1) (2009) 27–37.
[71] S.C.J. Bakkes, P.H.M. Spronck, H.J. Van den Herik, Rapid adaptation of video game AI (extended version), in: P. Hingston, L. Barone (Eds.), Proceedings of the IEEE 2008 Symposium on Computational Intelligence and Games (CIG'08), IEEE Press, Piscataway, New York, USA, 2008, pp. 79–86.
[72] S.C.J. Bakkes, P.H.M. Spronck, H.J. Van den Herik, Rapid adaptation of video game AI, in: V. Botti, A. Barella, C. Carrascosa (Eds.), Proceedings of the Ninth International Conference on Intelligent Games and Simulation (GAMEON'2008), EUROSIS-ETI, Ghent University, Ghent, Belgium, 2008, pp. 69–76.
[73] S.C.J. Bakkes, P.H.M. Spronck, Rapidly adapting game AI, in: A. Nijholt, M. Pantic, M. Poel, H. Hondorp (Eds.), Proceedings of the 20th Belgian–Dutch Artificial Intelligence Conference (BNAIC 2008), University of Twente, Enschede, The Netherlands, 2008, pp. 9–16.
[74] E. Hudlicka, Affective computing for game design, in: J. Kienzle, H. Vangheluwe, C. Verbrugge (Eds.), Proceedings of the Fourth International North American Conference on Intelligent Games & Simulation, McGil University, Montreal, Canada, 2008, pp. 5–12.
[75] R. Mandryk, M. Atkins, A fuzzy physiological approach for continuously modeling emotion during interaction with play technologies, International Journal of Human–Computer Studies 6 (4) (2007) 329–347.
[76] C. Conati, H. Maclaren, Empirically building and evaluating a probabilistic model of user affect, User Modeling and User-Adapted Interaction 19 (2009) 267–303, doi:10.1007/s11257-009-9062-8. ISSN 0924-1868. Available from: <http://portal.acm.org/citation.cfm?id=1569345.1569357/>.
[77] J. Gratch, S. Marsella, Evaluating a computational model of emotion, Journal of Autonomous Agents and Multi-Agent Systems 11 (1) (2006) 23–43.
[78] S. Marsella, J. Gratch, P. Petta, A Blueprint for a Affective Computing: A Sourcebook and Manual, Chap. Computational Models of Emotion, Oxford University Press, Oxford, 2010. ISBN-13: 978-0199566709.
[79] C. Pedersen, J. Togelius, G. Yannakakis, Modeling player experience for content creation, IEEE Transactions on Computational Intelligence and AI in Games 1 (2) (2009) 121–133.
[80] G. Lankveld, S. Schreurs, P. Spronck, Psychologically verified player modelling, in: L. Breitlauch (Ed.), Proceedings of the 10th International Conference on Intelligent Games and Simulation (GAMEON'2009), EUROSIS-ETI, Ghent University, Ghent, Belgium, 2009, pp. 12–19.
[81] G. Yannakakis, J. Hallam, Real-time game adaptation for optimizing player satisfaction, IEEE Transactions on Computational Intelligence and AI in Games 1 (2) (2009) 121–133.
[82] G. Yannakakis, M. Maragoudakis, J. Hallam, Preference learning for cognitive modeling: a case study on entertainment preferences, IEEE Transactions on Systems, Man, and Cybernetics, Part A: Systems and Humans 39 (6) (2009) 1165–1175.
[83] G. Yannakakis, J. Togelius, Experience-driven procedural content generation, IEEE Transactions on Affective Computing 2 (3) (2011) 147–161.
[84] G. Van Lankveld, P.H.M. Spronck, H.J. Van den Herik, A. Arntz, Games as personality profiling tools, in: M. Preuss (Ed.), Proceedings of the 2011 IEEE Conference on Computational Intelligence in Games (CIG'11), 2011, pp. 197–202.
[85] J. Digman, Personality structure: emergence of the five-factor model, Annual Review of Psychology 41 (1) (1990) 417–440.
[86] R. McCrae, P. Costa Jr., Personality trait structure as a human universal, American Psychologist 52 (5) (1997) 509–516.
[87] M. Mateas, A. Stern, Procedural authorship: a case-study of the interactive drama Façade, in: Digital Arts and Culture: Digital Experience: Design, Aesthetics, Practice (DAC 2005), Copenhagen, Denmark, December 1–4, 2005.
[88] M.S. El-Nasr, Interaction, narrative, and drama: creating an adaptive interactive narrative using performance arts theories, Interaction Studies 8 (2) (2007) 209–240. ISSN 1572-0373.
[89] C. Crawford, Chris Crawford on Interactive Storytelling (New Riders Games), New Riders Games, 2004, ISBN: 0321278909.
[90] M. Riedl, A. Stern, Believable agents and intelligent story adaptation for interactive storytelling, in: S. Gobel, R. Malkewitz, I. Iurgel (Eds.), Technologies for interactive digital storytelling and entertainment, Lecture Notes in Computer Science, vol. 4326, Springer-Verlag, 2006, pp. 1–12.
[91] D. Thue, V. Bulitko, M. Spetch, E. Wasylishen, Interactive storytelling: a player modelling approach, in: Artificial Intelligence and Interactive Digital Entertainment Conference, Stanford, CA, 43–48, 2007.
[92] J. Togelius, G.N. Yannakakis, K.O. Stanley, C. Browne, Search-based procedural content generation: a taxonomy and survey, IEEE Transactions on Computational Intelligence and AI in Games 3 (3) (2011) 172–186. Special Issue on Procedural Content Generation.
[93] J. Dormans, S.C.J. Bakkes, Generating missions and spaces for adaptable play experiences, IEEE Transactions on Computational Intelligence and AI in Games 3 (3) (2011) 216–228. Special Issue on Procedural Content Generation.
[94] S.C.J. Bakkes, J. Dormans, Involving player experience in dynamically generated missions and game spaces, in: E. Ayesh (Ed.), Eleventh International Conference on Intelligent Games and Simulation (Game-On'2010), 2010, pp. 72–79.