A Model for Reliable Adaptive Game Intelligence

Pieter Spronck
Universiteit Maastricht
Institute for Knowledge and Agent Technology
P.O. Box 616, 6200 MD Maastricht, The Netherlands
{p.spronck}@cs.unimaas.nl

Abstract

Adaptive game AI aims at enhancing computer-controlled game-playing agents with the ability to self-correct mistakes, and with creativity in responding to new situations. Before game publishers will allow the use of adaptive game AI in their games, they must be convinced of its reliability. In this paper we introduce a model for Reliable Adaptive Game Intelligence (RAGI). The purpose of the model is to provide a conceptual framework for the implementation of reliable adaptive game AI. We discuss requirements for reliable adaptive game AI, the RAGI model's characteristics, and possible implementations of the model.

1 Introduction

The behaviour of computer-controlled agents in modern computer games is determined by so-called 'game AI'. For artificial intelligence research, game AI of complex modern games (henceforth called 'games') is a truly challenging application. We offer four arguments for this statement: (1) Games are widely available, thus subject to the scrutiny of hundreds of thousands of human players [Laird and van Lent, 2001; Sawyer, 2002]; (2) Games reflect the real world, and thus game AI may capture features of real-world behaviour [Sawyer, 2002; Graepel *et al.*, 2004]; (3) Games require human-like (realistic, believable) intelligence, and thus are ideally suited to pursue the fundamental goal of AI, i.e., to understand and develop systems with human-like capabilities [Laird and van Lent, 2001; Sawyer, 2002]; and (4) Games place highly-constricting requirements on implemented game AI solutions [Laird and van Lent, 2001; Nareyek, 2002; Charles and Livingstone, 2004; Spronck *et al.*, 2004b].

We define 'adaptive game AI' as game AI that employs unsupervised online learning ('online' meaning 'during gameplay'). Adaptive game AI has two main objectives, namely (1) to enhance the agents with the ability to learn from their mistakes, to avoid such mistakes in future play (self-correction), and (2) to enhance the agents with the ability to devise new behaviour in response to previously unconsidered situations, such as new tactics used by the human player (creativity). Although academic researchers have achieved successful results in their exploration of adaptive game AI in recent research (e.g., [Demasi and Cruz, 2002; Spronck *et al.*, 2004b; Graepel *et al.*, 2004]), game publishers are still reluctant to release games with online-learning capabilities [Funge, 2004]. Their main fear is that the agents learn inferior behaviour [Woodcock, 2002; Charles and Livingstone, 2004]. Therefore, the few games that contain online adaptation, only do so in a severely limited sense, in order to run as little risk as possible [Charles and Livingstone, 2004].

Regardless of the usefulness of adaptive game AI, to convince game publishers to allow it in a game, the *reliability* of the adaptive game AI should be guaranteed, even against human players that deliberately try to exploit the adaptation process to elicit inferior game AI. Reliability of adaptive game AI can be demonstrated by showing that it meets eight requirements [Spronck, 2005], which are discussed in Section 2. However, meeting the requirements is easier said than done, because they tend to be in conflict with each other.

In this paper, we propose a model for Reliable Adaptive Game Intelligence (RAGI). The purpose of the model is to provide a conceptual framework for the implementation of reliable adaptive game AI. The model makes explicit two concepts which, in our view, are necessary for the design of reliable adaptive game AI, namely a knowledge base, and an adaptive opponent model.

The outline of this paper is as follows. In Section 2 we discuss requirements for the creation of reliable adaptive game AI. In Section 3 we discuss domain knowledge and opponent models for adaptive game AI. The RAGI model is introduced in Section 4. In section 5 we argue that the proposed model is a suitable framework for implementing reliable adaptive game AI. Section 6 describes possible implementations of the model. Finally, Section 7 concludes and looks at future work.

2 Requirements for Reliability

We define 'reliable adaptive game AI' as adaptive game AI that meets the eight requirements for online learning of game AI specified by Spronck [2005], who indicated that adaptive game AI that meets these eight requirements will go a long way in convincing game publishers to adopt it. The eight requirements are divided into four computational requirements and four functional requirements. The computational requirements are necessities: failure of adaptive game AI to meet the computational requirements makes it useless in practice. The functional requirements are not so much necessities, as strong preferences by game developers: failure of adaptive game AI to meet the functional requirements means that game developers will be unwilling to include it in their games, even when it yields good results (e.g., improves the effectiveness of agent behaviour) and meets all four computational requirements.

The four computational requirements are the following.

Speed: Adaptive game AI must be computationally fast, since learning takes place during game-play [Laird and van Lent, 2001; Nareyek, 2002; Charles and Livingstone, 2004; Funge, 2004].

Effectiveness: Adaptive game AI must be effective during the whole learning process, to avoid it becoming inferior to manually-designed game AI, thus diminishing the entertainment value for the human player [Charles and Livingstone, 2004; Funge, 2004]. Usually, the occasional occurrence of non-challenging game AI is permissible, since the player will attribute an occasional easy win to luck.

Robustness: Adaptive game AI has to be robust with respect to the randomness inherent in most games [Chan *et al.*, 2004; Funge, 2004].

Efficiency: Adaptive game AI must be efficient with respect to the number of trials needed to achieve successful game AI, since in a single game, only a limited number of occurrences happen of a particular situation which the adaptive game AI attempts to learn successful behaviour for. Note that the level of intelligence of the adaptive game AI determines how many trials can still be considered efficient adaptation; on an operational level of intelligence (as in the work by Graepel *et al.* [2004]), usually many more trials are available for learning than on a tactical or strategic level of intelligence (as in the work by Spronck *et al.* [2004b] and the work by Ponsen *et al.* [2005]).

The four functional requirements are the following.

Clarity: Adaptive game AI must produce easily interpretable results, because game developers distrust learning techniques of which the results are hard to understand.

Variety: Adaptive game AI must produce a variety of different behaviours, because agents that exhibit predictable behaviour are less entertaining than agents that exhibit unpredictable behaviour.

Consistency: The average number of trials needed for adaptive game AI to produce successful results should have a high consistency, i.e., a low variance, to ensure that it is rare that learning in a game takes exceptionally long.

Scalability: Adaptive game AI must be able to scale the effectiveness of its results to match the playing skills of the human player [Lidén, 2004]. This last functional requirement may be considered optional: without it, adaptive game AI aims to be as strong as possible; with it, adaptive game AI aims to be an appropriate match for the human player.

We observe that there are conflicts between several of these requirements. For instance, the requirements of speed and efficiency are in conflict with the requirements of robustness and consistency, because in a non-deterministic learning environment, robustness and consistency are typically acquired by always basing the learning on several repetitions of each test, which is costly in computation time and required number of trials. Also, the requirement of effectiveness is in conflict with the requirement of variety, because, in general, enforced variations on game AI make it less effective.

The core problem for online learning, especially in a non-deterministic, complex environment, is finding the right balance between exploitation and exploration [Carmel and Markovitch, 1997]. During exploitation, adaptive game AI does not learn, but deploys its learned knowledge to elicit successful agent behaviour in the game. During exploration, adaptive game AI attempts to learn new behaviour. If there is insufficient exploration, the adaptive game AI learns slowly, and may remain stuck in a local or even a false optimum, and thus fails to meet the requirement of efficiency. If there is too much exploration, the adaptive game AI will often generate inferior agent behaviour, and thus fails to meet the requirement of effectiveness. A possible solution for this issue is to automatically tune the amount of exploration to the observed results of the agent behaviour: good results require a low degree of exploration, while unsatisfying results require a higher degree of exploration. However, note that due to the non-determinism of most game environments, unsatisfying results may be the effect of a string of chance runs, in which case these results preferably should not lead to a higher degree of exploration [Spronck, 2005].

3 Necessary Concepts for Adaptive Game AI

In the few published games that contain online adaptation, changes made by the adaptive game AI are almost always limited to updating a small number of in-game parameters, such as the agents' strength and health. In the rare cases where a published game allows an agent's behaviour to be influenced, it is either through supervised learning (i.e., the human player actively training the agent to exhibit certain behaviour, as in BLACK & WHITE [Evans, 2001]), or through choosing between a few pre-programmed behaviours, such as different formations of enemy groups. Most academics will hesitate to call this 'adaptive game AI', since the agents do not design new behaviour autonomously (professional game developers might disagree, but they interpret the term 'artificial intelligence' much broader than academics [Tomlinson, 2003]).

In academic research of adaptive game AI, it is typically implemented as a direct feedback loop (cf., [Demasi and Cruz, 2002; Bakkes, 2003; Graepel *et al.*, 2004; Spronck *et al.*, 2004b]). In a direct feedback loop for agent control in a game (illustrated in Figure 1), the agent interacts with a game world. The agent's actions are determined by game AI. The agent feeds the game AI with data on its current situation, and with the observed results of its actions. The game AI adapts by processing the observed results, and generates actions in response to the agent's current situation.

Adaptive game AI is necessarily based on two concepts.

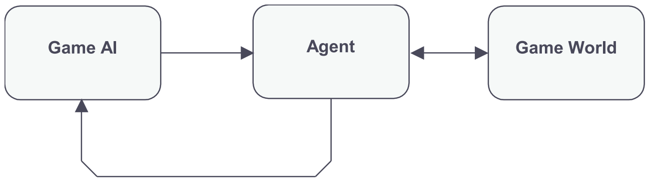

Figure 1: Game AI feedback loop.

The first concept is *domain knowledge* of the game environment. The reasoning behind this concept is that, to meet the four computational requirements, adaptive game AI must be of 'high performance'. According to Michalewicz and Fogel [2000], the two main factors of importance when attempting to achieve high performance for a learning mechanism are the exclusion of randomness and the addition of domain-specific knowledge. Since randomness is inherent in most games, it cannot be excluded. Therefore, it is imperative that the learning process is based on domain-specific knowledge [Manslow, 2002].

The second concept is an *opponent model*. The task of an opponent model is to understand and mimic the opponent's behaviour, to assist the game AI in choosing successful actions against this opponent. Without an opponent model, the game AI is unable to adapt adequately to human player behaviour.

The opponent model can be either *explicit* or *implicit*. An opponent model is explicit in game AI when a specification of the opponent's attributes exists separately from the decision-making process. An opponent model is implicit in game AI when the game AI is fine-tuned to a specific (type of) opponent, without the game AI actually referring that opponent's attributes [van den Herik *et al.*, 2005]. With an implicit opponent model, the adaptive game AI basically is a process that updates its opponent model by improving its decision making capabilities against particular human-player behaviour.

In most, if not all published research on adaptive game AI, the opponent model is implicit. However, in the comparable research field of adaptive multi-agent systems, Carmel and Markovitch [1997] have shown that adaptive agents that use an explicit opponent model are more effective than adaptive agents that use an implicit opponent model. Furthermore, the use of explicit opponent models is considered a necessary requirement for successful game-play in the research of such classical games as ROSHAMBO [Egnor, 2000] and POKER [Billings *et al.*, 2000], which have many features in common with modern commercial games. Therefore, we feel that there are sufficient reasons to suppose that an explicit opponent model is highly desired for adaptive game AI.

Figure 2 presents the feedback loop of Figure 1, enhanced with a data store of domain knowledge and an explicit opponent model. Examples are given, derived from a Computer RolePlaying Game (CRPG), of (1) a piece of domain knowledge, (2) an attribute of the opponent model, and (3) a rule of the game AI which takes the domain knowledge and opponent model into account. Note that, by simply removing the explicit opponent model from the figure, the opponent model becomes implicit in the game AI. Under the condition that we

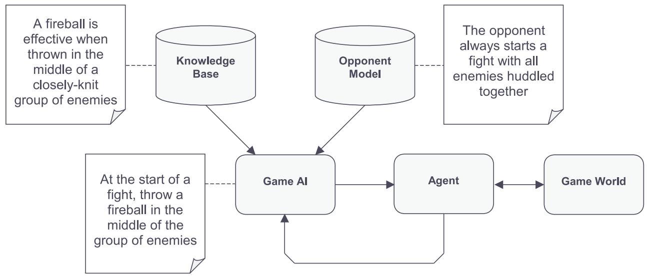

Figure 2: Game AI with domain knowledge and an explicit opponent model.

know that the game AI is effective, we can derive the explicit opponent model to a great extent by analysing the game AI.

Typically, opponent models of human players are not implemented statically, but are learned from observed behaviour [Carmel and Markovitch, 1997]. In contrast, domain knowledge for game AI is typically manually designed by game developers, usually by programming static game AI. However, as Ponsen *et al.* [2005] show, it is possible to generate domain knowledge automatically from game-play data.

In conclusion, we propose that successful adaptive game AI should incorporate a knowledge base of domain knowledge, and an adaptive opponent model of the human player (preferably explicit).

4 The Model

In this section we present our model for Reliable Adaptive Game Intelligence (RAGI). The RAGI model is illustrated in Figure 3. It is described below.

Basically, the RAGI model implements a feedback loop, as represented in Figure 1. The two differences between the feedback loop of Figure 1, and the RAGI model of Figure 3, are that (1) the RAGI model extends the feedback loop with explicit processing of observations distinguished from the game AI, and (2) the RAGI model also allows the use of game world attributes which are not directly observed by the agent (e.g., observations concerning different agents).

The RAGI model collects agent observations and game world observations, and extracts from those a 'case base'. The case base contains all observations relevant for the adaptive game AI, without redundancies, time-stamped, and structured in a standard format for easy access. A case consists of a description of a game-play situation, comprising selected features and actions undertaken by agents in that situation. All cases in the case base contain an identification of the particular agents involved, whether controlled by the computer or by a human player. In the case of multi-player games, we may expect the case base to expand rather fast. In the case of single-player games, the case base will probably expand slowly. Consequently, the RAGI model is most applicable to multi-player games, although under certain conditions it may be applicable to single-player games, too.

The case base has two uses. The first use is to build an opponent model. The second use is to generate domain knowledge.

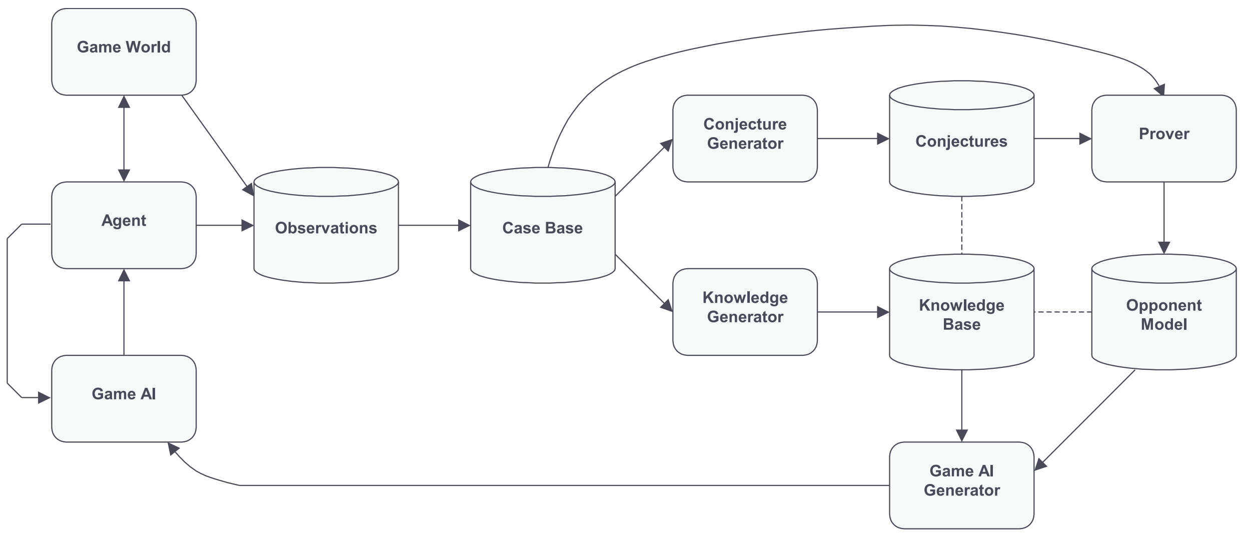

Figure 3: The RAGI model.

To build an opponent model, a 'conjecture generator' creates conjectures (i.e., statements and observations) on the way human players interact with the game world and with computer-controlled agents. These conjectures may not be generally applicable for all human players. However, a 'prover' (not to be confused with a theorem prover) selects those conjectures that might be of interest in building an opponent model of a specific human player, and uses the case base to attach a degree of confidence to the selected conjectures in this respect. Conjectures with a sufficiently high degree of confidence are stored as the opponent model of the human player. The representation of the conjectures depends on the application: for example, it might be in the form of first-order logic, or simply in the form of a collection of values for certain variables.

To generate domain knowledge, a 'knowledge generator', which can be considered a data mining process, analyses the case base and extracts relevant statements and rules. These are stored in a knowledge base. As with the opponent model, the case base is used to attach a degree of confidence to each statement in the knowledge base.

The opponent model and the knowledge base are used by a 'game AI generator' to create new game AI. Depending on the contents of the knowledge base, the game AI generator can be used to imitate the play of successful agents (for instance, those that are controlled by expert human players), or to design completely new tactics and strategies.

Through changes in the case base, changes might be caused in the opponent model and/or the knowledge base, which will automatically generate new game AI. For instance, if the human player changes his behaviour, the prover may assign a lower confidence to certain statements in the opponent model of this human player, which will influence the game AI generator to update the game AI.

Usually, there are connections between 'conjectures' and the 'knowledge base'. For instance, a conjecture might state that the human player has a preference for certain actions, while the knowledge base specifies a good defence against these actions. It is a good idea for implementations of the RAGI model to make these connections explicit. In Figure 3, this is represented by a dotted line between the conjectures and the knowledge base. Since the opponent model consists of a subset of the conjectures (enhanced with a degree of confidence), the same connections exist between the opponent model and the knowledge base.

5 Reliability

Why do we expect the RAGI model to be a good starting point for the creation of reliable adaptive game AI?

Besides the fact that the RAGI model encompasses an explicit opponent model and explicit domain knowledge, which we argued in Section 3 to be necessary for successful adaptive game AI, the RAGI model may meet the requirements specified in Section 2 as follows.

- The speed of the adaptive game AI relies, of course, on the speed of its components. In the past, authors have investigated speedy implementations of several of the components (e.g., for the knowledge generator [Ponsen *et al.*, 2005], and for the game AI generator [Spronck *et al.*, 2004b]). However, even if some components require too much processing time, since the model uses a case base the adaptive game AI may learn on a computer separate from the computer used to play the game, or in a separate thread, or on down-time of the game-playing computer (admittedly, in the last case this would amount to offline learning). This may allow the RAGI model to meet the requirement of speed, even when the processing itself is computationally intensive.
- Inferior behaviour on the part of any agent will automatically be translated into instances in the case base, that are processed into the opponent model or the knowledge base, to generate new game AI. This allows the RAGI model to meet the requirement of effectiveness.

- A lower limit to the required degree of confidence can be set so that the quality of the domain knowledge and of the opponent model is at an appropriate level. This allows the RAGI model to meet the requirement of robustness.
- The adaptive game AI does not learn only from experiences of the agent it controls, but also from the experiences of all other agents in the game world, whether controlled by the human or by the computer. It is even possible, in the case of single-player games, to collect cases from games played on different computers through the internet. Therefore, for the RAGI model the requirement of efficiency is simply not an issue.
- The use of an explicit opponent model and explicit domain knowledge helps the RAGI model in meeting the requirement of clarity.
- By varying over the domain knowledge used, the RAGI model meets the requirement of variety.
- Since the case base can be shared between all players of a game (whether in single-player or multi-player mode), all instances of the adaptive game AI learn at the same rate. This allows the RAGI model to meet the requirement of consistency.
- By using statements in the opponent model or the knowledge base with a lower confidence, or by excluding high-confidence domain knowledge, the generated game AI may function at an arbitrary level of skill [Spronck *et al.*, 2004a]. This allows the RAGI model to meet the requirement of scalability.

Depending on the implementation of the various processes, arguably the RAGI model may be too complex, and thus too computationally intensive, to be used for online learning. This issue holds in particular for single-player games, when only a single computer is available. It has less impact on multi-player games, where the case base is preferably situated on the game server, and domain knowledge and conjectures are generated centrally, so that they can be shared amongst all players. On the client computer, at maximum only two processes need to be executed, namely (1) the maintenance of the opponent model of the human player that uses the computer, and (2) the generation of new game AI on the basis of the opponent model and the centralised knowledge base. In general, opponent models do not change quickly. Furthermore, if connections between the conjectures and the domain knowledge (i.e., the dotted lines in Figure 3) are maintained centrally, the generation of new game AI can be fast.

6 Implementations

When implementing adaptive game AI according to the RAGI model, many findings of previous research can be incorporated. For instance, Spronck *et al.* [2004b] designed 'dynamic scripting', an adaptive-game-AI technique that makes use of a rulebase, which is equivalent to a knowledge base with domain knowledge. Ponsen *et al.* [2005]) investigated the automatic generation of domain knowledge for adaptive game AI, i.e., a knowledge generator. There is plenty of research available on the generation of opponent models (cf., [Fürnkranz, 1996; Carmel and Markovitch, 1997; Davison and Hirsh, 1998; Billings *et al.*, 2000; Egnor, 2000]), even in the area of commercial games (cf., [Alexander, 2002; McGlinchey, 2003]).

An interesting aspect of the RAGI model is that it can be implemented in stages. An easy implementation would use a static data store of manually-designed conjectures, and a static knowledge base of manually-designed knowledge, with the connections between the conjectures and the knowledge also programmed manually. Only the 'prover' would need to use the case base to constitute the opponent model, by selecting the conjectures that are most likely to be true. Depending on how the knowledge is formulated, the game AI generator would be trivial, because it only would need to select from the knowledge that is connected with the opponent model.

At a moderate level of difficulty for the implementation of the model, the connections between the conjectures and the knowledge base could be generated automatically. And at a high level of difficulty, a knowledge generator and conjecture generator could be implemented.

The possibility to start an implementation of the RAGI model at an easy level, gradually expanding it to become more complex, makes the model ideal for explorative research. The RAGI model can also be combined easily with the TIELT architecture [Aha and Molineaux, 2004], since TIELT has been designed to work with a task model (i.e., game AI), a player model (i.e., an opponent model), and a game model (i.e., a case base).

7 Conclusions and Future Work

In this paper we argued that reliable adaptive game AI needs to meet eight requirements, namely the requirements of (1) speed, (2) effectiveness, (3) robustness, (4) efficiency, (5) clarity, (6) variety, (7) consistency, and (8) scalability. Furthermore, we argued that successful adaptive game AI is necessarily based on domain knowledge and on an adaptive opponent model. We proposed a model for Reliable Adaptive Game Intelligence (RAGI), that is indeed based on domain knowledge and an explicit adaptive opponent model, and that may meet the eight specified requirements (at least for multi-player games).

Of course, the RAGI model must still undergo the proof of the pudding. In future work, we intend to structure our research into adaptive game AI around the RAGI model, and to explore to what extent elements of the model can learn while the adaptive game AI as a whole remains a reliable process.

Acknowledgments

This research is supported by a grant from the Dutch Organisation for Scientific Research (NWO grant No. 612.066.406).

References

[Aha and Molineaux, 2004] D.W. Aha and M. Molineaux. Integrating learning in interactive gaming simulators. In D. Fu, S. Henke, and J. Orkin, editors, *Proceedings of the AAAI-04 Workshop on Challenges in Game Artificial Intelligence*, pages 49–53, 2004. AAAI Press.

[Alexander, 2002] T. Alexander. GoCap: Game observation capture. In S. Rabin, editor, *AI Game Programming Wisdom*, pages 579–585, 2002. Charles River Media, Inc.

[Bakkes, 2003] S. Bakkes. *Learning to Play as a Team: Designing an Adaptive Mechanism for Team-Oriented Artificial Intelligence*. M.Sc. thesis. Universiteit Maastricht, Maastricht, The Netherlands, 2003.

[Billings *et al.*, 2000] D. Billings, A. Davidson, J. Schaeffer, and S. Szafron. The challenge of poker. *Artificial Intelligence*, 134(1–2):201–240, 2000.

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

[Chan *et al.*, 2004] B. Chan, J. Denzinger, D. Gates, K. Loose, and J. Buchanan. Evolutionary behavior testing of commercial computer games. In *Proceedings of the 2004 IEEE Congress on Evolutionary Computation*, pages 125–132, 2004. IEEE Press.

[Charles and Livingstone, 2004] D. Charles and D. Livingstone. AI: The missing link in game interface design. In M. Rauterberg, editor, *Entertainment Computing – ICEC 2004*, Lecture Notes in Computer Science 3166, pages 351–354, 2004. Springer-Verlag.

[Davison and Hirsh, 1998] B.D. Davison and H. Hirsh. Predicting sequences of user actions. In *Predicting the Future: AI Approaches to Time-Series Problems*, pages 5–12, 1998. AAAI Press. Proceedings of AAAI-98/ICML-98 Workshop, published as Technical Report WS-98-07.

[Demasi and Cruz, 2002] P. Demasi and A.J. de O. Cruz. Online coevolution for action games. *International Journal of Intelligent Games and Simulation*, 2(2):80–88, 2002.

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

[Evans, 2001] R. Evans. The future of game AI: A personal view. *Game Developer Magazine*, 8(8):46–49, 2001.

[Funge, 2004] J.D. Funge. *Artificial Intelligence for Computer Games*. A K Peters, Ltd., Wellesley, MA, 2004.

[Fürnkranz, 1996] J. Fürnkranz. Machine learning in computer chess: The next generation. *ICCA Journal*, 19(3):147–161, 1996.

[Graepel *et al.*, 2004] T. Graepel, R. Herbrich, and J. Gold. Learning to fight. In Q. Mehdi, N.E. Gough, S. Natkin, and D. Al-Dabass, editors, *Computer Games: Artificial Intelligence, Design and Education (CGAIDE 2004)*, pages 193–200, 2004. University of Wolverhampton.

[Laird and van Lent, 2001] J.E. Laird and M. van Lent. Human-level's AI killer application: Interactive computer games. *Artificial Intelligence Magazine*, 22(2):15–26, 2001.

[Lidén, 2004] L. Lidén. Artificial stupidity: The art of making intentional mistakes. In S. Rabin, editor, *AI Game Programming Wisdom 2*, pages 41–48, 2004. Charles River Media, Inc.

[Manslow, 2002] J. Manslow. Learning and adaptation. In S. Rabin, editor, *AI Game Programming Wisdom*, pages 557–566, 2002. Charles River Media, Inc.

[McGlinchey, 2003] S.J. McGlinchey. Learning of AI players from game observation data. In Q. Mehdi, N. Gough, and S. Natkin, editors, *Proceedings of the 4th International Conference on Intelligent Games and Simulation (GAME-ON 2003)*, pages 106–110, 2003. EUROSIS.

[Michalewicz and Fogel, 2000] Z. Michalewicz and D.B. Fogel. *How To Solve It: Modern Heuristics*. Springer-Verlag, Berlin, Germany, 2000.

[Nareyek, 2002] A. Nareyek. Intelligent agents for computer games. In T.A. Marsland and I. Frank, editors, *Computers and Games, Second International Conference, CG 2000*, volume 2063 of *Lecture Notes in Computer Science*, pages 414–422, 2002. Springer-Verlag.

[Ponsen *et al.*, 2005] M.J.V. Ponsen, H. Muñoz-Avila, P. Spronck, and D.W. Aha. Acquiring adaptive real-time strategy game opponents using evolutionary learning. In *Proceedings of the IAAI-05*, 2005. Accepted for publication.

[Sawyer, 2002] B. Sawyer. *Serious Games: Improving Public Policy through Game-based Learning and Simulation*. Foresight & Governance Project, Woodrow Wilson International Center for Scolars, Washington, DC, 2002.

[Spronck *et al.*, 2004a] P.H.M. Spronck, I.G. Sprinkhuizen-Kuyper, and E.O. Postma. Difficulty scaling of game AI. In A. El Rhalibi and D. Van Welden, editors, *GAME-ON 2004 5th International Conference on Intelligent Games and Simulation*, pages 33–37, 2004. EUROSIS.

[Spronck *et al.*, 2004b] P.H.M. Spronck, I.G. Sprinkhuizen-Kuyper, and E.O. Postma. Online adaptation of game opponent AI with dynamic scripting. *International Journal of Intelligent Games and Simulation*, 3(1):45–53, 2004.

[Spronck, 2005] Pieter Spronck. *Adaptive Game AI*. Ph.D. thesis, Universiteit Maastricht. Universitaire Pers Maastricht, Maastricht, The Netherlands, 2005.

[Tomlinson, 2003] S.L. Tomlinson. Working at thinking about playing or a year in the life of a games AI programmer. In Q. Mehdi, N. Gough, and S. Natkin, editors, *Proceedings of the 4th International Conference on Intelligent Games and Simulation (GAME-ON 2003)*, pages 5–12, 2003. EUROSIS.

[van den Herik *et al.*, 2005] H.J. van den Herik, H.H.L.M. Donkers, and P.H.M. Spronck. Opponent modelling and commercial games. In G. Kendall and S. Lucas, editors, *IEEE 2005 Symposium on Computational Intelligence and Games*, pages 15–25, 2005.

[Woodcock, 2002] S. Woodcock. AI roundtable moderator's report, 2002. www.gameai.com/cgdc02notes.html.