Agent Architecture in Social Games
-- The Implementation of Subsumption Architecture in Diplomacy

Aleksander Krzywinski, Weiqin Chen and Arne Helgesen

Department of Information Science and Media Studies, University of Bergen
Fosswinckelsgate 6 - 5007 Bergen, Norway
aleksander.krzywinski@infomedia.uib.no, weiqin.chen@infomedia.uib.no and arne.helgesen@odfjell.com

Abstract

Social games are challenging for AI research because they involve not only the mechanism of game (taking actions, planning, etc), but also the social aspects including communicating, cooperating and reaching agreements. Some social games also require cunning, duplicity, or bad faith. In this paper we present the implementation of a prototype that utilizes the properties of the subsumption architecture to provide an interesting computer opponent in a social board game -- Diplomacy. The evaluation result indicates that the subsumption architecture is appropriate for social games.

Introduction

Diplomacy is a strategic board game with strong emphasis on cooperation and strategizing with opponents, where alliances shifts quickly while players try to position themselves in the diplomatic struggle for ultimate victory (Calhamer, 1971). Although the underlying game mechanics are relatively simple and enjoyable, it is the Machiavellian contest that is the true allure of Diplomacy. Players are required to make deals and plan together with their opponents – creating and dissolving alliances from round to round. No agreements are binding in Diplomacy, and the simultaneously revealed orders ensure that players can never fully thrust their peers to do as agreed.

Diplomacy is played with up to seven players – controlling one European nation each – on a map of Europe anno 1900. The map is divided into 75 provinces/areas where 34 of them are supply centers. Occupying these allows players to raise and maintain an equal number of units – either armies or fleets. The game is played over a number of game years each containing five distinct phases, starting in 1901. The phases come in three flavors; order, movement and build. The order phase ends with all players writing down the orders they want to give to their units in secrecy. To avoid errors the orders has to follow a strict format, and any errors lead to the unit in question staying stationary. Four kinds of orders are possible: hold, move, support and convoy (the latter is only applicable for fleets).

All orders are revealed simultaneously in the movement phase and any conflicting orders are resolved according to the rules. No dice or other forms of randomness are used to determine the effects of the orders, and units prevail based on superior numbers alone, with defenders winning draws. At the end of the movement phase all orders have been resolved, and the game board represents the new game state.

Recreating the intense social experience with computer opponents is extremely difficult - even disregarding experiences such as finally beating a friend who has won too many games, or having a great gaming story to tell over lunch the next day - and hence there lies a real challenge in creating a computer opponent that plays an enjoyable game of Diplomacy, and not only is a sparring partner for the technical aspects of the game. The computer needs to at least be able to mimic the social interactions of a human player.

Diplomacy is what we define as a social game – a game where the core game-play requires the players to interact with a social environment, in addition to the game environment, in order to successfully compete in the game - and social games are a challenging domain for artificial intelligence research, not only because of the disparate nature of these two environments, but also because of the little emphasis put on social aspects in earlier AI research. Wooldridge (2002, p. 10) states that

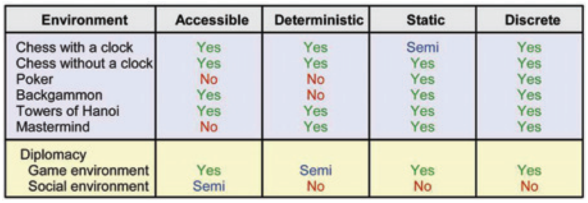
> Classical AI has largely ignored the social aspects of agency. […] part of what makes us unique as species on Earth is not simply our undoubted ability to learn and solve problem, but our ability to communicate, cooperate, and reach agreements with our peers.

Although Game AI takes different approaches from traditional AI and aims at intelligent behaviors and good game play, very little active research is directly pursuing to enable NPCs (agents) to exhibit human social skills required in strategy games (Laird, 2001). In this paper we present an agent architecture that is intended to enable game-playing agents to handle both the social and game environments - thus allowing the agents to be challenging opponents while retaining the appeal of the political power-struggle.

The Social and Game Environments of Diplomacy

An analysis of the game based on Russel and Norvig (1995) gives clear indications that the two environments need two separate approaches in architecture, and merging these two into one unified structure is the central property of the devised architecture.

The game environment is, despite an elaborate rule-set and players acting simultaneously, rather simple. As can be seen in the summary of the environmental properties in figure 1, the game environment is fully accessible, semi-deterministic (the rules are totally deterministic and none of the arbitration in the game is based on chance, but as all players act simultaneously a player cannot be certain of the outcome of his moves before the moves of all players are revealed - thus the semi-deterministic label), static and discrete. Except for the accessibility, the game environment shows all the properties of a simple environment, and it was the accessibility that was most challenging when creating the game-playing part of the prototype.



Environment	Accessible	Deterministic	Static	Discrete
Chess with a clock	Yes	Yes	Semi	Yes
Chess without a clock	Yes	Yes	Yes	Yes
Poker	No	No	Yes	Yes
Backgammon	Yes	No	Yes	Yes
Towers of Hanoi	Yes	Yes	Yes	Yes
Mastermind	No	Yes	Yes	Yes
Diplomacy				
Game environment	Yes	Semi	Yes	Yes
Social environment	Semi	No	No	No

Figure 1: Examples of Environments and their Characteristics including Diplomacy

Below is a more thorough analysis of the social environment as it is more special to Diplomacy.

Accessible versus inaccessible. The social environment of Diplomacy is partially accessible. It is not possible to know what the other players will do - one cannot read the minds of the other players. It is also not possible to access the deals between the players, unless they are stated openly (something that is very rare). The intentions of the other players are guesswork based upon: The proposals, previous actions, non-verbal communication and the game board situation.

Deterministic versus non-deterministic. The social environment of the Diplomacy is non-deterministic. Devious opponents break deals with impunity and shaking hands is a mere cosmetic activity. A player can only be sure of the outcome of actions when no other player can intervene. It is a common strategy to break alliances at the most appropriate moment taking the partner by surprise.

Static versus dynamic. The social environment is dynamic. Premises from previous rounds may suddenly be invalid as a result of decisions of the other players. Some diplomatic constellations might make the social environment static, but only for short periods.

Discrete versus continuous. The social environment is as continuous as any other social setting. The actions of players are not parceled out in predetermined options and agreements can be partly or completely ignored.

Summary

As shown in figure 1 the two environments in Diplomacy are more or less opposites regarding complexity. According to (Wooldridge, 2002) the most complex general class of environments would be noted with a 'no' in all the four columns in figure 1.[1]

We conclude that the game environment is accessible, semi-deterministic, static, and discrete. Hence it is a simple environment according to Russell and Norvig, and being governed by easily formalized rules it lends itself favorably to traditional AI techniques for state-space search and move arbitration based on heuristics. The social environment of Diplomacy is semi-accessible, non-deterministic, dynamic and continuous. Russell and Norvig label this a complex environment, thus being less suited for the aforementioned AI-techniques. To tackle such an environment we chose to implement a version of the Subsumption architecture (Brooks, 1985), both leveraging the properties of emergent intelligence provided by that architecture and allowing us to build the social part of the prototype incrementally. The next section will detail how the prototype agent is internally structured and what social modules were implemented.

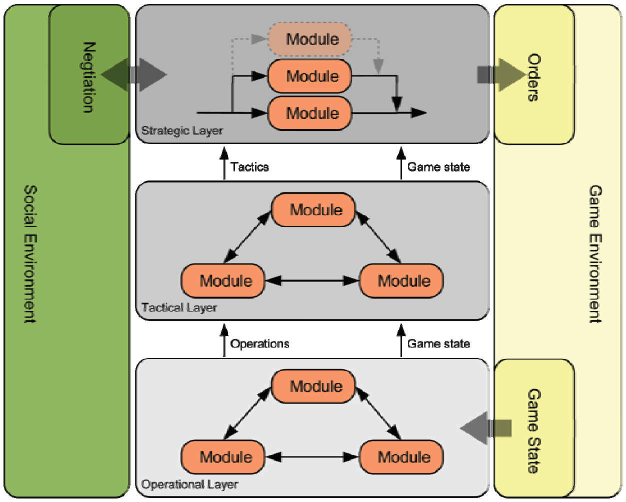


Figure 2: Overview of the architecture

Agent Architecture

The main structure is typical for three-layered architecture (Gat, 1998) where we have one layer responsible for each

[1] Another important property that plays a part in determining the complexity is the nature of the interactions between the agent and the environment - for instance the aspect of real time. Diplomacy is a turn-based game, where time is not an issue and an agent can deliberate for as long as required.

individual unit and to a limited extent the dependencies[2] between the action of these units, a second layer combining these individual actions into viable tactics with clear goals and weighted by objective importance (i.e. disregarding the diplomatic situation all together) and the final layer handles the communication with the other players and adjusts the weights of the tactics according to changes in the diplomatic status (figure 2).

Diplomacy is a turn based game. The operational and tactical layers are invoked once each turn, acting on the new game state that resulted from the previous round, and the strategic layer is active throughout the whole game session. Thus the agent is driven by the periodical updates of the game state, while still maintaining continuous diplomatic interaction with the opponents.

Operational Layer

In our implementation all possible actions are explored and reported from the lower layer. This is a relatively straightforward process as each unit a player controls can be given one out of four orders. Hold (stay at the current location) and move (relocate to an adjacent territory - which also constitutes as an attack if the territory is controlled by another player) are the simplest. Each unit is provided with a hold order, and a move order to all adjacent territories. A unit may be issued a support order signifying that it lends its strength to another unit performing any other order. Since it makes no sense to support orders that doesn't exist, support orders are created last. Finally all fleets that are in the open ocean (as opposed to being in coastal waters) may be issued orders to convoy armies from a land territory to another. This convoying is the only challenging part of this algorithm.

Tactical Layer

The end result of the processing in the operational layer is a list of all the possible actions that can be taken for every unit on the board including the units belonging to opponents. The first task of the tactical layer is to combine the actions from this list into tactics (i.e. a set of one action for each unit the agent controls), and this is done for all combinations of actions possible. With potentially billions of combinations the state-space search is pruned whenever an illegal (according to the game rules) or nonsensical (according to game experience) combinations are detected. For instance it is illegal for two pieces to swap position directly by moving into each others territories. Still the state-space would be too large for all possibilities to be completely exhausted, and hence we continuously apply the heuristics mentioned below, to prune potentially fruitless tactics.

Different heuristics are used in an effort to weigh the resulting tactics according to how good (i.e. how much the agent would gain in game terms if the orders were executed) they are. About half of these are static as they consider properties of the game board that are fixed regardless of the state of the game[3], while the rest take the dynamic situation on the game board into consideration (i.e. one will not be rewarded with a high weight for reoccupying already owned territory - except if said territory is threatened of course). The result is a ordered list of all tactics based on the objectively best modus operandi, disregarding the actions of other players.

This is however just partially the correct evaluation of the tactics. An important aspect of Diplomacy, the simultaneous execution of all players' orders, makes predicting the outcome of a given set of actions difficult. In games where each player's actions are carried out separately and in turn, the probability of success can easily be calculated, but when the orders of an agent are refereed concurrently with the orders of six other players the task of predicting the outcome beforehand is hard. The weight given to the tactic by the previously mentioned heuristics is therefore merely a potential value, which might be achieved under perfect circumstances, thus representing the gains offered to the agent by the set of orders, totally disregarding the probability for the circumstances to occur. This is the most complex issue of the game environment, and also the most difficult to tackle as predicted by the analysis of that environment.

As a consequence, the tactical layer applies a final set of heuristics to the tactics before they are presented to the strategic layer. These try to factor in the adverse conditions preventing a tactic from reaching its full potential and giving the tactics a new weight more accurately describing the actual value of the tactic. The discrepancy between the potential and actual weight is later used by the strategic layer to decide where the agent may pursuit diplomatic deals to increase the actual value of potentially good tactics.

Strategic Layer

The third and last layer of our architecture is responsible for communicating with the other players, and based on this diplomatic activity and the weighted tactics from the previous two layers, selects the appropriate tactic for the current round. As explained earlier this layer is organized according to the Subsumption Architecture (Brooks, 1986) and in our prototype it consists of four relatively simple modules. The result is a more flexible structure, as the modules impact each other in a non-linear manner and new modules can be added without breaking the former functionality. The internal wiring of the four modules are

[2] Convoying over a stretch of sea is an example of a movement/attack action by one unit that is totally dependent on at least another unit being given a convoy order.

[3] E.g. high values for occupying territories with support centers, and lower values for territories to the edges of the map or with few bordering territories.

shown in figure 3, and their basic tasks and inner workings are outlined below.

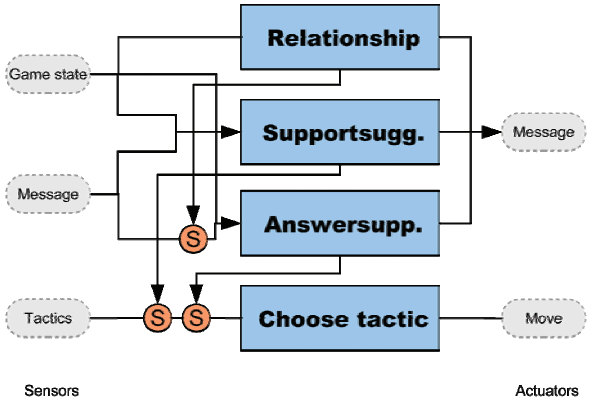


Figure 3: The wiring of the subsumption stack

Choose Tactic. The module is responsible for selecting the next action of the agent in the game environment. It has a list of sorted tactics to choose from and the algorithm within the module reduces the abundance of options down to one single action. It is supposed to be the simplest, functioning module of the strategic layer and hence it simply chooses one of the top weighted tactics at random. Random selection is used because we want the actions of the agent to be unpredictable. If the agent always chose the same action in a given situation it would be a predictable opponent and predictability is the opposite of engaging and fun which are what we aim for.

Answer Support Requests. When an opponent sends the agent a request for support this module processes the request and responds to the player. The module considers whether giving this support will significantly reduce the weights of the top tactics, and if it can be granted without sacrificing too much value it will be accepted. The module takes no other considerations for the diplomatic environment, and will gladly accept requests from friends and foes with the same attitude.

After accepting a suggested support the module must make sure that the agent complies with its promises. The next time a list of tactics is sent to the Choose Tactics module the list is suppressed by this module and the agreed upon support-operations substitutes their counter-parts in all the tactics. This way Choose Tactic module may never select a tactic where the promised supports are not present - provided that the Answer Support Requests module is allowed to operate uninhibited of course.

Support Suggester. This module is responsible for asking opponents for supports that is considered to be beneficial for the agent. This is a complex task. The module uses both the current game state and the list of weighted tactics as an input for these decisions. Potentially beneficial tactics (tactics with a low factual value and a high potential value) are located in the list, and the game state is searched for opponent units that can support. A few candidates are selected at random and the agent starts to pursue them in order of weight. If any of the requests get positive answers the tactics that require them have their actual value increased to reflect that the conditions for their success has become more favorable. It is important to note that even though support has been promised by another player (if they even keep their promise) it is not certain that the agent will choose a tactic that needs that support. It is also important to mention that this module, like the previous two, does not take into consideration the actual relationship with the other players before deciding what to do. This is wholly the responsibility of the presently last implemented module of the agent prototype.

Relationship. This module suppresses and inhibits lower modules in the hierarchy based on the perceived relationship between the agent and the other players in the game. The module uses six finite state machines to represent the relationship to each of the other players. At the beginning of each round the actions of the opponents are evaluated and their relationship status is changed accordingly. During the rest of the round the module suppresses the other modules and alters their behavior according to the status of the relevant relationships. In respect to the incoming requests for support for instance, only those from players regarded as friends are let through unconditionally, but it gets increasingly more difficult the worse the relationship is.

Summary

The heuristics of the tactical layer is codified human knowledge regarding the tactics of Diplomacy, and provides a game mechanical basis for the subsumption layer to interact with the other players and finally choose an appropriate tactic for the present round. Each of the described approaches of the subsumption sub-system uses relatively simple algorithms when viewed in isolation, but the structure of the layer as a whole leads to more sophisticated behaviors.

Evaluation of the Architecture

We set up an evaluation of our architecture to explore its performance, and determine if it is a suitable agent architecture for social games. Ferguson (1992, p. 151) argues that many of the characteristics, both positive and negative, of any particular agent architecture, only become evident when experimental evaluation is performed. Architecture can not be directly tested – it can only be tested through experimenting with an implementation. Therefore we evaluate the architecture by testing the agent we created based on the architecture – thus testing the architecture indirectly.

Evaluation design

The evaluation aims to answer two questions:
1. How does the agent perform?
2. How does the agent behave?

The two questions relates directly to the two environments the agent operates in. The two environments in social games is difficult to evaluate separately as they are so intertwined in the overall experience. According to Scott (2002), the performance of the agent is related to the deterministic environment and the behavior is related to the non-deterministic. By answering the preceding two questions the advantageousness of our architecture can be evaluated with respect to the two environments. We created a test where the computer would play against six human opponents, where the goal was not for the agent to win, but for the agent to be indistinguishable from a human player.

Seven computers were used in the test; one for the server running the game and the agent, and one for each of the human participants. A computer lab was arranged so that the six screens for the participants were facing away from each other, thus preventing the participants from cheating by looking at their opponents. The server was set up in a different room.

Every participant received a questionnaire that explained the purpose of the test and kindly asked them to answer two questions about the agent:

1. Which country was agent controlled during the last game?
2. Why have you come to that conclusion?

Therefore the participants were aware of the purpose of the test before we started the first session, and we anticipated that they then would use any means to find the agent. We could have masked the real intentions of the experiment for them, but we wanted to do more than one session and to prevent the first session from having different preconditions than the others, we exposed the questions at the start.

All six participants had played the game earlier, but our digital implementation of the game engine and game interface was a new experience for them. One severely limiting factor in the interaction was the options in the menu for diplomatic messages. We had only implemented a limited set of messages when the experiment was conducted, and this was the main complaint from the players.

An important aspect of the test was that the human participants were not allowed to communicate through any other means than the in-game interface. Otherwise it would be all too easy to establish which player was the agent. This limitation was strictly enforced by the facilitators.

A nation for the agent was randomly picked in each of the two sessions we had time for. The first nation was France and the second was Austria. France is one of the most isolated nations, a player do not need to make diplomatic talks in the first period. Austria is situated in the middle of Europe with potential enemies or friends on all sides. If not played wisely Austria will be eliminated early in the game. Both nations are interesting challenges as they require different strategies to play.

The first game lasted from 1900 to 1903 and the second lasted from 1900 to 1904. The second game was started after a short break with the same participants but with new nations.

Analysis and Findings

Performance. Figure 4 presents the results of the agent in the two test games. There is limited amount of information to infer from the data, though they show two properties of the performance of the agent. The agent conquers centers and the strategic position of the agent does not diminish in strength. The second property is the stagnation of the agent when it has reached a certain state. The number of conquered centers is above average in the first game, but not in the second.

		Controlled	
Session	Year	Centers	Provinces
1 (France)	1901	3	6
	1902	5	8
	1903	5	8
	1904	5	9
2 (Austria)	1901	3	6
	1902	4	6
	1903	4	6
	1904	4	6
	1905	4	6

Figure 4 The agent's results in the sessions.

Behavior. After each game the human participants were asked to answer which country the agent controlled and explain their answer. The answers show how the agent behaved in comparison to the human players, and the results are summed up in figure 5 and 6.

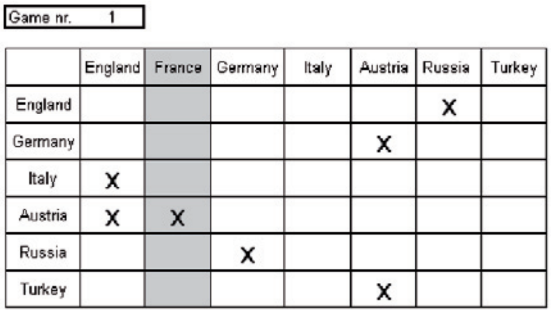
The agent played France in the first game. Only Austria guessed correct, but they also believed that it could be England or both of them. Nobody believed it was Italy or Turkey, but every other country was suggested. The agent played Austria in the second game. Two participants guessed correct. Nobody thought it was Russia, Turkey or England.

Even though every participant guessed at least one country at each round, some of them stressed that they were not sure or that they simply was guessing. One player explained:

> I guess the agent played France, due to a quite non-aggressive style. Only gut feeling, no real proof. All in all, my suggestions for AI are simply without any true proof.

Naturally a player would also focus on his nearest opponents, and not on the opponents that are on the other side of the board. A player stated that

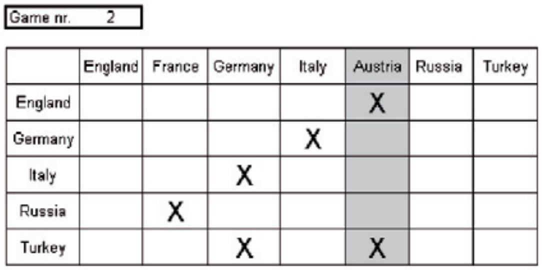
> I observed only the countries in my vicinity – Turkey, Italy and Russia – and I don't believe these where agent.



Game nr. 1

	England	France	Germany	Italy	Austria	Russia	Turkey
England						X	
Germany					X		
Italy	X						
Austria	X	X					
Russia			X				
Turkey					X		

Figure 5: The answers regarding the country of the agent in the first game



Game nr. 2

	England	France	Germany	Italy	Austria	Russia	Turkey
England					X		
Germany				X			
Italy			X				
Russia		X					
Turkey			X		X		

Figure 6 The answers regarding the country of the agent in the second game

From one perspective the overall result regarding the behavior of the agent is very positive. The agent did not easily reveal itself by doing stupid moves or ask for very unlikely deals. None of the human players complained about the deals that the agent asked about. One comment is particularly interesting:

> I don't think it was Austria because the way they negotiated the deals. For instance a proposed deal was rejected and then done anyway. It turned out to be a very clever move.

Though the apparent inability for most of the participants to guess the nation of the agent sounds promising for the evaluation, it is not necessarily the merit of the behavior of the agent. The interface puts constraints on the possible communications between players limiting the chance to discover the agent through diplomatic talks.

One subject explained:

> Italy was managed poorly. Though I did not make many deals with them. But I'm not sure as there were few choices in the [diplomatic] menu

Despite the limitations in the diplomatic options the players enjoyed the two games, and they were excited when the identity of the agent was revealed.

Conclusion and Future Work

In this paper we discussed the properties of social games and the application of the subsumption architecture for the computer implementation of a social game -- Diplomacy. To handle the different complexities of the two environments of social games, we created a three layered architecture with subsumption governing the topmost layer. In the discussion of the implemented prototype we detailed the three layers of our approach and described how they inter-operate.

In the evaluation of said structure we tested the agent, thus indirectly explored the suitability of the subsumption architecture in social games, by exposing the agent to a very limited Turing test. The findings show - within the relatively severe constraints of the digital interface - that the agent performed and behaved like a human player, indicating that the subsumption architecture could be an appropriate structure for handling the social interaction in social games.

We are currently in the process of evaluating the agent further, and will add more diplomatic options to its repertoire to make the game more diverse and challenging.

In order to further improve the game experience, we are currently implementing a personality module in the agent architecture. With this module, the agent can behave with different personalities (e.g. aggressive, taking risks) when playing the game. Using the extensibility of subsumption architecture the addition of even further modules is straightforward. Though, after sufficient modules are included, adding new ones and wiring them properly becomes akin to a black art (Brooks, 1990), and will severely reduce the simplicity of our implementation.

References

Amir, E. and Maynard-Zhang, P. 2004. *Logic-based subsumption architecture*. Artificial Intelligence, 153 (1-2) 167-237.

Brooks, R. A. 1985. *A Robust Layered Control System for a Mobile Robot*. IEEE Journal of Robotics and Automation, 2 (1) 14–23.

Brooks, R. A. 1990. *Challenges for Complete Creature Architectures.* In Animals to Animats: Proceedings of the First International Conference on Simulation of Adaptive Behavior (SAB90), MIT Press.

Calhamer, A. 1971. *The Invention of Diplomacy. Games & Puzzles*, 21(1).

Ferguson, I. A. 1992. *Touring Machines - Autonomous Agents with Attitudes*. IEEE Computer, 25 (5).

Gat, E. 1998*. Three-layer architectures*. In Kortenkamp, D., Bonasso, R. P., and Murphy, R., eds, Artificial Intelligence and Mobile Robots: Case Studies of Successful Robot Systems, Cambridge, MA.:MIT Press, 195-210.

Laird J. 2001. *Human-Level AI's Killer Application* Interactive Computer Games. AI Magazine, 22(2) 15-15.

Russell, S., and Norvig, P. eds. 1995. *Artificial Intelligence - A modern approach.* Prentice Hall. Upper Saddle River: New Jersey: Pearson Education, Inc.

Scott, B. 2002. *Architecting a Game AI*. In Rabin, S., eds, AI Game Programming Wisdom 3. Charles River Media.

Wooldridge, M. 2002. *An introduction to MultiAgent Systems*, Chichester, England: John Wiley & Sons.