Opponent modelling for case-based adaptive game AI

Alexander Römelt

University of Heidelberg

Seminar: Artificial Intelligence for Games

July 18, 2019

Agenda

Definition

”In general, an opponent model is an abstracted description of a player or a players behaviour in a game” Herik, Donkers, and P. H. M. Spronck n.d.

Build a model of the opponent player and utilize it for actual play

$\hookrightarrow$ Goal: Adapt to opponent and exploit his weaknesses!

Example: rock-paper-scissors

Possible other applications of opponent modelling:

- Military
- Robotics industry
- Understanding and representation of human models

Classic Games

General role of opponent modelling in classic games:

- Apply search techniques to find possible actions of the opponent and construct a model
- Guide the search process towards improved results

Short history:

- 1970s:
  - contempt factor in chess programs
  - chance of performing a non-rational action
  - rudimentary knowledge in the search process
- 1993: opponent-model search (research groups from Haifa and Maastricht)
- 1994: search technique to speculate on the fallibility of the opponent
- 2000s: probabilistic opponent models
- 2009: Opponent modelling for case-based adaptive game AI
- 2013: Generic opponent modelling approach for RTS games

Video Games

Two possible roles:

As a companion:

- Behave according to the human player's expectations
- Avoid being annoying

As an opponent:

- Adapt to the human players playing style
- Match the human players skills (play neither too weak nor too strong)

$\hookrightarrow$ Goal: raise the entertainment factor

Challenges:

1. Realistic and complex game environments
2. Little time for observation
3. Often only partial observability of the environment

Opponent modelling has to be performed in parallel to graphics rendering, rudimentary AI, ...

Explicit opponent models:

- Specification of opponents attributes separated from decision-making process

Implicit opponent models:

- Game AI is finetuned to a specific type of opponent (without actually referring the attributes)

Approaches:

- Modelling opponent's actions
- Modelling opponent's preferences

Preference-based approach:

- Opponent modelling as a classification problem
- Classification as one of multiple models based on data collected during the game
- AI behaves based on the classification

Successful implementations in RTS games (Civilization IV), shooters, ...

Adaptive game AI:

- Game AI capable of adapting to changing circumstances
- Typically implemented with machine-learning techniques
- Learning effective behaviour while the game is in progress

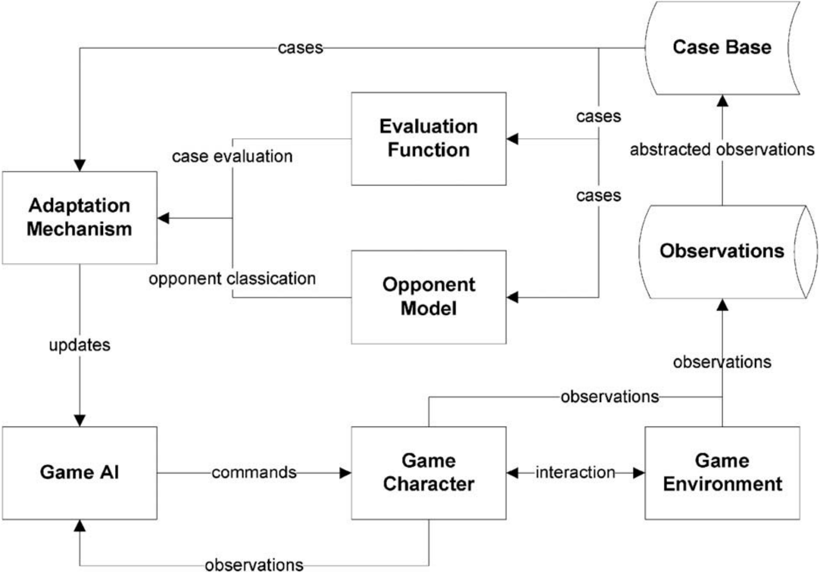
Problems with 'online learning':

- Often too many learning trials necessary for practical use
- Characters die or the game finishes before effective behaviour is learned
- Establishing effective behaviour of game AI in a stable and reliable manner is difficult

Case-based adaptive game AI:

- Game AI automatically gathers domain knowledge
- Results are immediately exploited

Particularly effective if observations from online games (MMOs) are available


cases
Case Base
Evaluation Function
cases
abstracted observations
case evaluation
cases
Adaptation Mechanism
Observations
opponent classication
Opponent Model
updates
observations
observations
Game AI
commands
Game Character
interaction
Game Environment
observations

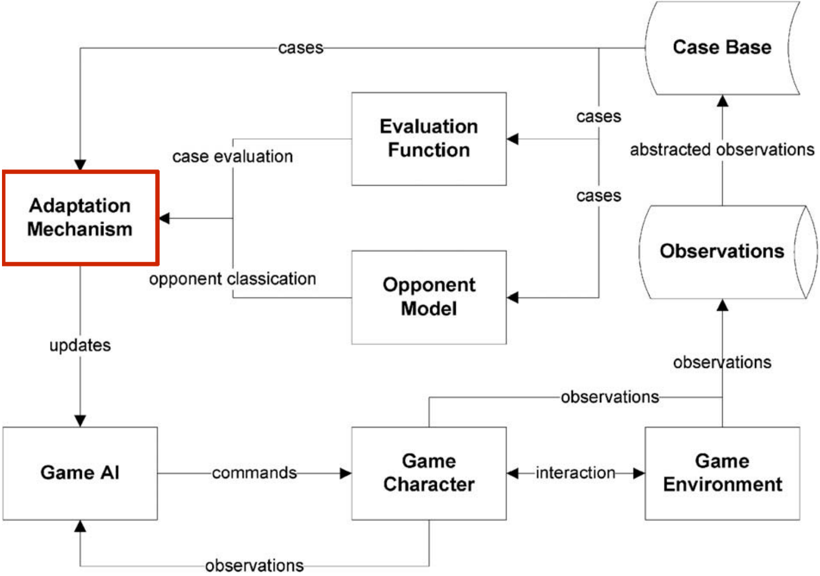
Case Base:

- Extracted from character and environment observations
- Structured in standard format with timestamp
- Taken from a multitude of games

Case Base is used to extract:

1. Opponent models
2. An evaluation function

Game environment: Spring

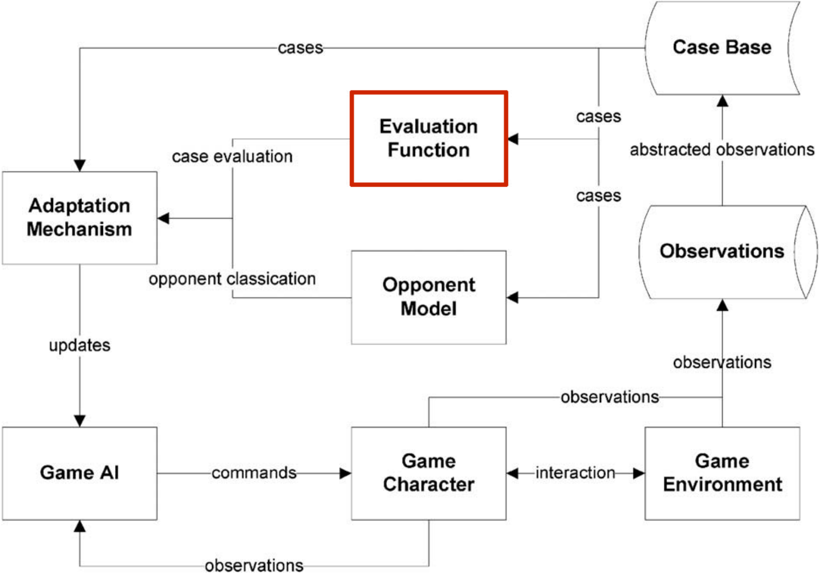

Case Base
cases
Evaluation Function
cases
abstracted observations
case evaluation
Adaptation Mechanism
cases
Observations
opponent classication
Opponent Model
updates
observations
observations
Game AI
commands
Game Character
interaction
Game Environment
observations

Adaptation mechanism:

- Indexes collected games and clusters observations (offline)
- Initialized with previously successful game strategy
- Strategy selection using similarity matching (online)

Game Strategy

”Configuration of parameters that determine strategic behaviour”
- (overall 27 parameters in the used game AI)


Case Base
cases
Evaluation Function
cases
abstracted observations
case evaluation
cases
Adaptation Mechanism
Observations
opponent classication
Opponent Model
updates
observations
observations
Game AI
commands
Game Character
interaction
Game Environment
observations

Evaluation Function

$V(p) = W_p V_1 + (1 - W_p) V_2$

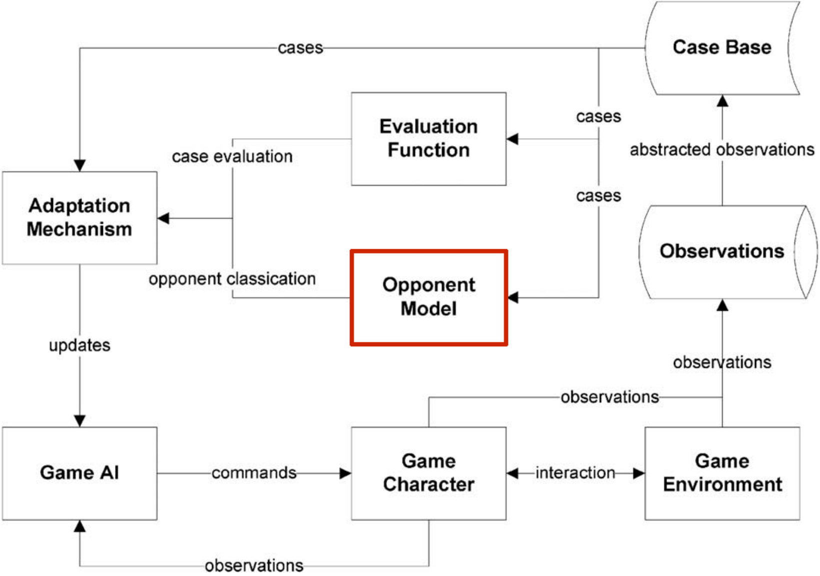
Parameter:

- p = phase of the game (opening, end game, ...)

Evaluative terms:

- V1 = material strength
- V2 = commander safety

$W \in [-1...1] = \textit{free parameter}$


Case Base
cases
Evaluation Function
cases
abstracted observations
case evaluation
Adaptation Mechanism
cases
Observations
opponent classication
Opponent Model
updates
observations
observations
Game AI
commands
Game Character
interaction
Game Environment
observations

Incorporating opponent modelling

10 features of high-level strategy:

Nr.	Feature	Meaning
1.	# observed k-bot units	Global strategic preference
2.	# observed tank units	
3.	# observed air units	
4.	# tech. adv. constructions	Technological development
5.	# metal extractors	Economy strength
6.	# solar panels	
7.	# wind turbines	
8.	first attack on metal extractors	Aggressiveness
9.	first attack on solar panels	
10.	first attack on wind turbines	

Opponent models are automatically established based on case base

Models are utilised in a game state ...

- ... early enough too have a strong impact on the outcome
- ... not too early for observing strategic choices

$\hookrightarrow$ Usually models are established after 150 game states
($\approx$ 10 *minutes of realtime play*)

Generation of opponent models:

- Cluster feature data using k-means algorithm
- Measure differences in opponent behaviour using Euclidean distance

Utilising opponent models:

- Offline processing: label each game in case base with information about the opponent
- Classify opponent based on identified clusters (nearest-neighbour)

Offline game AI initialisation:

1. Choose the most observed opponent as the most likely to be pitted against
2. Initialise game AI with the game strategy that has been observed to be the most effective against this particular opponent

Online strategy selection:

- Select strategy in phases of transition
- Choose opponent models if available

If no opponent models available:

1. Select N games from the Case Base that are similar to the current game
2. Select M games from the preselected N games that satisfy a goal criterion
3. Perform the strategy of the game most similar to the current game state

Fitness value

Metric for desired behaviour:
100 = significant victory / 0 = tied situation

Continuous validation of the chosen strategy:

- Measure difference in fitness value of current game and selected game
- Compensate by estimating ultimate fitness value with current strategy applied

If opponent models are available:

- Additional moment, when the opponent can be classified accurately
- Adapt game strategy when initially predicted opponent does not match observed opponent model

$\hookrightarrow$ Opponent classification process is incorporated in strategy selection

Experiments

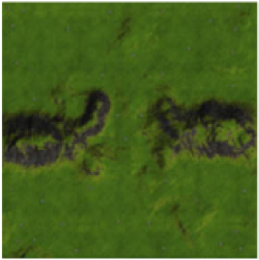
1. How well does the case based AI adapt to the original AI with medium strength?
2. How well does the case based AI adapt to a previously unobserved opponent playing a randomly generated strategy?

Setup:

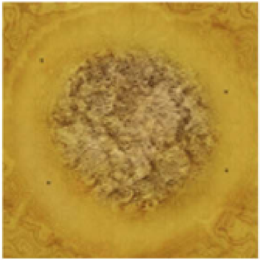
- Open source game AI: extended with case base vs. original version
- Three different maps
- All trials repeated 150 times

Baseline: All experiments performed with disabled case base and randomly selected strategy

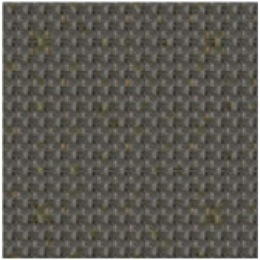
Experiments are performed in basic mode and in a mode with incorporated opponent modelling



(a) SmallDivide



(b) TheRing

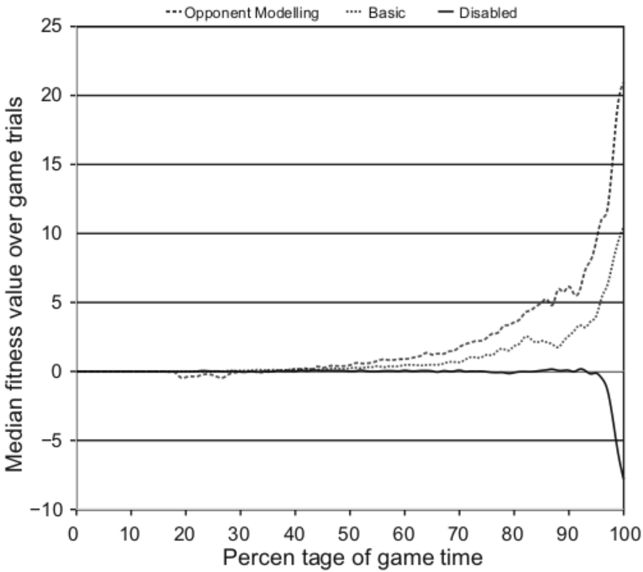


(c) MetalHeckv2

1. Experiment:

Adaptation mode	Trials	Goal achv.	Goal achv. (%)
SMALLDIVIDE			
Disabled	150	59	39
Basic	150	115	77
OM	150	135	90
THERING			
Disabled	150	90	60
Basic	150	122	81
OM	150	127	85
METALHECKV2			
Disabled	150	70	47
Basic	150	124	83
OM	150	130	87

Map: SmallDivide



2. Experiment:

Adaptation mode	Trials	Goal achv.	Goal achv. (%)
SMALLDIVIDE			
Disabled	150	71	47
Basic	150	96	64
OM	150	136	91
THERING			
Disabled	150	76	51
Basic	150	93	62
OM	150	93	62
METALHECKV2			
Disabled	150	54	36
Basic	150	60	40
OM	150	79	53

Per opponent:

SmallDivide				
Opponent	Trials	Adaptation mode		
		Disabled	Basic	OM
1	10	4	9	9
2	10	6	8	9
3	10	5	5	9
4	10	4	3	5
5	10	7	9	9
6	10	7	6	9
7	10	6	7	9
8	10	7	7	9
9	10	3	6	10
10	10	5	7	9
11	10	6	8	5
12	10	6	8	7
13	10	5	7	9
14	10	5	8	7
15	10	6	6	6
Goal achv. avg. (%)		55%	69%	81%

Findings:

- Opponent modelling techniques increase the effectiveness of case-based adaptive game AI
- Approach works best in highly strategic environments

Possible improvements:

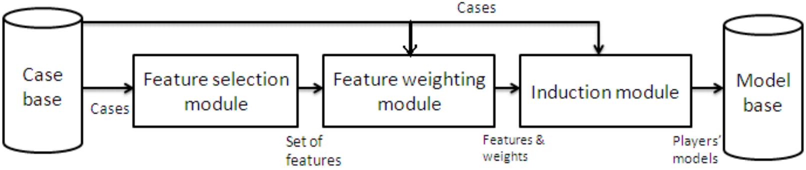
- Model opponent behaviour more detailedly with additional features
- Incorporate knowledge about feature weights

Generic approach (2013)

Main objectives:

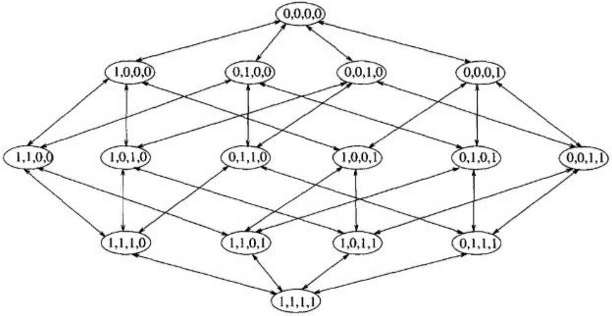
1. **Generalization:** Can be generically applied, without needing knowledge about specific game features
2. **Robust adaptability:** Cope with opponents that switch strategy by continuous tracking of classification
3. **Efficiency:** Avoid inefficient online learning

Offline phase:

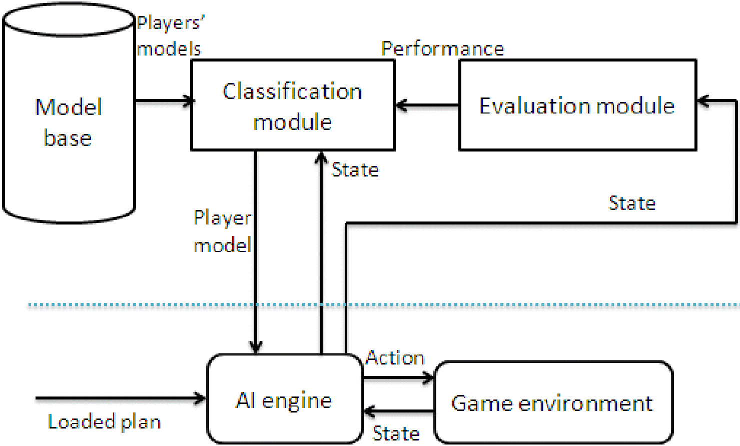


Offline feature processing:

- Feature selection: search for feature subset and corresponding state space
- Induction algorithm: Cluster observations and evaluate accuracy
- Find best fitting state using Best-first search



Online phase:



Constant adaptation:

- Game AI informs evaluation module with the current state periodically
- Evaluation module estimates the AI player's current performance against its opponent
- If performance is greater than or equal to a threshold: continue with current model
- Else: Reclassify the opponent model (k-nearest neighbour algorithm)

$\hookrightarrow$ Increased robustness against opponents changing their strategy

Representation of the model base:

Feature name	Feature index	Feature weight	Game index									
			G1	G2	G3	G4	G5	G6	G7	G8	G9	G10
AIR_DEFENCE	F2	0.4	2	8	10	3	1	23	17	5	0	8
FAST_UNITS_RATE	F3	0.2	4	22	16	18	10	22	9	20	21	15
MAX_DEFENCES	F6	0.6	22	88	31	44	19	3	8	9	15	23
AIR_DEFENCE	F7	0.1	11	9	15	9	4	7	12	11	17	7
UNIT_SPEED_SUBGROUPS	F8	0.7	12	3	11	5	2	6	19	7	2	5
Opponent Model			Type 1	Type 2	Type 1	Type3	Type 2	Type2	Type 1	Type3	Type 1	Type 1

Conclusion

- The case-based approach avoids the shortcomings of resource-intensive online learning approaches
- Opponent modelling increases effectiveness of game AI significantly
- Further improvements have been made applying generic feature selection and learning feature weights

Bibliography I

Bakkes, Sander C.J., Pieter H.M. Spronck, and H. Jaap van den Herik (2009). "Opponent modelling for case-based adaptive game AI". en. In: *Entertainment Computing* 1.1, pp. 27–37. ISSN: 18759521. DOI: 10.1016/j.entcom.2009.09.001. URL: https://linkinghub.elsevier.com/retrieve/pii/S1875952109000044 (visited on 03/15/2019).

Farouk, G. M., I. F. Moawad, and M. Aref (2013). "Generic opponent modelling approach for real time strategy games". In: *2013 8th International Conference on Computer Engineering Systems (ICCES)*, pp. 21–27. DOI: 10.1109/ICCES.2013.6707164.

Bibliography II

Herik, H. J. Van Den, H. H. L. M. Donkers, and P. H. M. Spronck. "Opponent modelling and commercial games". In: *Lucas (Eds.), Proceedings of the IEEE 2005 Symposium on Computational Intelligence and Games (CIGfffdfffdfffd05), 2005*, pp. 15–25.

Schadd, Frederik, Sander Bakkes, and Pieter Spronck (2007). "Opponent Modeling in Real-Time Strategy Games." In: pp. 61–70.