# Towards a game-independent model and data-structures in digital board games: an overview of the state-of-the-art

Luiz Jonata Pires de Araujo
l.araujo@innopolis.university
Innopolis University
Innopolis, Republic of Tatarstan
Russia

Mariia Charikova
m.charikova@innopolis.ru
Innopolis University
Innopolis, Republic of Tatarstan
Russia

Juliano Efson Sales
juliano-sales@uni-passau.de
Department of Computer Science and Mathematics - University of Passau
Germany

Vladislav Smirnov
v.smirnov@innopolis.ru
Innopolis University
Innopolis, Republic of Tatarstan
Russia

Ananga Thapaliya
a.thapaliya@innopolis.ru
Innopolis University
Innopolis, Republic of Tatarstan
Russia

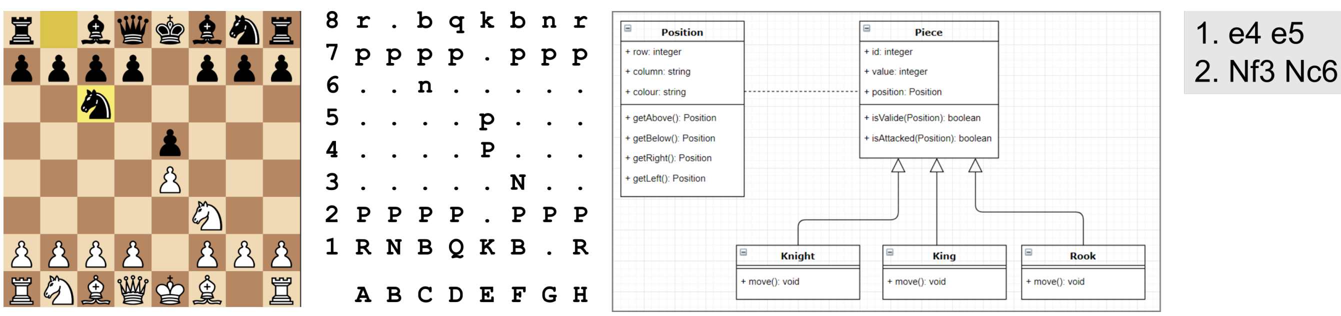


**Figure 1: Different representations for a particular game state in Chess.**

## ABSTRACT

The increasing number of options of digital board games is exciting not only from an entertainment perspective but also from an academic prospect. It enables, for example, the application of modern computational techniques and algorithms for extracting and analyzing data from a variety of games – which range from classics like Chess and Go to modern board games with complex rules like Settlers of Catan and Terra Mystica. It is intuitive that different digital board games require distinct representation schemes and data structures to save, for example, the status or snapshot of a particular game at a specific moment. The choice for a representation model and data-structures is a crucial design decision that affects the selection of an algorithmic solution as well as the suitableness for artificial intelligence agents. This survey focuses on the different schemes and data structures used to represent game states, physical components, players, actions and rules for digital board games that have been reported in the academic literature. This study aims to lay the groundwork for the development of a game-independent computational framework which includes a generic game representation to facilitate and promote the application of computational techniques such as, for example, artificial intelligence and machine learning to this domain.


*FDG 2019, August 26-30, San Luis Obispo, CA*

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/1122445.1122456

## CCS CONCEPTS

• **Computer Simulation of Board Games** → **Knowledge Representation**; • **Game Theory** → *Representation scheme*; Data Structure; • **Artificial Intelligence** → Agents and Environments.

## KEYWORDS

board games, game states, knowledge representation, data structures

**ACM Reference Format:**
Luiz Jonata Pires de Araujo, Mariia Charikova, Juliano Efson Sales, Vladislav Smirnov, and Ananga Thapaliya. 2018. Towards a game-independent model and data-structures in digital board games: an overview of the state-of-the-art. In *FDG2019: 2nd Workshop on Tabletop Games, August 26–30, 2018, San Luis Obispo, CA.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/1122445.1122456

## 1 INTRODUCTION

Digital board games have gained increasing popularity in the last decades and attracted the attention of practitioners of old-fashioned board games, the entertainment industry and the academic community [32]. This affirmation can be confirmed by observing, for example, the growing number of users of online platforms like *Tabletop Simulator* on *Steam*[1], *Board Game Arena*[2] and *Online Terra Mystica*[3], to mention some.

Despite some criticism when comparing the digital platform to the traditional media, digital board games have created opportunities for applying computational techniques to enhance game experience and gain new strategy insights [29]. For example, an algorithm can be used to automate the game progression by managing resources, counting points and checking winning conditions. Another possibility is the implementation of competitive artificial intelligence players that leverage from computational resources and stored data from previous games [32].

In addition to its appeal to the market and entertainment industries, digital board games are also attracting from an academic perspective as they offer appealing study cases to university level courses [21]. Teaching and research of topics like algorithms, artificial intelligence (AI), software engineering and machine learning (ML) have found fertile soil for application into this domain [9].

An important design decision when developing digital board games – and games in general – is the representation scheme employed to describe a discrete game state, which is a detailed snapshot of a particular game at a certain moment. Representation of game states and the knowledge held by each player are non-trivial topics that touch diverse research fields including game development and game theory [23]. More complex issues arise when considering abstract concepts such as player's biases, satisfaction and goals [4].

From a computer science perspective, representation schemes and data structures for game states, players, actions and rules are relevant as they can enable the collection of data that would not be available in physical board games and the use of algorithmic approaches [27]. For example, Samuel [39] combined a tree-search algorithm with seminal ideas in self-learning to leverage from a library of master play and identify the promising sequence of moves in a game of checkers. Moves were represented as edges in a tree data structure, while game states were nodes that contained a table with parameters and values to describe the 'board situation'. The employed data structures were, therefore, convenient for the used algorithmic approach and contributed to the success of the experiment in which Samuel's program defeated a top-ranked checkers player [11].

It seems reasonable to affirm that certain representation schemes are more suitable than others depending on the characteristics of a particular digital board game. For example, the information available to each player regarding the current game state can differ from perfect (*e.g.* as in Chess) to imperfect (*e.g.* as in Poker) [23]. Moreover, the representation model and its containing data affect the appropriateness of an algorithm (e.g. an AI agent) to process the information and also take decisions in the context of games.

This paper reviews some of the most commonly used representations for game states, agents or players, rules and actions reported in the literature. By introducing a classification for such schemes, we expect to lay the groundwork for a unified and game-independent reference representation to model tabletop games and enable the effortless application of computational techniques (e.g. AI and ML) to digital board games. This study also draws attention to the potential use of an agnostic-game representation as part of a framework that can be used in an academic environment to promote ludic applications of algorithms in university courses.

The rest of this paper is organized as follows. Section 2 presents some concepts in the realm of games, agents, environment and main components. Section 3 describes the related work of representation schemes and the employed data structures. Section 4 discusses how these approaches are connected to different types of digital board games as well as the perspective of implementing a game-independent framework supporting education projects. Finally, Section 5 summarizes this paper and shows research directions.

## 2 GAME STATES, PLAYERS, ACTIONS AND RULES

There is a wide taxonomy of what can be considered to fall under the domain of tabletop games [6, 7]. For this paper, "*games*" must present the following distinctive features outlined by [19]: predefined rules and regulations, different variables, results that can be measured with different qualities, players contribution of their time and effort are connected to the results, outcomes are debatable and at least one characterizing objective. Concerning regulations, for example, they model how players are allowed to proceed onward, describe abstractions for expressing game coordination, board setup, players' interactions, end game conditions and possible outcomes.

We have followed the strict definition and representation for agents and environment to make the survey specific and not vague. An agent is whatever can be seen as recognizing its environment through sensors and following up on that condition through actuators [37]. Before the simulation process, it is necessary to understand certain agents criteria and how they behave in a certain type of environments. These criteria and environment can affect the outcome and further optimization, thus making it necessary to know and understand the agent and the environment [12, 17, 30]. They include deterministicness (deterministic or stochastic), staticness (static or dynamic), observability (fully or partially), knowledge (known or unknown), episodicness (episodic or sequential) and discreteness (discrete or continuous) and the agent types (single or multi) [3, 33].

While the previous elements find correspondence onto physical objects, other elements require some level of abstraction to be understood. Any game consists of a sequence of states, where each state is characterized by a combination of visual, audio and/or animation cues [14]. Figure 2 presents the intuitive idea of a simplified game state process. For example, which characteristics in a game state point that the game reached the end? As an example, a game state for Chess where the King attacked in an indefensible position characterizes the end game with loss as a result. Such elements are recurrent in the field of knowledge representation and appear in the early stages of the development of such simulation software. Figure 3 gives us the idea of different states in a board game and how it changes after some specific actions by the player or computer itself.

[1]https://steampowered.com
[2]https://boardgamearena.com/
[3]https://terra.snellman.net/

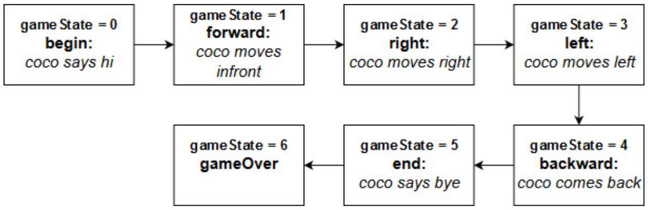


**Figure 2: Example of a simplified game state process**

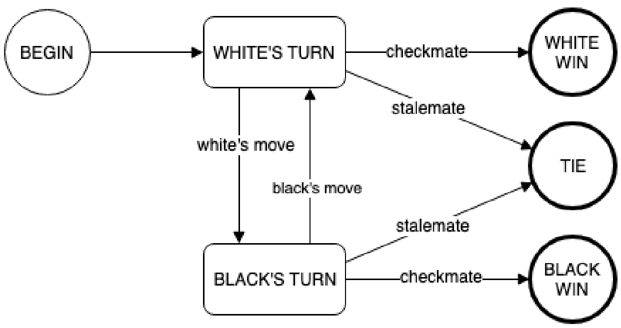


**Figure 3: Example of game states of chess abstracting the board configuration.**

The change of state of the game depends upon actions. For example in Chess, moving the queen to three spaces forward. This is called an action. Thus, an action is any kind of simple change that a player can make to the condition of the game [8]. It is a solitary choice. Furthermore, it quite often implies that at least one game segments are moved around. As a player, you do something which changes the state of the game. This action has its consequences; for instance, your move on Chess will either take you one step toward a win, tie or lose the game. As mentioned before, in this paper "game" must have six distinctive features and have its actions based on a ruleset [38]. Every game has its own rules and regulations. For example, you can only move one step forward with a pawn in a game of Chess.

## 3 REPRESENTATION SCHEMES

There are many representation schemes and related data structures that are used for computer simulation of board games. Here, we discuss some of the available schemes based on the previous works within the last few years. It includes representation via primitive data types and dictionaries, logical expressions, graphs and neural networks. Game theory is included in this overview as it presents several of the concepts connected to representations and actions which are common to tabletop games [2].

### 3.1 Representation in Game Theory

*Game theory* aims at studying the interaction of self-interested agents from the perspective of a mathematical model. This model, whose representation is usually based on the *utility theory*, describes the agents' interests as their space of actions associated with both

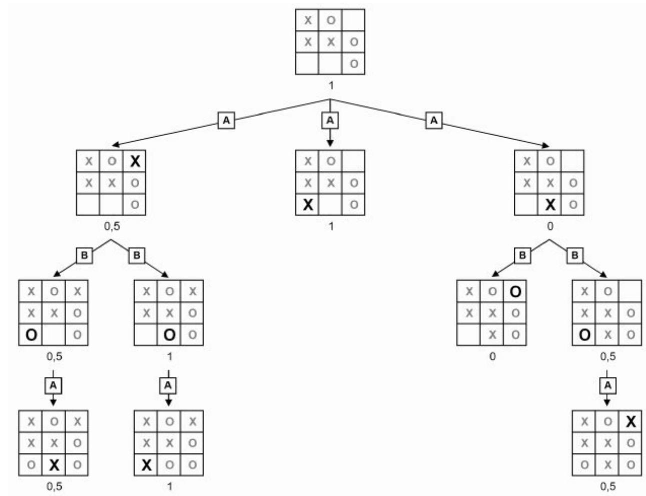

**Figure 4: Game tree of tic-tac-toe from a random state.**

the respective rewards and the probability of the other agents' actions [23].

Game Theory uses typically three forms of representation: *normal*, *extensive* and *characteristic function*. The *normal form* represents the model in tensors, each dimension representing a player [23]. In a two-player game, represented in a matrix, the lines account for the actions of the first player; the columns account for the second's actions while the cells store the rewards as depicted in Figure 5.

But in general, any function that relates to result for every player with a conceivable mix of activities can be used to represent this form. At the point when a game is displayed in normal form, it is assumed that every player has synchronous action if only the other player's action is unknown. Again, extensive form will be used if the players have some data about the decision of other players.

Also, each normal form game is within the equivalent extensive form game. Nevertheless, the change to normal form may result in an exponential blowup in the measure of the portrayal, making it computationally unfeasible [23].

In the *extensive form*, games are represented by tree structures. Figure 6 shows an example which displays the representation scheme using the tree data structure [23]. This form is utilized to normalize games with the period ordering of moves. For a player, point of decision is represented using every vertex. These players are indicated using the vertex number. Any conceivable activity of players is entitled using vertex lines. The base of the tree determines the adjustments (playoffs). The extensive form is the multi-player speculation of a decision tree [13]. This form can likewise catch synchronous game moves with defective data. This is displayed either using the spotted line for the connection of various vertices of similar data set or using a closed line.

In certain cooperative games, players are allowed to form groups to cooperate (to an extent) and negotiate how to share a given commodity. Such games are sometimes described as "cooperative games with transferable utilities" [22]. The *characteristic function form* is a mathematical formulation to describe the profits from

| | | SUBJECT 2 | |
|---|---|---|---|
| | | Method P | Method Q |
| SUBJECT 1 | Method P | S(1P), S(2P) | S(1P), S(2Q) |
| | Method Q | S(1Q), S(2P) | S(1Q), S(2Q) |

**Figure 5: Normal form representation with expected payoff matrix for two subjects.**

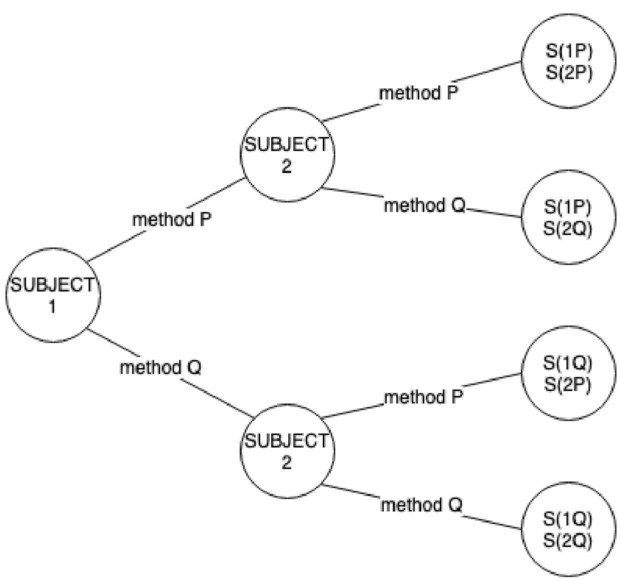


**Figure 6: An extensive form representation equivalent to Figure 5.**

every possible coalition of players. It accepts that all settlements are estimated in similar units and that there is a transferable utility which enables side instalments to be made among the players [23]. Individual players use side instalments as actuation to utilize certain commonly advantageous systems. For framing coalitions or alliances there is an inclination of players who have close targets in the game [23]. Coalition arrangements provide an outline which is advantageously considered by lessening the game to a structure in which it assumes a focal job.

The *normal form* is the fundamental representation method for finite games. It assumes, however, that players act simultaneously. When there is a temporal aspect in the game, in which the actions occur sequentially, the extensive form offers a better representation. These first two forms model the *non-cooperative games*, which describes single agents acting as players. *Cooperative games*, however, deals with groups of agents interacting in coalition with other groups, to which the *characteristic function* variant is designed [41].

## 3.2 Primitive data types and dictionaries

Primitive data types can be utilized to describe game states of digital versions of simple (in terms of rules and components) board and card games. For example, a portable game notable (PGN) is a standard that enables the interpretation of strings into a valid state of a game of Chess, as illustrated in Figure 1. It also permits one to know the sequence of moves that led to that particular game state. In the context of digital Chess, this representation enables the exploitation by learning algorithms of massive amounts of data from highly qualified players [40].

Another example of a digital game to which an artificial intelligence player has been implemented is Poker. The representation of a hand of Texas Hold'em, for instance, includes the following properties: the number and identification of the flopped cards, number of cards that the player owns, values of betting, values for the small blind and the big blind and the identification of the community cards (flop) [5]. This set of attributes can fit within a dictionary data structure and, because Poker is an imperfect information game (when the player has limited or incomplete knowledge about the game state), one dictionary is maintained per every player to express what it knows about the environment [10]. Game states encompassing not only the position of pieces on the virtual board but also statistics have been used in simulations of the game Risk [31].

As evident from the previous examples, strings are convenient representations for pieces on a board of a perfect information game. Another example is observed in [15] to represent for example Go, Chess and Backgammon. In that particular implementation, fixed sized strings can be used as input for learning algorithms without requiring preprocessing to extract relevant features or predictors [42].

It can be observed that representations using primitive data types and dictionaries has been restricted to describing game states and sequence of actions. The following sections present more suitable data structures employed to persist in statistical models used by AI players.

## 3.3 Logical expression representation

The symbolic representation can be used to describe finite, discrete and deterministic games. Game Description Language (GDL), for example, is based on concepts from the field of knowledge representation [20]. It describes game states, actions, roles and goals in a general way using predicates resembling first-order logic.

Logic predicates have also been used to describe or generate rules and validate actions for the digital versions of Diplomacy, Checkers and Tic-tac-toe [34].

One of the advantages of logic-based representation for game states, actions and rules is the possibility of validation prior to their execution. This method is observed, for example, in implementation for Chinese Checkers and games with similar structure tree in which representation of games and players' actions are represented by logical expressions on the edges of a game tree [35].

## 3.4 Graphs

As mentioned earlier, trees are convenient data structures for representing sequences of actions (edges) that lead to reachable game states (nodes) from an initial configuration.

Some algorithmic approaches are recurrently applied to digital board games that employ the tree-based representation scheme. For example, Monte-Carlo tree search has been successfully employed in simulations of games like Chess and checkers as well as modern eurogames like Settlers of Catan [8]. This particular approach evaluates game states by repeatedly assessing random moves and evaluating the probability that a certain state lead to a winning condition [24].

Graphs are recurrently used for representing the relationship between game states and actions. Keller and Schiffel [20] mention two special types of game states, initial and terminal states, and defines gameplay in terms of the interactions through the game state tree. The authors argue, for example, that simple uninformed search algorithms like depth-first search and breadth-first search are good-enough solutions simple games. On the other hand, the information in the nodes, in this case, described in terms of logic predicates, can be used to develop heuristics that enable pruning the search space of promising actions.

The mentioned example using graph representations resemble game trees as examined in the field of game theory. They are useful as they provide a temporal representation of a sequence of actions and their expected trade-offs. While states are always represented as nodes, actions are edges in such data structure. It is, therefore, suitable for representing turn-based games; on the other hand, it may give rise to issues when modelling simultaneous players' actions.

Finally, trees (i.e. acyclic connected graphs) are a suitable data structure to represent adaptable evaluation functions used to identify promising actions in games with large search space like Checkers and Chess [39, 40].

## 3.5 Artificial neural networks

Artificial neural networks (ANN) are computational models inspired by the intercommunication between numerous neuron within the nervous system [28]. Intelligence in such a network is collective or distributed and emerges from the interactions between its simpler processing units. This learning model has demonstrated to be particularly useful for pattern classification, clustering and forecasting to mention a few of its applications [16].

An advantage of ANN is its resilience for noisy input data, which is convenient for incomplete information games such as Poker [42]. ANN has been increasingly applied to predict opponents' actions and persist strategies in a variety of games (e.g. Chess and Backgammon) [5, 40]. Such an approach requires the collection of a large number of predictors, or data features to appropriately describe the game state and then predict one of the possible discrete outcomes. However, one disadvantage of such learning method is the resulting non-human-readable model.

# 4 DISCUSSION

## 4.1 General Representation Discussion

Among the properties of the *representation schemes* aforementioned, it's important to shed some light to three interrelated aspects: *readability*, *effectiveness*, and *redundancy*. *Readability* measures to what extent a human reader can easily understand the data representation; and *effectiveness* concerns the appropriateness of such a structure to be applied to a certain algorithm. Although simple games, as some instances of *game theory*, can conciliate readability and *effectiveness*, in many cases they conflict. *Redundancy* comes as a side effect, mediating the other two aspects.

In a context focused on education, both readability and effectiveness are equally important since students need to understand the data content and learn how to develop high-performance algorithms. However, the use of human-readable data resources often comes at the cost of algorithm efficiency. While representations of decisions and outcomes in game theory, primitive data types, dictionaries, logical expressions enable straightforward interpretation of generated rules, such models prevent quickly processing and high accuracy by machine learning and predictive models which also occurs in other domains like machine translation [18]. Although allow graph data-structures are comprehensible to humans to an extent, they favour especially graph algorithms capable of analyzing the relationships (edges) between entities (nodes). Finally, progressing towards more complex data representations, ANNs have proved to result in high performance regarding predictive models and have been one of the dominating techniques on ML approaches in the recent years [1].

## 4.2 Analyses of the works

It can be observed that most of the studies have an algorithm-oriented design for representing a particular game and its elements. In other words, the format of the dataset is tailored according to the targeted processing strategy. Although such an approach has been useful to the moment, it prevents utilization and, as a result, every new digital board game requires an entire development cycle.

The investigated literature surveyed in this paper has different focal points. Representations using game trees, payoff matrices, primitive data types and dictionaries, logical expressions and graphs prioritize readability and are accessible to both humans and algorithms. Neural networks, although not accessible, can be used to predict an outcome given imprecise data, that is, when some of the input features are either unknown or might be wrong. The authors argue that such a data structure appears to be appropriate to model a player's guess based on incomplete information as in a game of poker, for example, [42].

It can also be observed that some game elements are typically represented in a similar manner. For example, both players' actions are rules regarding legal game states reachable from the current one are represented as a graph, with nodes containing dictionary-like structures. Logic symbols, on the other hand, have been used to denote rules and enable some reasoning by an automatized agent. Finally, neural networks, typically models or simulates how an agent processes information extracted from game states and acts accordingly.

**Table 1: Examples from the literature on representations of game states, actions and auxiliary data structures**

| Reference | Board game | Game state representation | Actions | Auxiliary data structures |
|---|---|---|---|---|
| Billings et al. 2002 | Poker | Primitive data type | Primitive data type | |
| Sigan and Malinowski 2004 | Chess | Primitive data type | Primitive data type | |
| Ghory 2004 | Go, Chess, Backgammon | Primitive data type | Primitive data type | ANN* |
| Magerkurth et al. 2004 | Monopoly | Logical expressions | Logical expressions | |
| Keller and Schiffel 2009 | Chess, Checkers, Go | Trees | Graphs | |
| Osborne 2010 | Risk | Primitive data type | Primitive data type | |
| Polberg et al. 2011 | Diplomacy, checkers, Tic-tac-toe | Logical expressions | Logical expressions | |
| Mahlmann et al. 2012 | Quoridor | Primitive data type | Graphs | ANN* |
| Polk and Oommen 2015 | Chinese Checkers | Logical expressions | Trees | |
| Respall et al. 2018 | Quoridor | Primitive data type | Trees | |

(*) ANN: Artificial neural network

### 4.3 Game-specific issues

**Perfect and imperfect information:** Games are sometimes classified as perfect and imperfect information. Such a taxonomy has been studied in the field of game theory but also occur in board games. Perfect information games (e.g. Checkers, Chess and Terra Mystica) are undoubtedly easier to represent, while imperfect information games (e.g. Tigris and Euphrates, Settlers of Catan and Fief France 1429) require additional data structures to represent what is known by each player and how it takes decisions given partial information.

**Time:** In certain board games, some players' actions are resolved simultaneously. In Diplomacy, for example, players orders are written within a fixed time window but resolved simultaneously. Current representation schemes seem to require customization to handle possible outcomes and payoffs, which resemble the payoff matrix in game theory.

**Abstract concepts:** Although not in the scope of this study, some fascinating concepts arise from the interactions between players. For example, how to models a player's biases, animosities and beliefs? Backstabbing games offer exciting research directions in this context as players face situations in which subjective elements play an essential factor.

Several other aspects involving the player's consciousness or motivations can come into play. The objective of this paper is not to provide an exhaustive list but to draw attention that simulating or persisting data from a human player is a daunting task.

### 4.4 Towards a game-independent framework

As mentioned previously, the choice for a particular representation is intrinsically connected to the targeted processing approach. While this specialization favours the performance, it prevents the utilization of source code and data. For example, heuristics and evaluation functions that work for Diplomacy might not be feasible for a newly released strategic and negotiation board game.

A specific representation scheme designed to a particular game, or games which share similar characteristics, can be addressed by different algorithms. As a result, a more flexible game representation or model could accommodate a more extensive range of algorithms applicable to a game. On the other hand, machine-readable disallow such an extension, tying the problem to its solution.

The solution for this conflict points to the construction of a flexible framework which stores game-related datasets in a human-readable format according to a meta-description language from which machine-readable formats can be generated using algorithm-specific transforms.

Figure 7 depicts a reference framework to store and transform data sets support the teaching of computer science courses in the context of game-related problems.

Such an alternative model could enable the observation of complex relationships between algorithmic approaches and types of games of the kind: "Algorithm X is more successful than algorithm Y for complete-information games which involve data with features A, B and C".

## 5 CONCLUSION AND FUTURE RESEARCH

Digital version of board games have gained increasing popularity among the community and caught the attention of the entertainment industry and academic researchers. This study aims to support more efficient implementation of digital board games by facilitating the development of simulated games and a better understanding of the alternatives for representing game states, players, actions and rules.

This study provided a brief description of contextual studies on different representation scheme and data structures for digital

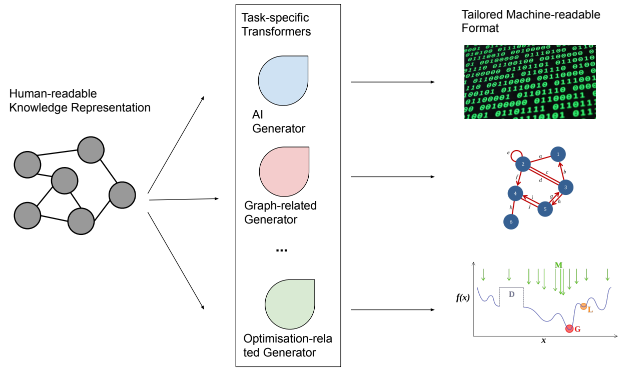


**Figure 7: The reference framework comprehending a repository in a human-readable format and a set of transformers able to convert data of a particular game to an algorithm-specific format targeting a given problem in a computer science field, such as optimization, graph theory and artificial intelligence.**

board games to pave the way towards a game-independent representation. By providing such a categorization, it was possible to observe some patterns arising regarding representation schemes for particular elements. For example, graphs (including trees) are popular data structures in digital board games as they easily allow the representation of actions as iterators between game states, represented as edges and nodes respectively. It was also possible to observe that different representations are often combined into the same game due to the complexity of factors connected to game features and players knowledge. The taxonomy here provided is, therefore, a useful resource that aims to help practitioners in this field.

Future research aims to combine some of the representation schemes and data structures into an extensive data set and provide a learning platform for students in a university. An expected outcome from such an experiment is an improved concentration and productivity of students for learning advanced algorithms and their application. This will help future researchers to overcome some of the challenges and difficulties listed in this work. Finally, the authors expect to conduct research assessing the relationship between game features, solving algorithms and representation models to gain valuable insights into the design on new board game and software development.

## Acknowledgment

The authors are grateful to Doctor Joseph Brown from Innopolis University for his valuable opinions and essential advice during the writing of this paper.

## REFERENCES

[1] Itamar Arel, Derek C Rose, Thomas P Karnowski, et al. 2010. Deep machine learning-a new frontier in artificial intelligence research. *IEEE computational intelligence magazine* 5, 4 (2010), 13–18.

[2] D. Ashlock, Eun-Youn Kim, and N. Leahy. 2006. Understanding representational sensitivity in the iterated prisoner's dilemma with fingerprints. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)* 36, 4 (July 2006), 464–475. https://doi.org/10.1109/TSMCC.2006.875423

[3] Randall D Beer. 1995. A dynamical systems perspective on agent-environment interaction. *Artificial intelligence* 72, 1-2 (1995), 173–215.

[4] Salem Benferhat, Didier Dubois, Souhila Kaci, and Henri Prade. 2002. Possibilistic Logic Representation of Preferences: Relating Prioritized Goals and Satisfaction Levels Expressions. In *Proceedings of the 15th European Conference on Artificial Intelligence (ECAI'02)*. IOS Press, Amsterdam, The Netherlands, The Netherlands, 685–689. http://dl.acm.org/citation.cfm?id=3000905.3001051

[5] Darse Billings, Aaron Davidson, Jonathan Schaeffer, and Duane Szafron. 2002. The challenge of poker. *Artificial Intelligence* 134, 1-2 (2002), 201–240.

[6] Yngvi Björnsson. 2012. Learning rules of simplified boardgames by observing. In *Frontiers in Artificial Intelligence and Applications*, Vol. 242. IOS Press, 175–180. https://doi.org/10.3233/978-1-61499-098-7-175

[7] Joseph Alexander Brown and Marco Scirea. 2020. Procedural Generation for Tabletop Games: User Driven Approaches with Restrictions on Computational Resources. In *Proceedings of 6th International Conference in Software Engineering for Defence Applications*, Paolo Ciancarini, Manuel Mazzara, Angelo Messina, Alberto Sillitti, and Giancarlo Succi (Eds.). Springer International Publishing, Cham, 44–54.

[8] Guillaume Chaslot, Sander Bakkes, Istvan Szita, and Pieter Spronck. 2008. Monte-Carlo Tree Search: A New Framework for Game AI.. In *AIIDE*.

[9] Nian-Shing Chen and Gwo-Jen Hwang. 2014. Transforming the classrooms: innovative digital game-based learning designs and applications. *Educational Technology Research and Development* 62, 2 (apr 2014), 125–128. https://doi.org/10.1007/s11423-014-9332-y

[10] Thomas H Cormen, Charles E Leiserson, Ronald L Rivest, and Clifford Stein. 2009. *Introduction to algorithms*. MIT press.

[11] Edward A Feigenbaum and Julian Feldman. 1963. *Computers and Thought*. McGraw-Hill, Inc., New York, NY, USA.

[12] Jacques Ferber and Gerhard Weiss. 1999. *Multi-agent systems: an introduction to distributed artificial intelligence*. Vol. 1. Addison-Wesley Reading.

[13] Drew Fudenberg and Jean Tirole. 1991. Perfect Bayesian equilibrium and sequential equilibrium. *Journal of Economic Theory* 53, 2 (1991), 236–260. https://econpapers.repec.org/RePEc:eee:jetheo:v:53:y:1991:i:2:p:236-260

[14] Dan R Ghica. 2009. Applications of game semantics: From program analysis to hardware synthesis. In *2009 24th Annual IEEE Symposium on Logic In Computer Science*. IEEE, 17–26.

[15] Imran Ghory. 2004. Reinforcement learning in board games. *Department of Computer Science, University of Bristol, Tech. Rep* (2004), 105.

[16] John A Hertz. 2018. *Introduction to the theory of neural computation*. CRC Press.

[17] Nicholas R Jennings and Michael Wooldridge. 1998. Applications of intelligent agents. In *Agent technology*. Springer, 3–28.

[18] Douglas Jones, Edward Gibson, Wade Shen, Neil Granoien, Martha Herzog, Douglas Reynolds, and Clifford Weinstein. 2005. Measuring human readability of machine generated text: three case studies in speech recognition and machine translation. In *Proceedings.(ICASSP'05). IEEE International Conference on Acoustics, Speech, and Signal Processing, 2005.*, Vol. 5. IEEE, v–1009.

[19] Jesper Juul. 2018. The game, the player, the world: Looking for a heart of gameness. *PLURAIS-Revista Multidisciplinar* 1, 2 (2018).

[20] Ingo Keller and Dipl-Inf Stephan Schiffel. 2009. A General Game Playing IDE. (2009).

[21] Jonathan Lean, Jonathan Moizer, Michael Towler, and Caroline Abbey. 2006. Simulations and games: Use and barriers in higher education. *Active Learning in Higher Education* 7, 3 (nov 2006), 227–242. https://doi.org/10.1177/1469787406069056

[22] Jean Lemaire. 1991. Cooperative game theory and its insurance applications. *ASTIN Bulletin: The Journal of the IAA* 21, 1 (1991), 17–40.

[23] Kevin Leyton-Brown and Yoav Shoham. 2008. Essentials of game theory: A concise multidisciplinary introduction. *Synthesis lectures on artificial intelligence and machine learning* 2, 1 (2008), 1–88.

[24] SM Lucas and G Kendall. 2006. Evolutionary computation and games. *IEEE Computational* (2006). https://ieeexplore.ieee.org/abstract/document/1597057/

[25] Carsten Magerkurth, Maral Memisoglu, Timo Engelke, and Norbert Streitz. 2004. Towards the next generation of tabletop gaming experiences. In *Proceedings of Graphics interface 2004*. Canadian Human-Computer Communications Society, 73–80.

[26] Tobias Mahlmann, Julian Togelius, and Georgios N Yannakakis. 2012. Evolving card sets towards balancing dominion. In *2012 IEEE Congress on Evolutionary Computation*. IEEE, 1–8.

[27] Javier Marco, Eva Cerezo, and Sandra Baldassarri. 2012. ToyVision: A Toolkit for Prototyping Tabletop Tangible Games. In *Proceedings of the 4th ACM SIGCHI Symposium on Engineering Interactive Computing Systems (EICS '12)*. ACM, New York, NY, USA, 71–80. https://doi.org/10.1145/2305484.2305498

[28] Warren S McCulloch and Walter Pitts. 1943. A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics* 5, 4 (1943), 115–133.

[29] Miguel A Nacenta, David Pinelle, Dane Stuckel, and Carl Gutwin. 2007. The effects of interaction technique on coordination in tabletop groupware. In *Proceedings of Graphics interface 2007*. ACM, 191–198.

[30] Andrea Omicini, Alessandro Ricci, Mirko Viroli, Cristiano Castelfranchi, and Luca Tummolini. 2004. Coordination artifacts: Environment-based coordination for intelligent agents. In *Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems-Volume 1*. IEEE Computer Society, 286–293.

[31] Jason A. Osborne. 2010. Markov Chains for the RISK Board Game Revisited. *Mathematics Magazine* 76, 2 (apr 2010), 129. https://doi.org/10.2307/3219306

[32] Joseph A Pape. 2012. *The Effects of Digitization and Automation on Board Games for Digital Tabletops*. Ph.D. Dissertation.

[33] S Peng, S Mukhopadhyay, R Raje, M Palakal, and J Mostafa. 2001. A comparison between single-agent and multi-agent classification of documents. In *Proceedings 15th International Parallel and Distributed Processing Symposium. IPDPS 2001*. 935–944. https://doi.org/10.1109/IPDPS.2001.925048

[34] Sylwia Polberg, Marcin Paprzycki, and Maria Ganzha. 2011. Developing intelligent bots for the Diplomacy game. *2011 Federated Conference on Computer Science and Information Systems (FedCSIS)* (2011), 589–596. https://ieeexplore.ieee.org/abstract/document/6078245/

[35] Spencer Polk and B. John Oommen. 2015. Novel AI Strategies for Multi-Player Games at Intermediate Board States. 33–42. https://doi.org/10.1007/978-3-319-19066-2_4

[36] V.M. Respall, J.A. Brown, and H. Aslam. 2018. Monte carlo tree search for quoridor. In *19th International Conference on Intelligent Games and Simulation, GAME-ON 2018*. EUROSIS, 5–9.

[37] Stuart J Russell and Peter Norvig. 2016. *Artificial intelligence: a modern approach*. Malaysia; Pearson Education Limited,.

[38] Katie Salen, Katie Salen Tekinba\cs, and Eric Zimmerman. 2004. *Rules of play: Game design fundamentals*. MIT press.

[39] A L Samuel. 1959. Some Studies in Machine Learning Using the Game of Checkers. *IBM Journal of Research and Development* 3, 3 (jul 1959), 210–229. https://doi.org/10.1147/rd.33.0210

[40] Mr Jack Sigan and Aleksander Malinowski. 2004. Complex Problem Solving With Neural Networks: Learning Chess. (2004).

[41] J Szép and F Forgó. 1985. *Cooperative games in characteristic function form*. Springer Netherlands, Dordrecht, 237–248. https://doi.org/10.1007/978-94-009-5193-8_22

[42] G. Tesauro. 2018. Temporal Difference Learning and TD-Gammon. *ICGA Journal* 18, 2 (2018), 88–88.