# A game-based model for human-robots interaction*

**Aniello Murano** and **Loredana Sorrentino**
Dipartimento di Ingegneria Elettrica e Tecnologie dell'Informazione
Università degli Studi di Napoli Federico II, Italy
{aniello.murano,loredana.sorrentino}@unina.it

## I. Abstract

Game theory has exhibited to be a fruitful metaphor to reason about multi-player systems. Two kinds of games are mainly studied and adopted: turn-based and concurrent. They differ on the way the players are allowed to move. However, in real scenarios, there are very simple interplays among players whose modeling does not fit well in any of these settings.

In this paper we introduce a novel game-based framework to model and reason about the interaction between robots and humans. This framework combines all positive features of both turn-based and concurrent games. Over this game model we study the reachability problem.

To give an evidence of the usefulness of the introduced framework, we use it to model the interaction between a human and a team of two robots, in which the former tries to run away from the latter. We also provide an algorithm that decides in polynomial time whether at least one robot catches the human.

## II. Introduction

In recent years, *game theory* has exhibited to be a fruitful metaphor in multi-agent system modeling and reasoning, where the evolution of the entire system emerges from the coordination of moves taken by all agents being involved [1], [20], [21], [22], [15], [23], [7], [8]. In the simplest setting, we consider finite-state games consisting of just two players (or agents), conventionally named *Player* 1 and *Player* 2. Depending on the possible interactions among the players, games can be either *turn-based* or *concurrent*. In the former case, the moves of the players are interleaved. In the latter case, instead, the players move simultaneously. In a turn-based game, the states of the game are partitioned into those belonging to Player 1 and those belonging to Player 2. When the game is at a state of Player $i$, only Player $i$ determines the next state. In a concurrent game, instead, the two players choose, simultaneously and independently, their moves and the next state of the game depends on the combination of such moves.

A game consists of two main objects, the *arena* and the *winning condition*. The arena is used to describe the players, the game states (configurations), and the possible evolution of the game in accordance to the moves the players can take. The winning condition is used to express the objective the players aim to achieve.

* This work is partially supported by the FP7 EU project 600958-SHERPA.

Solving a game corresponds to check whether a designed player has a winning strategy in the game, i.e. a sequence of moves that let him satisfy the winning condition no matter how the other players behave. In the literature, we distinguish between the case the condition is given as an "external entity", for example via a formula of a logic [1], [14], or internally as a condition over the states of the arena. While external conditions offer some modularity and allow to formalize very intricate targets, they require solutions with a very high complexity [12]. Internal conditions instead offer a good balance between expressiveness and complexity and this is the setting we consider in this paper.

A very simple and largely used (internal) winning condition consists of labeling some states of the arena as "*good*" and then consider as target the *reachability* of at least one of them. Properly speaking, the resulting setting is called a *reachability game*. These games have been exploited in both the (two-player) turn-based and (multi-player) concurrent settings and fruitfully applied in several interesting real scenarios. However, there are specific situations that cannot be casted in any of these settings. In particular, this happens when we want to model the interplay among agents with a different essence. This is the case, for example in human-robot interaction. To give an evidence of this necessity, we discuss along the paper a scenario involving the interaction between a human and two robots. The shape of the arena is a maze and the three players are initially placed in three different positions. The goal of the human is to run away from the robots, while the robots have the opposite goal. Therefore, looking at the game from the robots side, the good states are those in which at least one of the robots and the human sit in the same place in the same moment. We assume that whenever the human decides to move, none of the robots can interfere and vice-versa. In other words the human uses the game on its side as turn-based while the robots use it as being concurrent. We introduce a novel model framework that is able to represent efficiently such a scenario and provide a solution algorithm that can decide in polynomial time whether the robots have a winning strategy, i.e., they have a sequence of moves that, no matter how the human behaves, they reach a desired state.

**Related work.** Robotic technology is quickly advancing and this rapid progress inevitably is having a huge effect over the people. The interaction between human and robots is a complex issue that challenges both engineering and cognitive science. In several settings, such an interaction has been modeled in terms of a suitable interplay between all actors involved (see [5] for a survey). Several models in this context take inspiration from the field of open-system formal verifications [9], [19], [11], [16]. A system is considered *open* if it has an ongoing interaction with an external unpredictable environment and the system behavior fully depends on this interaction. To verify an open system means to check that the system is correct no matter how the environment behaves. Several models based on two-player games (system vs. environment) have been proposed in order to model such an interaction as well as suitable algorithms to check whether the system is correct (i.e. wins the game) [3], [24]. In this context, multi-agent games have been also proposed in order to model and reason about multiple players able to play in a cooperative or adversarial manner [1], [14].

The game setting we consider in this paper has several connections with planning problems as well [2], [4], [6]. Indeed, planning can be rephrased as the problem of synthesizing a strategy (the *plan*) for an agent to achieve a determined task within an environment. Often the environment is hostile and consists of an aggregation of several entities working together. By casting such a scenario in our model setting one can see the environment as the team of cooperative agents working against the one aiming for the planning.

Since the environment can be seen as an adversarial player the correlation the our setting follows

## III. Case Study

In this section we introduce a simple but effective human-robot interaction scenario that will be used along the paper as an application for the game-model framework we introduce. The scenario is described in the following and depicted in Figure 1.

The interaction takes place in a maze and involves tree players: a human $H$ and two robots $R_1$ and $R_2$. The goal of the human is to escape from both the robots. The robots work in team and aim just the opposite. For simplicity we assume the maze divided in square rooms and we start by considering that the players sit all in different rooms. All players from every room can access to an adjacent one unless there is a wall (drawn with a bold line in the figure). We assume to have an interleaving of moves between the human and the team of robots. Hence, the robots move simultaneously. We assume that all players can move in the four directions *up*, *down*, *left*, and *right*, unless the shape of the maze forbids it.

Let us now discuss how such an human-robot interaction can be rephrased in terms of a game. We make this reasoning more formal in the next section. Starting from an initial position in the maze, all players by taking the allowed moves can change their position. Each possible placing of the players can be seen as a *configuration* of the game and the starting one is usually denoted the *initial configuration*. Clearly, we can move in one step from one configuration to another only if we have moves that allow it. In particular, as seen from the human, moves are interleaved in a *turn-based* fashion, while they are taken in a *concurrent* way as seen by the robots. All legal moves can be collected by an opportune data structure or a table. Following the target of the robots, we have that the human loses if along

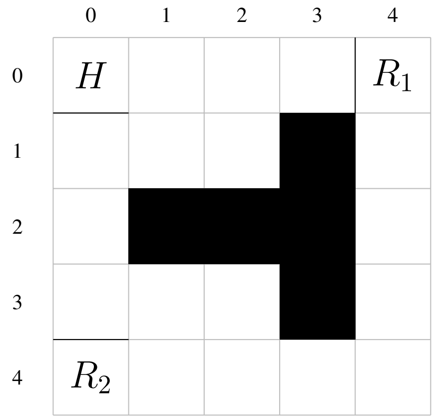


Fig. 1. Maze Game.

an interplay, the game reaches a configuration in which both he and one of the robots sit in the same room. Accordingly, we label all such configurations as good ones (as seen by the robots). Thus, deciding whether the team of robots can defeat the human corresponds to solve a *reachability game*.

It is important to note that the scenario we have discussed is neither (just) turn-based, as the robots move simultaneously, nor concurrent, as the human moves independently from the robots. Moreover, the discussed game involves three players and it is not trivially reducible to a two-player one because of the particular target: at least one of the robots has to catch the human.

To model this scenario, we introduce in the next section a novel game model in which all players, except a designed one, work in team and move simultaneously. The designed one instead will move alternately and independently from the other players.

## IV. The conceived model

In this section we introduce a novel multi-player game-based framework that is suitable to represent, under a unique structure, both the players moving turn-based and those acting concurrently. This framework, which we call *hybrid*, opportunely combines and extends the main features behind *concurrent game structures* [1] and two-player turn-based games (see [18] for an introduction).

*Definition 1 (Hybrid Game models):* A *hybrid reachability game structure* is a tuple $G = < Ag, Ac, St, s_0, tr, St_f >$, where $Ag$ is a finite non-empty set of *agents*, partitioned into two teams $Ag_0$ and $Ag_1$. $Ac$ and $St$ are enumerable non-empty sets of *actions* and *states*, respectively, and $s_0 \in St$ is a designated *initial state*. The set of states is partitioned in $St_0$ and $St_1$ as those states belonging to the teams $Ag_0$ or $Ag_1$, respectively. For $i \in \{0, 1\}$, let $Dc_i = Ag_i \rightharpoonup Ac$ to be the set of *decisions* of team $i$, i.e., partial functions describing the choices of an action by all agents in $Ag_i$. Then, $tr : Dc_i \times St_i \rightarrow St_{1-i}$ denotes the *transition function* mapping every pair of decisions $\delta \in Dc_i$ and state $s \in St_i$ for the team $i$ to a successive state over the deterministic graph belonging the the adversary team. Finally, we define $St_f \subseteq St$ the subset of *terminal* (or *accepting*) states.

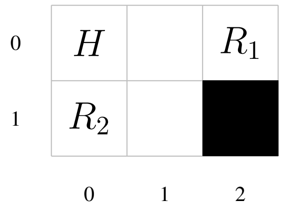


Fig. 2. A Reduced Maze Game.

We now show how the case study we have presented in the previous section can be easily and formally described by means of a hybrid game model $G$.

For a sake of clarity, instead of considering the scenario depicted in Figure 1, we consider a reduced one as reported in Figure 2. Also, we allow the robots to move only horizontally (left and right), while the human is still able to move in all directions. While this new scenario may look too naive, it will avoid a bunch of tedious calculations in the sequel. We model the human-robots interaction over this maze by setting $Ag_0$ as the team consisting of the unique player *human* and $Ag_1$ as the team of *robots* $R_1$ and $R_2$ . We consider the maze as a grid-board made by $K \times J$ positions, with $K = \{0, 1\}$ and $J = \{0, 1, 2\}$. In each position, zero, one, or more than one player can sit. States are just a set of positions of the three players plus a flag we use to recall which team is playing in that state. Formally, we have as set of states $St = ((K \times J)^3) \times \{0, 1\}$. $St_i$ contains those states having the flag equal to $i$. The initial state is just the initial position of the players.

Accordingly to the picture depicted in Figure 2, assuming $R_1$ and $R_2$ are the next to move, the state is $((0, 0), (1, 0), (0, 2), 1)$. We assume in our example that this is the initial state. The possible actions for the robots are set to $r$ for *right* and $l$ for *left*. For the possible actions of the human we set $u$ for *up*, $d$ for *down*, $l$ for *left*, and $r$ for *right*. Decisions are defined accordingly and must respect the limitations imposed by the shape of the maze. As far as the set of accepting states concerns, recall that the target of the robots is to reach a configuration where at least one of them catches the human, being both siting in the same position. This means that $St_f$ must contain all those states in which the first pair of coordinates (corresponding to the position of the human) is equal to the second or third pair. Formally, $St_f = \{((a, b), (c, d), (e, f), i) \ |(a = c \text{ and } b = d) \text{ OR } (a = e \text{ and } b = f)\}$. Finally, it remains to define the transition relation. For the sake of exposition, we only report the part corresponding to the team $Ag_0$. Note that this is coherent with the shape of the maze.

$$tr(\delta, ((i, j)(k, l)(m, n)), 0) =$$

$$\begin{cases} (((i-1, j)(k, l)(m, n)), 1), & \text{if } \delta = u \text{ and } i > 0; \\ (((i, j-1)(k, l)(m, n)), 1), & \text{if } \delta = l \text{ and } j > 0; \\ (((i, j+1)(k, l)(m, n)), 1), & \text{if } \delta = r \text{ and } j < 2; \\ (((i+1, j)(k, l)(m, n)), 1), & \text{if } \delta = d \text{ and } i < 1. \end{cases}$$

To better clarify the meaning of the above formalization, let us discuss some examples over the maze. From the initial state, which belongs to the team $Ag_1$, by using the decision $lr$, the game moves to the state $(((0, 0)(0, 1)(1, 1)), 0)$. In accordance with the flag, this is now a state belonging to the team $Ag_0$ and thus this team (the human) takes the turn to move. From this state, the human agent has two available moves, that are $d$ and $r$. In the first case the game moves in the state $(((1, 0)(0, 1)(1, 1)), 1)$ and in the second case it moves in the state $(((0, 0)(0, 1)(1, 1)), 1)$, both belonging to the players in the coalition $Ag_1$. And so on. In Figure 3, we report the computations of the game. It is easy to observe that the accepting states are $((1, 0), (0, 2)(1, 0)), 0)$, $((1, 0), (0, 0)(1, 0)), 0)$ and $((0, 1), (0, 1)(1, 1)), 1)$ since one of the two robots and the human are both in the room.

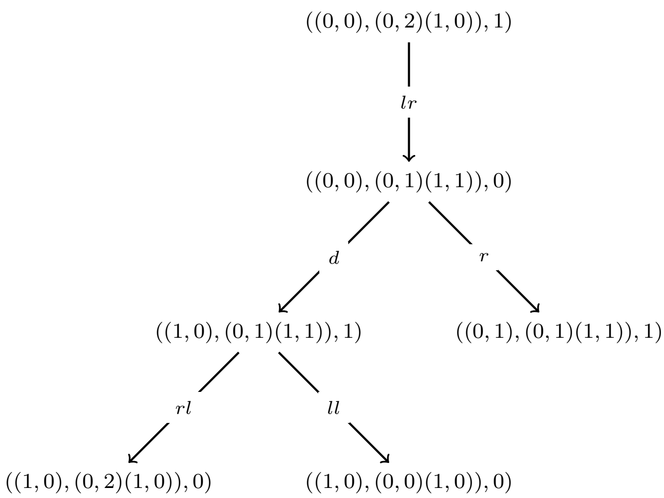


Fig. 3. A game model $G$ for the simplify maze in Figure 2.

## V. The proposed solution

Under the conceived model, we can handle all possible targets that can be represented in terms of *reachability*, i.e., the players in the coalition $Ag_1$ set some configurations of the game as "good" and aim to reach them. These configurations are those represented by states $St_f$ in the model. The coalition $Ag_1$ *wins* the game if its players have a *winning strategy*, i.e., they can force the game, by means of a sequence of moves, to reach a state in $St_f$, *no matter* how the players in the team $Ag_1$ behave. *Deciding a game* means deciding whether the coalition $Ag_1$ wins the game.

In this section we provide an algorithm to decide a game under the hybrid framework we have introduced. This algorithm aims to find the set of states of the model from which the players in the coalition $Ag_1$ win the game, that is the set of states $reach_1(St_f)$. As the complement of this set contains the states from which players in the coalition $Ag_0$ win the game, as a corollary we obtain that our game model is zero-sum (i.e. from each node only one team can win the game).

The algorithm proceeds as follows. We start from the set $St_f$ containing all winning states for players in $Ag_1$. Then, in a backward manner, we recursively build the set

$reach_1^i(St_f)$ containing all states $s \in St$ such that players in $Ag_1$ can force a visit from $s$ to a state in the set $St_f$ in less than $i$ steps. Formally, we have that $reach_1^i(St_f) = \{s \in St \,|Ag_1$ can force a visit from $s$ to $St_f$ in less than $i$ moves$\}$.

Formally, we have the following.

$$reach_1^0(St_f) = St_f;$$

$$reach_1^{i+1}(St_f) = reach_1^i(St_f) \cup$$
$$\{s \in St_1 | \exists s' \in St : tr(Dc_1, s) = s' \wedge s' \in reach_1^i(St_f)\} \cup$$
$$\{s \in St_0 | \forall s' \in St : tr(Dc_0, s) = s' \wedge s' \in reach_1^i(St_f)\}.$$

In words, from the set $reach_1^i(St_f)$ we select all states that have incident edges in this set. From each of these states, say $s$, if it belongs to the coalition $Ag_1$, then, this state is immediately added to $reach_1^{i+1}(St_f)$ (as we *may* move from $s$ to $reach_1^i(St_f)$ and thus reach $St_f$). Conversely, if $s$ is a state belonging to the coalition $Ag_0$, then it is added to $reach_1^{i+1}(St_f)$ only if all outgoing edges from $s$ fall in $reach_1^i(St_f)$ (i.e., from $s$, players in $Ag_0$ are forced to move to $reach_1^i(St_f)$).

Finally we have that

$$reach_1(St_f) = reach_1^{|St|}(St_f).$$

As the calculation of $reach_1(St_f)$ requires a number of iterations linear in number of states $St$, we have that the whole algorithm requires at most quadratic time in the size of the model, as reported in the following theorem.

*Theorem 1:* Given a hybrid reachability game, it can be decided in quadratic time.

By a matter of calculation, one can see that by applying the algorithm above over our reduced case study, the coalition $Ag_1$ wins the game from every state. In fact, for each state of the model, there exists always a winning strategy for the players in the team $Ag_1$.

## VI. Discussion and Future Works

In the last years, human-robots interaction is receiving large attention in several research fields. An important aspect in this study is to come up with appropriate models to design and reasoning about how such interactions can take place and how they affect the future behavior of the involved actors. In this setting, *game theory* is a powerful framework that is able to formalize the interplay between the human and the robots in a very natural way.

In this paper, we have introduced a game model framework that allows to represent and reasoning about scenarios in which the interaction between the humans and the robots results in a hybrid two-team game: the game between the two teams is turn-based while all players in each team play concurrently among them. We have observed that classic turn-based and concurrent games are not powerful enough to model such a setting. Over the conceived model, we study the reachability problem and show that it is solvable in quadratic time. Therefore, with no extra cost with respect to classic turn-based and concurrent games. We give an evidence of the usefulness of the introduced framework by means of a case study.

This work opens to several future directions. First, it would be interesting to reasoning about quantitative aspects about the human-robots interaction. For example, to determine what is the best strategy to perform.

Another direction would be to consider other possible winning conditions. In particular, one could import some winning conditions studied in formal verification such as the *Büchi* and the *parity* ones (see [17], [10] for an introduction) or external winning conditions given in terms of formulas of a suitable logic. Some scenarios along these lines have been already investigated in the case of turn-based and concurrent game settings [9], [19] and showed to have some useful applications [13], [14]. We plan, as future work, to investigate them in the settings of multi-robots and human-robots interactions.

## References


[1] R. Alur, T. Henzinger, and O. Kupferman. Alternating-Time Temporal Logic. *Journal of the ACM*, 49(5):672–713, 2002.

[2] D. Calvanese and G. De Giacomo and M.Y Vardi. Reasoning about actions and planning in LTL action theories. *KR*, 2:593–602, 2002.

[3] L. de Alfaro, T. Henzinger, and O. Kupferman. Concurrent reachability games. In *Foundations of Computer Science, 1998. Proceedings. 39th Annual Symposium on*, pages 564–575. IEEE, 1998.

[4] G. De Giacomo and M. Y. Vardi. Automata-theoretic approach to planning for temporally extended goals. pages 226–238, 2000.

[5] M. A. Goodrich and A. C. Schultz. Human-robot interaction: a survey. *Foundations and trends in human-computer interaction*, 1(3):203–275, 2007.

[6] H. Geffner and B. Bonet. A concise introduction to models and methods for automated planning. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, 8(1):1–141, 2013.

[7] J. Gutierrez and M. Wooldridge. Equilibria of concurrent games on event structures. In *Proceedings of the Joint Meeting of the Twenty-Third EACSL Annual Conference on Computer Science Logic (CSL) and the Twenty-Ninth Annual ACM/IEEE Symposium on Logic in Computer Science (LICS)*, page 46. ACM, 2014.

[8] J. Van Benthem. *Logic games: From tools to models of interaction*. Institute for Logic, Language and Computation (ILLC), University of Amsterdam, 2007.

[9] W. Jamroga and A. Murano. On Module Checking and Strategies. In *Autonomous Agents and MultiAgent Systems'14*, pages 701–708. International Foundation for Autonomous Agents and Multiagent Systems, 2014.

[10] O. Kupferman, M. Vardi, and P. Wolper. An Automata Theoretic Approach to Branching-Time Model Checking. *Journal of the ACM*, 47(2):312–360, 2000.

[11] O. Kupferman, M. Vardi, and P. Wolper. Module Checking. *Information and Computation*, 164(2):322–344, 2001.

[12] F. Mogavero, A. Murano, G. Perelli, and M. Vardi. Reasoning About Strategies: On the Model-Checking Problem. *Transactions On Computational Logic*, 15(4):34:1–42, 2014.

[13] F. Mogavero, A. Murano, and L. Sorrentino. On Promptness in Parity Games. In *Logic for Programming Artificial Intelligence and Reasoning'13*, LNCS 8312, pages 601–618. Springer, 2013.

[14] F. Mogavero, A. Murano, and M. Vardi. Reasoning About Strategies. In *Foundations of Software Technology and Theoretical Computer Science'10*, LIPIcs 8, pages 133–144. Leibniz-Zentrum fuer Informatik, 2010.

[15] S. Parsons and M. Wooldridge. Game theory and decision theory in multi-agent systems. *Autonomous Agents and Multi-Agent Systems*, 5(3):243–254, 2002.

[16] P. Schobbens. Alternating-Time Logic with Imperfect Recall. 85(2):82–93, 2004.

[17] W. Thomas. Automata on infinite objects. *Handbook of theoretical computer science*, 2, 1990.

[18] W. Thomas. Monadic Logic and Automata: Recent Developments. 1998.

[19] W. Jamroga and A. Murano. Module checking of strategic ability. pages 227–235, 2015.

[20] M. Wooldridge. Intelligent Agents. In G. Weiss, editor, *Multiagent Systems. A Modern Approach to Distributed Artificial Intelligence*. MIT Press: Cambridge, Mass, 1999.

[21] M. Wooldridge. Computationally Grounded Theories of Agency. In *International Conference on MultiAgent Systems'00*, pages 13–22. IEEE Computer Society, 2000.

[22] M. Wooldridge. *An Introduction to Multi Agent Systems*. John Wiley & Sons, 2002.

[23] Y. Jiang and J. Hu and D. Lin. Decision making of networked multiagent systems for interaction structures. *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on*, 41(6):1107–1121, 2011.

[24] W. Zielonka. Infinite Games on Finitely Coloured Graphs with Applications to Automata on Infinite Trees. *Theoretical Computer Science*, 200(1-2):135–183, 1998.