Designing Autonomous Social Agents under the Adversarial Risk Analysis Framework

Pablo G. Esteban[1] and David Ríos Insua[2]

[1] Rey Juan Carlos University of Madrid
pablo.gomez.esteban@urjc.es
[2] Royal Academy of Sciences, Spain
david.rios@urjc.es

Abstract. We describe how the Adversarial Risk Analysis framework may be used to support the decision making of an autonomous agent which needs to interact with other agents and persons. We propose several contextualizations of the problem and suggest which is the conceptual solution in some of the proposed scenarios.

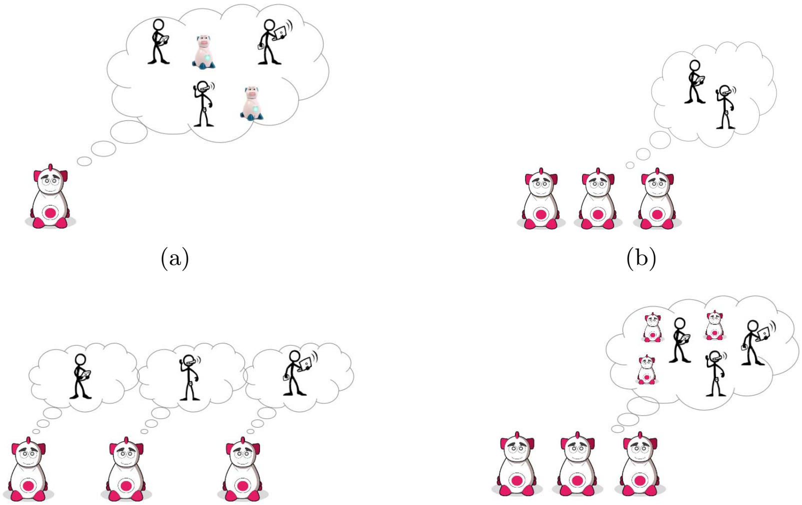
Keywords: Game Theory, Adversarial Risk Analysis, Multi-agent systems, Intelligent Agents.

1 Introduction

In [1], we have described a behavioural model for an autonomous decision agent which processes information from its sensors, facing an intelligent adversary using multi-attribute decision analysis at its core, complemented by models forecasting the decision making of the adversary. We call Adversarial Risk Analysis (ARA) to this framework, see [2]. Generally speaking, ARA views a two-person game through two coupled influence diagrams, one for the supported agent and one for the adversary. The supported agent would build an explicit model for the decision-making of the adversary. Given such model, the supported agent may simulate outcomes under it, which will draw on subjective probabilities about the adversary's beliefs, preferences, capabilities and resources. Following such approach, we avoid the standard and unrealistic game theoretic assumptions of common knowledge, through a nested hierarchy of decision analysis models. From the point of view of supporting our agent, the problem is understood as a decision analytic one, see [3], but we consider principled procedures which employ the adversarial structure to forecast the adversary's actions and the evolution of the environment surrounding both of them, therefore, embracing also adaptability: the agent performs as best as it can, given the circumstances. On doing this, the agent would forecast what the other participant thinks about him, thus starting the above mentioned hierarchy. Depending on the level the agent climbs up in such hierarchy, we would talk about a 0-level analysis, 1-level analysis and so on, borrowing the k-level thinking terminology, see [4], [5] and [6]. Our approach has a Bayesian game theoretic flavor, as in [7] and [8].

J.M. Corchado et al. (Eds.): PAAMS 2013 Workshops, CCIS 365, pp. 292–303, 2013.

This model has been implemented within an AISoy1 robot, see [9]. In this paper, we shall refer to multi-agent systems, exploring the social needs of our robotic agent, and how it handles interactions with other agents, both human and robotic ones. We have in mind four possible scenarios, shown in Fig. 1. On the top left, Fig. 1(a), we consider a single agent facing multiple adversaries (agents and users). On its right, Fig. 1(b), several agents compete in their interaction with several users. At the bottom left, several agents, each of them related with only one user, compete in a global scenario, see Fig. 1(c). Finally, bottom right, there are multiple agents cooperating to satisfy themselves and the users, see Fig. 1(d).



(a) (b) (c) (d)

Fig. 1. Different scenarios take into account

Throughout this paper, we shall explore the interaction among different agents and users, within the scenarios outlined above. Due to space limitations we shall only describe in detail the two first scenarios (Figs. 1(a) and 1(b)), briefly introducing the third one (Fig. 1(c)). Our motivation is the design of societies of robotic agents that interact among them and with one or more users. Those agents may be used as interactive robotic pets, robotic babysitters and teaching assistants or cooperative caregivers for the elderly.

The paper is structured as follows. In Section 2, we provide the basic model for a single agent facing a single adversary. In Section 3, we consider a case in which a decision agent is identifying several users and robotic agents, and makes decisions depending on the adversary is facing, scenario (1(a)). Next, in Section 4, we define two cases of a society of competitive robots which interact with humans, scenarios (1(b)) and (1(c)). For comparative purposes, we deal with them through the standard game theoretic and the novel ARA frameworks. We remain at a conceptual level, describing the solution concepts, although we outline the required modeling. Finally, in Section 5, we end up with some discussion.

2 The Basic Model

We briefly describe, as a starting point, the model in [1] which supports the decision making of an agent A facing a user B. This model will serve as a basis for later elaborations. A and B make decisions, respectively a and b, within finite sets $\mathcal{A}$ and $\mathcal{B}$, which possibly include a *do nothing* action. They are placed within an environment E which changes with the user's actions, adopting a state e within a set $\mathcal{E}$. Essentially, we plan our agent's activities over time within the decision analytic framework, see [3], including models to forecast the adversary behaviour (Adversarial Risk Analysis) and the evolution of the environment. Note that we could view the problem within the game-theoretic framework, see [10], but with our alternative approach we avoid the much debated common knowledge assumptions, see [8] or [11].

Assume that, for computational reasons, we just forecast one period ahead based on a two period memory. We are interested in computing, at each time t,

$$\begin{aligned} p(e_t, b_t \mid a_t, (e_{t-1}, a_{t-1}, b_{t-1}), (e_{t-2}, a_{t-2}, b_{t-2})) = \\ = p(e_t \mid a_t, b_t, (e_{t-1}, a_{t-1}, b_{t-1}), (e_{t-2}, a_{t-2}, b_{t-2})) \times \\ \times\, p(b_t \mid a_t, (e_{t-1}, a_{t-1}, b_{t-1}), (e_{t-2}, a_{t-2}, b_{t-2})) \ , \end{aligned} \quad (1)$$

which forecast the reaction of the user and the evolution of the environment, given the agent action, and the recent history. This constitutes the adversarial part of the model. The first term in (1) will be simplified to

$$p(e_t \mid b_t, e_{t-1}, e_{t-2}) \ ,$$

which we call the *environment model*, thus assuming that the environment is fully under control by the user. The second term in (1) will be simplified to

$$p(b_t \mid a_t, b_{t-1}, b_{t-2}) \ . \quad (2)$$

The agent will maintain two models, M_i with $i \in \{1, 2\}$, in relation with (2). The first one, M_1, describes the evolution of the user by himself, assuming that he is not affected by the agent's actions. We call it the *user's model* and describe it through

$$p(b_t \mid b_{t-1}, b_{t-2}) \ .$$

The second one, M_2, refers to the user's reactions to the agent's actions, which we describe through

$$p(b_t \mid a_t) \ .$$

We call it the *classical conditioning model*, with the agent possibly conditioning the user. We combine both models to recover (2), through model averaging, see [12]:

$$p(b_t \mid a_t, b_{t-1}, b_{t-2}) =$$
$$= p(M_1)\ p(b_t \mid b_{t-1}, b_{t-2}) + p(M_2)\ p(b_t \mid a_t)\ ,$$

where $p(M_i)$ denotes the probability that the agent gives to model M_i, with $p(M_1) + p(M_2) = 1,\ p(M_i) \geq 0$.

Assume that the agent faces multiple consequences $c = (c_1, c_2, \ldots, c_l)$, that will be of the form $c_i(a_t, b_t, e_t),\ i = 1, \ldots, l$. We shall assume that they are evaluated through a multi-attribute utility function, see [3]. Specifically, we adopt an additive form

$$u(c_1, c_2, \ldots, c_l) = \sum_{i=1}^{l} w_i u_i(c_i)\ ,$$

with $w_i \geq 0,\ \sum_{i=1}^{l} w_i = 1$, where u_i represents the robot's i-th component utility function and w_i represent the corresponding utility weight.

Our agent aims at maximizing the predictive expected utility, i.e. implements the alternative solving

$$\max_{a_t \in \mathcal{A}} \psi(a_t) = \int \int u(c(a_t, b_t, e_t)) \times [p(e_t|b_t, e_{t-1}, e_{t-2})\ p(b_t|a_t, b_{t-1}, b_{t-2})]\, db_t de_t\ .$$

Planning $(r+1)$ instants ahead follows a similar parth, but may turn out to be very expensive computationally.

For details on the implementation of this model, including learning, forecasting and decision making, see [1].

3 Supporting an Agent Facing Several Agents and Users

In this Section, we extend our basic model to a case in which the agent faces several adversaries, which may be agents or users, see Fig. 1(a). As an example, assume that our agent (A) is supporting two children (B_1 and B_2) in their daily school assignments, so that, A should be able to identify who is who, to evaluate how correctly each of them is working, and deliver the corresponding score and support.

For that purpose, the agent must be able to identify the adversary he is facing and will have different forecasting models in relation with each of the known opponents. We assume that the agent will face just one adversary at each of the time steps of the scheme described in Fig. 2.
Using some identification method, the agent will guess who is the user/agent it is dealing with and adapt its behaviour accordingly. The difference between facing another agent or a user would essentially be the set of actions available for the corresponding adversary forecasting model. Adversary identification is not a core element of our work. For that purpose we could base the identification of the opponent B_x on eigenface recognition algorithms, see [13] for a face recognition survey, and implement it with the OpenCv libraries, see [14], as we have done.

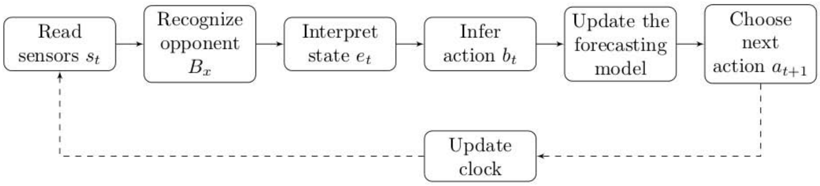


Fig. 2. Agent loop with advesary recognition

We assume that the user is that which maximizes $p(B_x|D_t)$, after obtaining an image of the face of the participant. Our agent will not make any difference among robotic agents as there is no physical difference among them, because we assume they are all robots of the same type.

3.1 Model

As in Section 2, our agent A makes decisions within a finite set $\mathcal{A}$. In this case, there are r adversaries $B_1, \ldots, B_r$ which interact with A. An index x will be used to identify the corresponding adversary. As B_x may be an agent or a user, he makes decisions within the set $\mathcal{A}$, in case it is an agent, or a set $\mathcal{B}$ that will designate the set of available actions to the users, which we assume are the same for all of them.

The agent decision model is similar to that in Section 2. However in this case, the forecasting model is conditional on the guessed adversary, so that (1) becomes

$$\begin{aligned} & p(e_t, b_t \mid a_t, (e_{t-1}, a_{t-1}, b_{t-1}), (e_{t-2}, a_{t-2}, b_{t-2}), B_x) = \\ & = p(e_t \mid b_t, a_t, (e_{t-1}, a_{t-1}, b_{t-1}), (e_{t-2}, a_{t-2}, b_{t-2}), B_x) \times \\ & \times p(b_t \mid a_t, (e_{t-1}, a_{t-1}, b_{t-1}), (e_{t-2}, a_{t-2}, b_{t-2}), B_x) \ . \end{aligned} \tag{3}$$

Using a similar decomposition, now

$$p(e_t \mid b_t, a_t, (e_{t-1}, a_{t-1}, b_{t-1}), (e_{t-2}, a_{t-2}, b_{t-2}), B_x) = p(e_t \mid b_t, e_{t-1}, e_{t-2}, B_x) \ , \tag{4}$$

and

$$p(b_t \mid a_t, (e_{t-1}, a_{t-1}, b_{t-1}), (e_{t-2}, a_{t-2}, b_{t-2}), B_x) = p(b_t \mid a_t, b_{t-1}, b_{t-2}, B_x) \ .$$

We should note that, when B_x is a robotic agent, the environment model (4) would become $p(e_t \mid e_{t-1}, e_{t-2}, B_x)$, because the agent's action does not affect the environment.

Again, we view this as a problem of model averaging, for each agent B_x:

$$p(b_t \mid a_t, b_{t-1}, b_{t-2}, B_x) =$$

$$= p(M_1|B_x) \ p(b_t \mid b_{t-1}, b_{t-2}, B_x) + p(M_2|B_x) \ p(b_t \mid a_t, B_x) \ ,$$

where $p(M_i|B_x)$ denotes the probability that the agent gives to model M_i, assuming that the adversary is B_x, with $p(M_1|B_x)+p(M_2|B_x) = 1,\ p(M_i|B_x) \geq 0$. Finally, we shall use model averaging over users, defined through

$$p(e_t, b_t \mid a_t, (e_{t-1}, a_{t-1}, b_{t-1}), (e_{t-2}, a_{t-2}, b_{t-2})) =$$
$$= \sum_{B_x} \Big[p(e_t \mid b_t, e_{t-1}, e_{t-2}, B_x) \times p(b_t \mid a_t, b_{t-1}, b_{t-2}, B_x) \times p(B_x) \Big] \ .$$

The core of the *classical conditioning model* and the *adversary's model* remains as before. In our implementation, we use two matrix-beta models, see [15], to store the corresponding data, a $n \times m$ matrix for the classical conditioning model and a $n \times n \times n$ matrix for the adversary's model, as the set $\mathcal{A}$ had m elements and set $\mathcal{B}$ had n. As the robot faces users and agents, the size of the data structures would be different depending on the type of adversary it is dealing with. This corresponds to a 0-level implementation in that we only appeal to past behaviour of the adversary, possibly as a response to our previous behaviour.

We include also some details about the preference model. As described in [1], each agent aims at satisfying five objectives which are: *being charged*, *being secure*, *being taken into account*, *being accepted* and *being updated*. The first and the last objectives would remain unvaried within the multiagent model, but the other three should be extended to face several users and agents. Generally speaking, those objectives and subobjectives which refer to inference of user's actions, should take into account the actions of each user within the scenario, specifically,

$$u_{21}(attack) = \begin{cases} 1, & \text{if none of the users attacked} \\ 0, & \text{otherwise} \end{cases} ,$$

and

$$u_{41}(play) = \begin{cases} 1, & \text{if the robot inferred a user or another agent} \\ & \text{playing around} \\ 0, & \text{otherwise} \end{cases} .$$

Some subobjectives ought to be extended to include agents' actions as well as users' actions, as it is the case of:

$$u_{313}(asked\ to\ play) = \begin{cases} 1, & \text{if the robot is asked to play by the user} \\ & \text{or by another agent} \\ 0, & \text{otherwise} \end{cases} ,$$

where *asked to play* refers to detecting an order to play from the user, including the game's title, or a request for playing by another agent (action a_8: *ask for playing*). For additional details, see [1]. The expected utility model would remain the same as in Section 2.

4 Supporting an Agent Competing with Other Agents

We deal now with two competing scenarios in which agents interact with one or more users. In the first case, several agents compete among them to be selected by the users, so that the competition is among the agents. In the second case, each agent interacts with its own user forming a team. Each tandem agent-user will compete against the other participating teams. For comparison, both cases are solved computing the corresponding Nash Equilibria (NE) and under the ARA framework. We assume that there is communication among the agents. Moreover, under the NE framework, we shall assume that there is a computerised trusted third party (CTTP) that would handle the conflict, computing the NE when needed. This may be an external computer or one of the robotic agents that could adopt the role of trusted party. Working under the ARA framework we shall not make such assumptions, but, for convenience, we shall allow agents to communicate.

There will be two different types of communication: among the participating robotic agents and each of them with the CTTP. The agents would be periodically transferring information to interact with each other. Whenever a conflict arises, participating agents would send their beliefs, matrices and parameters, as well as their utilities to the CTTP, who would compute the required solution and send back the corresponding strategies to each participating agent.

For both models, the preference model and the expected utility model would be the same as in Section 3.

4.1 Supporting an Agent within a Society of Competing Agents

In this case, several agents compete among them to accomplish an identical goal, involving users in the scene, see Fig. 1(b). As an example, consider a case in which there are three robots (A, B, C) and two kids (X, Y) in a scene. The kids want to play "Simon says". They would like to have at least one more player to do so. All agents want to play with the kids, but just one of them will play. The robotic agents would compete to be chosen as the third player, being nicer, funnier or whatever, in order to be selected.

Model. We have several agents, under a competing attitude, facing simultaneously one or several users within an environment. To fix the discussion, assume that, as in the example, we have three agents (A, B, C) and two users (X, Y). Agents will perform actions a_{A_t}, a_{B_t} and a_{C_t}, respectively, whereas users will perform b_{X_t} and b_{Y_t} actions, respectively.

We use again a multi-attribute utility function. However, in this case the consequences will depend on the actions of all agents and users:

$$c_i(a_{A_t}, a_{B_t}, a_{C_t}, b_{X_t}, b_{Y_t}, e_t) \ ,$$

for $i = 1, \dots, l$, where e_t is the environmental state as in Section 2. The utility that the agents will obtain will be, respectively:

$$u_A(a_{A_t}, a_{B_t}, a_{C_t}, b_{X_t}, b_{Y_t}, e_t), \quad u_B(a_{A_t}, a_{B_t}, a_{C_t}, b_{X_t}, b_{Y_t}, e_t),$$

$$u_C(a_{A_t}, a_{B_t}, a_{C_t}, b_{X_t}, b_{Y_t}, e_t) \ .$$

We next describe the forescasting model for agent A,

$$p_A(a_{B_t}, a_{C_t}, b_{X_t}, b_{Y_t}, e_t \mid \tag{5}$$
$$\mid a_{A_t}, (a_{A_{t-1}}, a_{B_{t-1}}, a_{C_{t-1}}, b_{X_{t-1}}, b_{Y_{t-1}}, e_{t-1}), (a_{A_{t-2}}, a_{B_{t-2}}, a_{C_{t-2}}, b_{X_{t-2}}, b_{Y_{t-2}}, e_{t-2})) \ .$$

Assuming that e_t remains exclusively under the users' control, (5) will be decomposed as:

$$p_A(e_t \mid b_{X_t}, b_{Y_t}, e_{t-1}, e_{t-2}) \times p_A(a_{B_t}, a_{C_t}, b_{X_t}, b_{Y_t} \mid$$
$$\mid a_{A_t}, (a_{A_{t-1}}, a_{B_{t-1}}, a_{C_{t-1}}, b_{X_{t-1}}, b_{Y_{t-1}}, e_{t-1}), (a_{A_{t-2}}, a_{B_{t-2}}, a_{C_{t-2}}, b_{X_{t-2}}, b_{Y_{t-2}}, e_{t-2})) \ .$$

Note that in the scheme described in Fig. 2, we assumed that each agent chooses its action depending on the action performed by the user, so that when several agents face the same user, the actions performed by them would be considered simultaneous. For that reason, when our agent A is facing another agent, we assume that the forecasted action of the robotic agent will depend only on the actions previously perfomed by itself and the action of the agent A. Users' actions will depend on all agent's actions. Equation (5) then becomes:

$$p_A(e_t \mid b_{X_t}, b_{Y_t}, e_{t-1}, e_{t-2}) \times p_A(a_{B_t} \mid a_{B_{t-1}}, a_{B_{t-2}}, a_{A_{t-1}}) \times \tag{6}$$
$$\times\ p_A(a_{C_t} \mid a_{C_{t-1}}, a_{C_{t-2}}, a_{A_{t-1}}) \times p_A(b_{X_t} \mid a_{A_t}, a_{B_t}, a_{C_t}, b_{X_{t-1}} b_{X_{t-2}}) \times$$
$$\times\ p_A(b_{Y_t} \mid a_{A_t}, a_{B_t}, a_{C_t}, b_{Y_{t-1}}, b_{Y_{t-2}}) \ .$$

Finally, we find out that our forecasting models for agent A are: the first term of (6) (the *environmental model*), and the rest of (6) which is the model to forecast the adversaries' actions. This second term in (6) will be decomposed in the *adversary models* and the *classical conditioning model*, similarly to what we did in Section 2. The adversary models would be those in which the forecasted action depends on the evolution of their own behaviour as, e.g.:

$$p_A(a_{B_t} \mid a_{B_{t-1}}, a_{B_{t-2}}) \text{ and } p_A(b_{X_t} \mid b_{X_{t-1}}, b_{X_{t-2}}) \ .$$

The classical conditioning models would be those reflecting the reaction to our agent's behaviour as, e.g.:

$$p_A(a_{B_t} \mid a_{A_{t-1}}) \text{ and } p_A(b_{X_t} \mid a_{A_t}) \ .$$

They are combined through model averaging techniques. Forecasting other agents' actions shall be defined as forecasting the user's actions in Section 2, evaluating the evolution of its own behaviour and how reactive is to agent A's actions:

$$p_A(a_{B_t} \mid a_{B_{t-1}}, a_{B_{t-2}}, a_{A_{t-1}}) = \tag{7}$$
$$= p(M_1^B) p_A(a_{B_t} \mid a_{B_{t-1}}, a_{B_{t-2}}) + p(M_2^B) p_A(a_{B_t} \mid a_{A_{t-1}}) \ ,$$

with $\sum_i p(M_i^B) = 1$, $p(M_i^B) \geq 0$, and, similarly, for $p_A(a_{C_t} \mid a_{C_{t-1}}, a_{C_{t-2}}, a_{A_{t-1}})$. In this case, forecasting the users' actions would be extended to include the reaction of the user to the actions of every agent:

$$p_A(b_{X_t} \mid a_{A_t}, a_{B_t}, a_{C_t}, b_{X_{t-1}} b_{X_{t-2}}) = p(M_1^X) p_A(b_{X_t} \mid b_{X_{t-1}}, b_{X_{t-2}}) +$$
$$+ p(M_2^X) p_A(b_{X_t} \mid a_{A_t}) + p(M_3^X) p_A(b_{X_t} \mid a_{B_t}) + p(M_4^X) p_A(b_{X_t} \mid a_{C_t}) \ ,$$

with $\sum_i p(M_i^X) = 1,\ p(M_i^X) \geq 0$, and, similarly, for $p_A(b_{Y_t} \mid a_{A_t}, a_{B_t}, a_{C_t}, b_{Y_{t-1}}, b_{Y_{t-2}})$. Note that

$$p(M_i^X \mid D_t) = \frac{p(D_t \mid M_i^X) p(M_i^X)}{\sum_{i=1}^4 p(D_t \mid M_i^X) p(M_i^X)},\ i = 1, \ldots, 4 \ .$$

Computing Nash Equilibria. As we are in a competitive scenario, we are dealing with selfish agents so that each agent will aim at maximizing its expected utility. For example, when A implements a_{A_t}, and the other agents implement a_{B_t} and a_{C_t}, agent A's expected utility would be:

$$\psi_A(a_{A_t}, a_{B_t}, a_{C_t}) = \int\int\int u_A(a_{A_t}, a_{B_t}, a_{C_t}, b_{X_t}, b_{Y_t}, e_t) \times$$
$$\times \Big[p_A(b_{X_t} \mid a_{A_t}, a_{B_t}, a_{C_t}, b_{X_{t-1}} b_{X_{t-2}}) \times p_A(b_{Y_t} \mid a_{A_t}, a_{B_t}, a_{C_t}, b_{Y_{t-1}} b_{Y_{t-2}}) \times$$
$$\times\ p_A(e_t \mid b_{X_t}, b_{Y_t}, e_{t-1}, e_{t-2}) \Big] \ db_{X_t} db_{Y_t} de_t \ ,$$

and, analogously, for the other agents. As we pointed out above, we assume that a CTTP would play the role of a trusted party solving the existing conflicts, and there will be communication among the agents. Each agent would send its beliefs, matrices, parameters and utilities, so that, in our example, the CTTP will have available ψ_A, ψ_B and ψ_C which would be common knowledge. Then, the CTTP would compute the Nash Equilibria with methods described, e.g. in [16] or [17].

ARA Solving agents Let us write the problem from the ARA framework point of view. In this case, communication is not required. Under this framework, we are supporting one of the agents (say, agent A), to make a decision facing several users (X and Y) and other agents (B and C). The agent will aim at maximizing its expected utility based on forecasts of the other agents defined through

$$\max_{a_{A_t}} \ \psi_A(a_{A_t}) = \int\int \psi_A(a_{A_t}, a_{B_t}, a_{C_t}) \times \Big[p_A(a_{B_t} \mid a_{B_{t-1}}, a_{B_{t-2}}, a_{A_{t-1}}) \times$$
$$\times\ p_A(a_{C_t} \mid a_{C_{t-1}}, a_{C_{t-2}}, a_{A_{t-1}}) \Big] \ da_{B_t} da_{C_t} \ ,$$

where the relevant probability models were described in (7). In a 0-level approach, we may use matrix-beta models to implement these.

4.2 Agent Supporting a User within a Competitive Society of Users

In this case, each agent is interacting with its own user, supporting her within a competition against other user-agent teams, see Fig. 1(c). As an example, consider a case in which three teams are involved, couples robot A - child X, robot B - child Y and robot C - child Z. Each of the teams work on school assignments willing to be chosen as the favourite by the teacher and get the highest grade. Each agent shall support its own user in making decisions, forecasting what the other agents would do. Assumptions similar to those in the previous Section will be made here.

Model. We will have several agents, under a competing attitude, supporting simultaneously their corresponding user within an environment. To fix the discussion, assume that we have three agents (A, B, C) and three users (X, Y, Z) forming agent-user teams. Agents will perform actions a_{A_t}, a_{B_t} and a_{C_t}, whereas users will perform b_{X_t}, b_{Y_t} and b_{Z_t} actions.

We use again a multi-attribute utility function. In this case, the consequences will depend on the actions of all agents and the supported user. The consequences that agent A would face when it is supporting user X are:

$$c_A(a_{A_t}, a_{B_t}, a_{C_t}, b_{X_t}, e_t) \ ,$$

where e_t is the environmental state, as described in Section 2. The utilities that the agents will obtain in our example will be, respectively:

$$u_A(a_{A_t}, a_{B_t}, a_{C_t}, b_{X_t}, e_t), \quad u_B(a_{A_t}, a_{B_t}, a_{C_t}, b_{Y_t}, e_t),$$

$$u_C(a_{A_t}, a_{B_t}, a_{C_t}, b_{Z_t}, e_t) \ .$$

The forescasting model for agent A would be

$$p_A(a_{B_t}, a_{C_t}, b_{X_t}, e_t \mid \\ \mid a_{A_t}, (a_{A_{t-1}}, a_{B_{t-1}}, a_{C_{t-1}}, b_{X_{t-1}}, e_{t-1}), (a_{A_{t-2}}, a_{B_{t-2}}, a_{C_{t-2}}, b_{X_{t-2}}, e_{t-2})) \ . \quad (8)$$

Simplifications and assumptions related to the forecasting models would be analogous to those of the previous case. Equation (8) ends up decomposed in:

$$p_A(e_t \mid b_{X_t}, e_{t-1}, e_{t-2}) \ ,$$

the *environmental model*, and

$$p_A(a_{B_t} \mid a_{B_{t-1}}, a_{B_{t-2}}, a_{A_t}) \times p_A(a_{C_t} \mid a_{C_{t-1}}, a_{C_{t-2}}, a_{A_t}) \times \\ \times \ p_A(b_{X_t} \mid a_{A_t}, b_{X_{t-1}} b_{X_{t-2}}) \ , \quad (9)$$

the model to forecast the adversary's action, which will be decomposed in the *adversary model* and the *classical conditioning model*, as we did in the previous scenario, then combined, through model averaging techniques.

Computing Nash Equilibria. Again, we are dealing with selfish agents so that each agent will aim at maximizing expected utility. Agent A's expected utility will be

$$\psi_A(a_{A_t}, a_{B_t}, a_{C_t}) = \int \int u_A(a_{A_t}, a_{B_t}, a_{C_t}, b_{X_t}, e_t) \times \\ \times \left[p_A(b_{X_t} \mid a_{A_t}, b_{X_{t-1}} b_{X_{t-2}}) \times p_A(e_t \mid b_{X_t}, e_{t-1}, e_{t-2}) \right] \ db_{X_t} de_t \ ,$$

and, similarly, for the other agents. As in the previous scenario, we assume that a CTTP would solve the existing conflicts, and there will be communication among the agents, so that ψ_A, ψ_B and ψ_C would have common knowledge. The Nash Equilibria may be computed as described in the previous scenario.

ARA Solving agents From the ARA perspective, we are supporting one of the agents (agent A) facing his own user (X) and other agents (B and C), so that, it shall be willing to maximize its expected utility, defined through

$$\max_{a_{A_t}} \psi_A(a_{A_t}) = \int\int \psi_A(a_{A_t}, a_{B_t}, a_{C_t}) \times \Big[p_A(a_{B_t} \mid a_{B_{t-1}}, a_{B_{t-2}}, a_{A_t}) \times \\ \times\, p_A(a_{C_t} \mid a_{C_{t-1}}, a_{C_{t-2}}, a_{A_t})\Big] \; da_{B_t} da_{C_t} \ .$$

The relevant probability models are described in (9). As before, we use matrix-beta models for their 0-level implementation.

5 Discussion

We have described different scenarios in which a decision agent is facing several adversaries (human and robotic ones).

As future work, we have two more scenarios to develop. Within the first one, we aim at supporting a society of agents, where n agents would like to behave cooperatively towards one or several users, see Fig. 1(d). As an example, suppose three robotic agents that want to support their children with their corresponding weekly school assignments, trying to emulate a cooperative environment in the school. They are under a cooperative attitude, so that they would look for helping the child together to find the best solution that satisfy their common goal. Within the other scenario, moving from the competing towards cooperating attitude shall be studied: agents will then modify their behaviour depending on their experience. To do such a thing, we should define two types of behaviour: selfish and cooperative. Based on certain parameters, the agent would move from a cooperative attitude to a competitive one, or viceversa. Note also, that the ARA models proposed here correspond to 0-level thinking and we could explore 1-level and 2-level thinking ideas.

The field of cognitive processes has recently shown that emotions may have a direct impact on decision-making processes, see e.g. [18]. Advances in areas such as affective decision making [19], neuroeconomics [20] and affective computing [21] are based on this principle. Following this, a potential future work, concerning these models will be addressed towards providing a model for an autonomous agent that makes decisions influenced by emotional factors when interacting with humans and other agents. Our aim with this would be to make interactions between humans and agents more fluent and natural.

Acknowledgments. Research supported by grants from the MICINN project RIESGOS, the RIESGOS-CM project and the INNPACTO project HAUS. We are grateful to discussion with Diego García, from AiSoy Robotics S.L., Jesus Ríos and David Banks.

References

1. Rázuri, J.G., Esteban, P.G., Insua, D.R.: An adversarial risk analysis model for an autonomous imperfect decision agent. In: Guy, T.V., Kárný, M., Wolpert, D.H. (eds.) Decision Making and Imperfection. SCI, vol. 474, pp. 165–190. Springer, Heidelberg (2013)
2. Ríos Insua, D., Ríos, J., Banks, D.: Adversarial risk analysis. Journal of the American Statistical Association 104(486), 841–854 (2009)
3. Clemen, R.T., Reilly, T.: Making Hard Decisions with Decision Tools. Duxbury, Pacific Grove (2004)
4. Stahl, D.O., Wilson, P.W.: On players models of other players: Theory and experimental evidence. Games and Economic Behavior 10(1), 218–254 (1995)
5. Banks, D., Petralia, F., Wang, S.: Adversarial risk analysis: Borel games. Applied Stochastic Models in Business and Industry 27, 72–86 (2011)
6. Kadane, J.B.: Adversarial risk analysis: What's new, what isn't?: Discussion of adversarial risk analysis: Borel games. Journal Applied Stochastic Models in Business and Industry 27(2), 87–88 (2011)
7. Kadane, J.B., Larkey, P.D.: Subjective probability and the theory of games. Management Science 28(2), 113–120 (1982)
8. Raiffa, H.: Negotiation Analysis: The Science and Art of Collaborative Decision Making. Press of Harvard University Press, Cambridge (2007)
9. AISoyRobotics (2010), http://www.aisoy.es
10. Aliprantis, C., Chakrabarti, S.: Games and Decision Making. Oxford University Press (2010)
11. Lippman, S., McCardle, K.: Embedded nash bargaining: Risk aversion and impatience. Decision Analysis 9, 31–41 (2012)
12. Hoeting, J., Madigan, D., Raftery, A., Volinsky, C.: Bayesian model averaging: A tutorial. Statistical Science 4, 382–417 (1999)
13. Zhao, W., Chellappa, R., Phillips, P.J., Rosenfeld, A.: Face recognition: A literature survey. ACM Comput. Surv. 35(4), 399–458 (2003)
14. Hewitt, R.: Seeing With OpenCV, Part 4: Face Recognition With Eigenface (2007)
15. Ríos Insua, D., Ruggeri, F., Wiper, M.: Bayesian Analysis of Stochastic Process Models. Wiley (2012)
16. Nisan, N., Roughgarden, T., Tardos, E., Vazirani, V.V.: Algorithmic Game Theory. Cambridge University Press (2007)
17. Menache, I., Ozdaglar, A.: Network Games: Theory, Models, and Dynamics. Morgan and Claypool Publishers (2011)
18. Busemeyer, J.R., Dimperio, E., Jessup, R.K.: Integrating emotional processes into decision-making models, pp. 213–229. Oxford University Press, New York (2006)
19. Loewenstein, G., Lerner, J.S.: The role of affect in decision making. In: Davidson, R., Scherer, K., Goldsmith, H. (eds.) Handbook of Affective Science, pp. 619–642. Oxford University Press, Oxford (2003)
20. Glimcher, P.W., Camerer, C., Poldrack, R.A., Fehr, E.: Neuroeconomics: Decision Making and the Brain. Academic Press (2008)
21. Picard, R.W.: Affective Computing. MIT Press, Cambridge (1997)