Chapter 2
Designing Societies of Robots

Pablo G. Esteban and David Ríos Insua

Abstract We provide a framework to model competition and cooperation within a group of agents. Competition is dealt with through adversarial risk analysis, which provides a disagreement point and, implicitly, through minimum distance to such point. Cooperation is dealt with through a concept of maximal separation from the disagreement point. Mixtures of both problems are used to refer to in-between behavior. We illustrate the ideas with several experiments in relation with groups of robots.

2.1 Introduction

Personal robots are becoming increasingly present in our daily lives, helping us at museums, airports or even at work and home as personal assistants and companions. The long-term aim of this work is to design an autonomous emotional decision making agent capable of interacting with several persons and agents. This means that our agent will learn the appropriate behavior based on its own experience and the interactions with users and other agents. It will decide its actions based on its system of values, incorporating emotional elements, and on the impact it has on its surrounding environment. Such agents may be used e.g. as interactive robotic pets, robotic babysitters and teaching assistants or cooperative care-givers for the elderly.

In [21], we described a behavioral 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 this the basic Adversarial Risk Analysis (ARA) framework, see [19].

P.G. Esteban (✉)
Robotics and Multi-Body Mechanics Research Group, Vrije Universiteit Brussel, Pleinlaan 2, 1050 Brussels, Belgium
e-mail: pablo.gomez.esteban@vub.ac.be

D.R. Insua
Instituto de Ciencias Matemáticas ICMAT-CSIC, Madrid, Spain
e-mail: david.rios.insua@gmail.com

T.V. Guy et al. (eds.), *Decision Making: Uncertainty, Imperfection, Deliberation and Scalability*, Studies in Computational Intelligence 538,
DOI 10.1007/978-3-319-15144-1_2

In this chapter, we refer to multi-agent systems, see [27], exploring the social needs of our agent and how it handles interactions with several agents, both human and robotic ones, considering competitive as well as cooperative scenarios. Within competitive scenarios, the agent faces conflicts through ARA models (and implicitly minimizes the distance to the ARA solution). Conversely, within cooperative scenarios, we use maximum separation from the disagreement point. As agents may evolve from a cooperative to a competitive attitude, and vice versa, we introduce a parametric model that mixes both models allowing for such evolution.

The chapter is structured as follows. In Sect. 2.2 we consider a case in which a decision agent identifies several users and agents and competes with these in their interaction with humans. In Sect. 2.3, we present a method to compute cooperative solutions within a society of agents. Then, we describe the evolution from a competitive to a cooperative attitude, see Sect. 2.4. Finally, we provide some computational experience with a set of robots in Sect. 2.5 and end up with a discussion.

2.2 Supporting a Competing Agent

We assume that several agents compete to accomplish a certain goal involving users in a scene. Traditionally, the favored solution within such environments is Nash equilibria and related concepts, see [8, 15] or [14], but this typically requires too strong common knowledge assumptions. We shall rather use ARA concepts, which avoid such assumption through an explicit Bayesian model of the capabilities, probabilities and utilities of the adversaries.

In ARA, we aim at supporting one of the agents who will use a decision analytic approach to solve its decision-making problem. It will aim at maximizing expected utility, taking into account a random model over the probabilities and utility functions of its opponents. This is developed using a hierarchy of nested models of decision processes, following a Bayesian version of level-k thinking, see [22]. This level-k hierarchy is indexed by how deep the player thinks its opponents' decision-making processes are. If the agent behaves randomly, it is a level-0 thinker; if the agent behaves as its opponents are level-0 thinkers, it is a level-1 thinker; and so on. For this, it needs to forecast the actions of the other agents and, consequently, the outcomes which it and its opponents will receive as a result of their interaction. This can be viewed as a Bayesian approach to games, as initiated, in non-constructive ways, by [12, 16, 17]. The approach has been criticized in [10] or [15], among others. The main obstacle to operationalizing such analysis has been the lack of mechanisms that allow the supported decision maker to encode its subjective probabilities about all components in its opponents' decision making. We described a first approach to such problem within a robotics context in [7]. Here we extend such approach. Other ideas may be seen in [19, 20].

Consider a set of r agents $A_1, A_2, \ldots, A_r$, possibly in presence of a set of q users $B_1, B_2, \ldots, B_q$, within an environment E. At each planning instant t, the agents will perform their respective actions $a_t = (a_{1t}, a_{2t}, \ldots, a_{rt})$, all of them in a finite set $\mathscr{A}$,

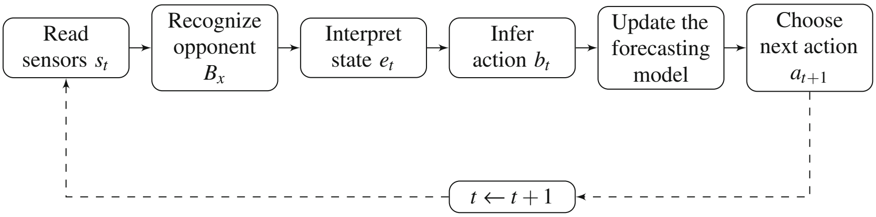


Fig. 2.1 Agent loop with adversary recognition

whereas users will implement their corresponding actions $b_t = (b_{1t},\ b_{2t}, \ldots, b_{qt})$, all of them in a finite set $\mathscr{B}$. Both $\mathscr{A}$ and $\mathscr{B}$ will typically include a *do nothing* action. The (multi-attribute) utilities that the r agents will obtain will be, respectively:

$$u_1(a_t, b_t, e_t),\ \ u_2(a_t, b_t, e_t), \ldots,\ u_r(a_t, b_t, e_t).$$

Thus, each agent receives a utility which depends not only on what it has implemented, but also on what the other agents and users have done, as well as on the resulting environmental state $e_t \in \mathscr{E}$, which we also assume to be finite.

With respect to the agent's decision model, we assume that the agent faces just one adversary at each of the time steps of the scheme described in Fig. 2.1, which we detail in later sections.

Using some identification method, e.g. based on voice and/or vision, the agent will guess which is the user/agent it is dealing with and adapt its behavior accordingly. The difference between facing another agent or a user would essentially be the set of actions available for the corresponding adversary forecasting model. We assume that agent A_i computes the probabilities $p_i(B_x|D_t)$ of various adversaries B_x faced, both users or agents, given the data D_t available.

Assume we support agent A_1. For computational reasons, we limit the agent's memory to only the previous two periods, i.e. at $(t-1)$ and $(t-2)$. Then, the forecasting model will be of the form

$$p_1(e_t, b_t, a_{-1t} \mid a_{1t}, (e_{t-1}, a_{t-1}, b_{t-1}), (e_{t-2}, a_{t-2}, b_{t-2}), D_t), \qquad (2.1)$$

where a_{-1t} would be the actions of all agents performed at time t, excluding that of our supported agent action a_{1t}. Thus, it aims at forecasting the reaction b_t of users, the evolution e_t of the environment and the actions a_{-1t} of the other agents, given the action of the supported agent a_{1_t} and the recent history $(e_{t-1}, a_{t-1}, b_{t-1})$ and $(e_{t-2}, a_{t-2}, b_{t-2})$. We shall drop the D_t dependence from now on to simplify the notation.

We shall condition (2.1) on the faced adversary through

$$p_1(e_t, b_t, a_{-1t} \mid a_{1t}, (e_{t-1}, a_{t-1}, b_{t-1}), (e_{t-2}, a_{t-2}, b_{t-2})) =$$
$$= \sum_{B_x} \Big[p_1(e_t, b_t, a_{-1t} \mid a_{1t}, (e_{t-1}, a_{t-1}, b_{t-1}), (e_{t-2}, a_{t-2}, b_{t-2}), B_x) \times p(B_x) \Big]. \tag{2.2}$$

From now, we refer to each of the terms in the summation in (2.2) dropping the dependence on B_x, except when convenient for expository reasons. By standard computations, each term $p_1(e_t, b_t, a_{-1t} \mid a_{1t}, (e_{t-1}, a_{t-1}, b_{t-1}), (e_{t-2}, a_{t-2}, b_{t-2}))$ in (2.2) becomes

$$p_1(e_t \mid b_t, a_t, (e_{t-1}, a_{t-1}, b_{t-1}), (e_{t-2}, a_{t-2}, b_{t-2})) \times$$
$$\times \; p_1(b_t, a_{-1t} \mid a_{1t}, (e_{t-1}, a_{t-1}, b_{t-1}), (e_{t-2}, a_{t-2}, b_{t-2})).$$

We assume that the environment remains exclusively under the users' control who may solely manipulate light, sound, temperature and other environmental variables. Then, this becomes

$$p_1(e_t \mid b_t, e_{t-1}, e_{t-2}) \times p_1(b_t, a_{-1t} \mid a_{1t}, (e_{t-1}, a_{t-1}, b_{t-1}), (e_{t-2}, a_{t-2}, b_{t-2})). \tag{2.3}$$

The second term in (2.3) may be decomposed taking into account the forecasting models for the adversaries involved in the scene. Note that when our supported agent A_1 faces a robotic agent A_j, the forecasted action a_{jt} of such agent, will depend on the last action of agent A_1, since we consider the agents to act simultaneously. On the other hand, the users' actions will depend on all of the incumbent agent's actions a_t. Thus, (2.3) is decomposed as

$$p_1(e_t \mid b_t, e_{t-1}, e_{t-2}) \times \prod_{k=1}^{q} p_1(b_{kt} \mid a_t, b_{k(t-1)}, b_{k(t-2)}) \times$$
$$\times \prod_{j=2}^{r} p_1(a_{jt} \mid a_{j(t-1)}, a_{j(t-2)}, a_{1(t-1)}), \tag{2.4}$$

where we remove dependence on (e_{t-1}, e_{t-2}) in the last two groups of factors because we assume adversaries prioritize making their decisions given others' actions, rather than reacting to environmental changes. When forecasting the k-th user action the supported agent will maintain several models M_i^k with $i \in \{0, \ldots, r\}$, in connection with the second group of factors in (2.4). The first model, M_0^k, describes the evolution of the user by herself, assuming that she is in control of the whole environment and is not affected by the agent actions. The rest of them, M_i^k with $0 < i \leq r$, refers to the user reactions to the various agents' actions. We combine them using model averaging, see [5, 11]:

$$
\begin{aligned}
p_1(b_{kt} \mid a_t, b_{k(t-1)} b_{k(t-2)}) = p(M_0^k) p_1(b_{kt} \mid b_{k(t-1)}, b_{k(t-2)}) \\
+ \sum_{j=1}^{r} p(M_j^k) p_1(b_{kt} \mid a_{jt}),
\end{aligned}
\tag{2.5}
$$

with $\sum_{i=0}^{r} p(M_i^k) = 1,\ p(M_i^k) \geq 0$.

Similarly, when forecasting the j-th agent actions, $j \in \{2, \ldots, r\}$, the supported agent will maintain two models N_i^j with $i \in \{0, 1\}$, in connection with the third group of factors in (2.4). The first model, N_0^j, describes the evolution of the incumbent robotic agent, assuming that it is not affected by any other agent's actions. The second one, N_1^j, refers to the j-th agent's reaction to the agent A_1's actions. Again, they are combined through model averaging:

$$
\begin{aligned}
p_1(a_{j_t} \mid a_{1_{t-1}}, a_{j_{t-1}}, a_{j_{t-2}}) = p(N_0^j) p_1(a_{j_t} \mid a_{j_{t-1}}, a_{j_{t-2}}) \\
+ p(N_1^j) p_1(a_{j_t} \mid a_{1_{t-1}}),
\end{aligned}
\tag{2.6}
$$

with $p(N_0^j) + p(N_1^j) = 1,\ p(N_i^j) \geq 0$, for each agent $j \neq 1$.

In summary, the components of the forecasting model for agent A_1 are: the first term in (2.4), called the *environment model*, and the rest, which are models to forecast the adversaries' (agents and users) actions. The *environment model* is described in [21] and comprises variables referring to the battery level, temperature, inclination, sound, presence of an identified adversary, light and being touched. Regarding forecasting of adversaries' actions, we consider that each opponent may be reactive or independent to our supported agent A_1: each adversary forecasting model will be decomposed into the *adversary* and the *classical conditioning models*, respectively.

As we are in a competitive scenario, each agent aims at maximizing its expected utility. When the agents implement $a_t = (a_{1t},\ a_{2t}, \ldots,\ a_{rt})$, agent A_1's expected utility will be:

$$
\begin{aligned}
\psi_1(a_t) = \int \ldots \int u_1(a_t, b_t, e_t) \\
\times\ p_1(e_t \mid b_t, e_{t-1}, e_{t-2}) \times \prod_{k=1}^{q} p_1(b_{kt} \mid a_t, b_{k(t-1)}, b_{k(t-2)})\, \mathrm{d}b_{1t} \ldots \mathrm{d}b_{qt} \mathrm{d}e_t .
\end{aligned}
$$

The agent will aim at maximizing its expected utility based on forecasts of the other agents defined through

$$
\max_{a_{1t}} \psi_1(a_{1t}) = \int \ldots \int \psi_1(a_t) \left[\prod_{j=2}^{r} p_1(a_{jt} \mid a_{1(t-1)}, a_{j(t-1)}, a_{j(t-2)}) \right] \mathrm{d}a_{2t} \ldots \mathrm{d}a_{rt} .
\tag{2.7}
$$

The relevant probability models regarding users' and agents' actions are described in (2.5) and (2.6), respectively.

The solution of this problem provides the maximum expected utility $f_{1_t}^*$ that the agent A_1 may achieve by thinking about itself and forecasting what the other agents would do, as well as the corresponding optimal action $a_{1_t}^*$, which is the one that the agent should implement.

2.3 Supporting Cooperative Agents

We focus now on cooperative cases: several agents collaborate to find out the solution that best satisfy them when interacting with users in achieving a specific task. We assume that a Computerized Trusted Third Party (CTTP) plays the role of an arbitrator solving the cooperative game. There will be communication among the agents and with the arbitrator. Each agent individually aims at maximizing its expected utility as in (2.7), which would be sent to the CTTP.

Once the CTTP has received ψ_j, for each agent j, we may use cooperative game theory, to find the solution within this scenario. There are different methods within that framework, see [24, 25] for reviews, including the Nash Bargaining and the Kalai-Smorodinsky solutions. We shall use a method that finds a solution maximizing the distance to the ARA solutions or, more generally, to a disagreement point.

A cooperative game is defined by a tuple $(F,\ d)$. F is the set of attainable (expected) utilities by the agents. In our case, $F = \{x \in R^r : x = (\psi_1(a), \ldots, \psi_r(a)), \text{for } a \in \mathscr{A}^r\}$, thus being finite. The set F will be changing over time since it depends on the forecasting models of the agents, which evolve dynamically, therefore modifying the expected utilities. $d = (d_1, \ldots, d_r)$ is the disagreement point, i.e. the pre-specified utilities obtained when there is no agreement among the agents. d will also typically change over time, as F does. By repeating the procedure in Sect. 2.2 for each participant, we obtain f_j^*, $j = 1, \ldots, r$. If $f^* = (f_1^*, \ldots, f_r^*)$ belongs to F, then f^* will play the role of disagreement solution d. Note, however, that since f_j^* is solved unilaterally, it could be the case that $f^* \notin F$. In such case, we could solve the problem $d = \arg\min_{x \in F}(L_p(x, f^*))$ for some L_p distance, and use d as the disagreement point.

If, otherwise, an agreement is reached, the alternative chosen is the solution concept of the game, defined by $\phi_j(F, d)$, for each agent j. For our approach, we stem from the classic cooperative game solution in [28], which looks for minimizing an L_p distance to an ideal point. Based on that idea, given the disagreement point d, we shall look for a point $x \in F$, with $x \geq d$, which maximizes an L_p distance from the disagreement point.

$$\phi(F, d) = \arg \max_{\substack{s.t.\ x \in F \\ x \geq d}} L_p(x, d) = \arg \max_{\substack{s.t.\ x \in F \\ x \geq d}} \left[\sum_{j=1}^{r} (x_j - d_j)^p \right]^{1/p} .$$

Note that the set $F \cap \{x \geq d\}$ will be non-empty, since, at least, d belongs to such set. Therefore, the solution is well defined. Note that, intuitively, since d is a disagreement solution, and we are promoting agreement, we want to separate as much as possible from it.

Note that when $p = 1$, the optimization problem is equivalent to

$$\arg \max_{\substack{s.t.\ x \in F \\ x \geq d}} \sum_{j=1}^{r} x_j,$$

which corresponds to the utilitarian solution, see [23]. When, $p = \infty$, the optimization problem is

$$\arg \max_{j \in R} \max \ (x_j - d_j),$$

thus aiming at maximizing the maximum payoff for the agents. A validation of this solution concept in axiomatic terms may be seen in [6], in which we compare it with the Nash Bargaining and Kalai-Smorodinsky solutions.

2.4 Competition or Cooperation?

As described above, agents may compete or cooperate among them to reach their objectives and goals. As we are referring to autonomous agents, we expect them to choose when and how to cooperate or compete, forming an autonomous society.

To model this possibility, each agent j would have two parameters λ_{j1} and λ_{j2}, with $\lambda_{j1}, \lambda_{j2} \geq 0$ and $\lambda_{j1} + \lambda_{j2} = 1$. The parameter λ_{j1} refers to the cooperativeness of the agent, whereas λ_{j2} refers to its competitiveness. Such parameters may be influenced by different factors, including the agent's experience, as in Sect. 2.5.5. Depending on such factors, the agent will modify its behavior. Let us imagine a scenario in which the parameters depend on the opponents' actions. Suppose that most agents are attacking or ignoring the j-th agent. Then, A_j will not likely want to behave cooperatively, but competitively, so that λ_{j1} will be close to 0 (and λ_{j2} will be close to 1).

As there is communication between the agents and the CTTP, each agent j will send its parameters λ_{j1} and λ_{j2}. We assume that the agents operate under the FOTID (Full, Open, and Truthful Intermediary Disclosure) framework in [18]. The CTTP would compute an average value of those parameters to find the *society's attitude towards cooperation*, e.g. through

$$\lambda_k = \frac{1}{r} \sum_{j=1}^{r} \lambda_{jk}, \ k \ = \ 1, \ 2.$$

Note that $\lambda_1, \lambda_2 \geq 0$ and $\lambda_1 + \lambda_2 = 1$. As we are interested in combining cooperative and competitive behavior, we propose the following solution concept

$$\phi(F, d) = \arg \max_{\substack{s.t.\ x \in F \\ x \geq d}} \Big(\lambda_1 \, L_p(x, d) - \lambda_2 \, L_q(x, d)\Big), \tag{2.8}$$

for given L_p and L_q distances, with $p \neq q$.[1] Depending on these parameters, λ_1 and λ_2, the proposed method shall allow the agents to modify their social behavior. The CTTP will compute the solution concept of such game, sending back the suggested agreement to the involved agents in the game, which would accept it in an arbitration sense.

Indeed, under a fully cooperative environment, $\lambda_{j1} = 1$ for each agent j, the society will have parameters $\lambda_1 = 1$ and $\lambda_2 = 0$, and (2.8) becomes:

$$\phi(F, d) = \arg \max_{\substack{s.t.\ x \in F \\ x \geq d}} L_p(x, d),$$

corresponding to the solution concept in Sect. 2.3. Similarly, under a fully competitive environment, $\lambda_{j1} = 0$ for each agent j, the society parameters will be $\lambda_1 = 0$ and $\lambda_2 = 1$. The arbitrator would then solve

$$\phi(F, d) = \arg \max_{\substack{s.t.\ x \in F \\ x \geq d}} \big(- \max L_q(x, d)\big) = \arg \min_{\substack{s.t.\ x \in F \\ x \geq d}} L_q(x, d),$$

whose solution is the disagreement point d.

When at least one of the agents is not fully cooperative, neither fully competitive, $0 < \lambda_{j1} < 1$, and $0 < \lambda_1, \lambda_2 < 1$. We then have a mixed behavior as we illustrate through an example, see Fig. 2.2. Given the set F of alternatives, marked by white points, and the disagreement point $d = (0.3, 0.3)$, in red, we look for the solutions of the game (represented in green and pointed by an arrow) when we change the cooperativeness and competitiveness parameters. Points attaining objective function level 1 in (2.8) are shown in grey. We distinguish three cases: in Fig. 2.2a, the solution is $x^* = (0.7, 0.6)$, and this happens whenever the cooperativeness parameter is $\lambda_1 \geq 0.5$; in Fig. 2.2b, the solution is $x^* = (0.55, 0.55)$, and this happens whenever $0.3 < \lambda_1 < 0.5$. Finally, in Fig. 2.2c, the solution is $x^* = d = (0.3, 0.3)$, which happens whenever $\lambda_1 \leq 0.3$.

[1] This is made to allow for behavior in between cooperation and competition. If $p = q$, then $\phi(F, d) = \arg\max(\lambda_1 - \lambda_2) L_p(x, d)$ which leads to the fully competitive or the fully cooperative solution, depending on the sign of $(\lambda_1 - \lambda_2)$.

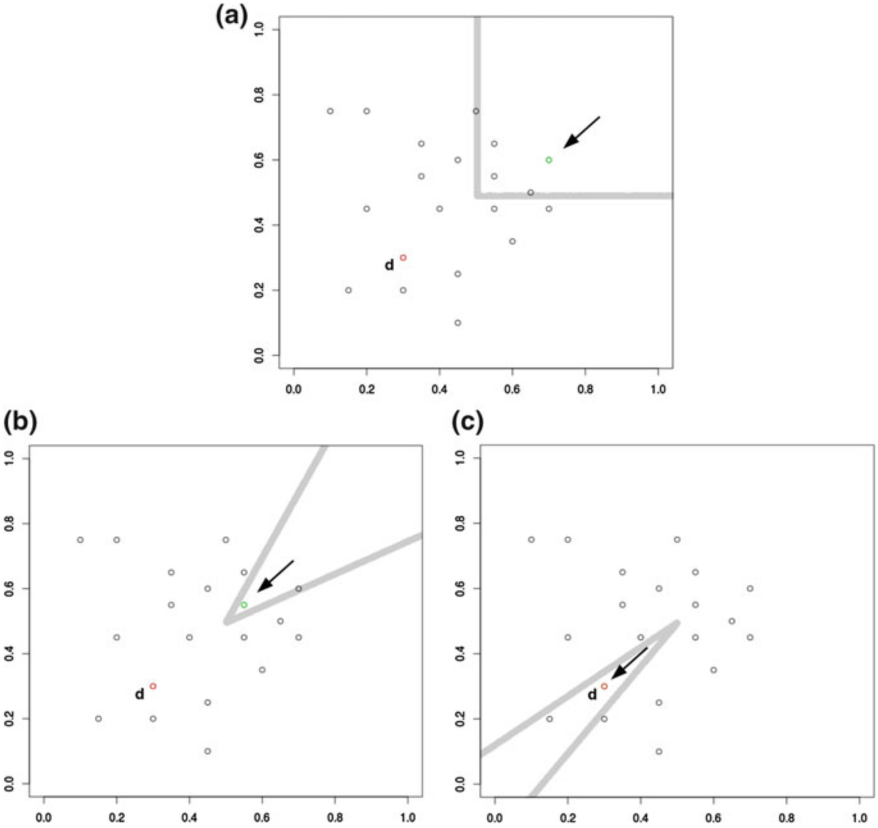


Fig. 2.2 The compromise solution varies when **a** $\lambda_1 \geq 0.5$; **b** $0.3 < \lambda_1 < 0.5$; **c** $\lambda_1 \leq 0.3$

2.5 Computational Experience

In this Section, we assess the solution concepts presented in Sects. 2.2, 2.3 and 2.4, using simulations with the robotic platform AiSoy1, see [1]. This platform includes as sensors a camera to detect objects or persons within a scene; a microphone used to recognize when the user talks and understand what she says, through an ASR (Automatic Speech Recognition) component; several touch sensors to interpret when it has been stroked or attacked; an inclination sensor so as to know, whether or not, the robot is in vertical position; a light sensor and a temperature sensor. As actuators, it includes several servos that allow it to move some parts of its body, but it mostly uses a text-to-speech system (TTS) combined with a led matrix to simulate a mouth when talking. Using an RGB led in the middle of its body, it is capable of showing different colors that symbolize the predominant emotion at that moment. It is based on a Raspberry Pi board.

2.5.1 Basic Setting

We simulate an environment in which a user (B_1) is simultaneously interacting with two robotic agents (A_1 and A_2). Both agents make their decisions based on the ARA framework described in Sect. 2.2 considering their opponents as non-strategic thinkers. Thus, they are in a first level of the level-k thinking hierarchy, as in [22]. We assume that the user interacts with both agents simultaneously. They will start with the same level of battery and environmental conditions.

The agents have fifteen available alternatives for implementation at each time step with $\mathscr{A} = \{a_1, a_2, a_3, a_4, a_5, a_6, a_7, a_8, a_9, a_{10}, a_{11}, a_{12}, a_{13}, a_{14}, a_{15}\} = \{$*ask for help, salute, ask for charging, complain, play, speak, ask for playing, ask for shutting down, tell jokes, tell events, obey, flatter, offend, apologize, do nothing*$\}$. On the user's side, based on the sensor readings, the agents are able to infer sixteen user actions with $\mathscr{B} = \{b_1, b_2, b_3, b_4, b_5, b_6, b_7, b_8, b_9, b_{10}, b_{11}, b_{12}, b_{13}, b_{14}, b_{15}, b_{16}\} =$ $\{$*recharge, stroke, flatter, apologize, attack, offend, move, blind, shout, discharge, speak, ignore, order, play, update the robot software, do nothing*$\}$.

To simulate the user's behavior we make the following assumptions:

- Whenever an agent *asks for charging*, the user will actually *recharge* it.
- At least, the user will perform *update* once per 10,000 iterations, to simulate periodical software updates.
- To simulate a user which pays some attention to the robot, 50 % of time the user will behave reactively to our agents' actions. For example, if the agent performs $a_4 =$ *complain*, the user would randomly choose an action among the set $\{$*flatter, speak, play, stroke, apologize, recharge, ignore, do nothing*$\}$, probably looking for cheering the agent up.
- Otherwise, the action performed by the user will be randomly generated as explained below.

2.5.2 Forecasting Models

As the agents are performing at level-1, we may use matrix-beta models with prior and posterior Dirichlet distributions for the adversary and the classical conditioning models in Sect. 2.2.

Assume that we are supporting agent A_1, which will face user B_1 and agent A_2. For the adversary models we have, for user B_1, a posterior Dirichlet distribution

$$p_1(b_{1t} = b_k \mid b_{1(t-1)} = b_j, b_{1(t-2)} = b_i, D_t) \sim Dir(\rho^{B_1}_{ij1} + h^{B_1}_{ij1}, \ldots, \rho^{B_1}_{ijn} + h^{B_1}_{ijn}),$$

with $b_{1t} \in \mathscr{B}$, and $h^{B_1}_{ijk}$ designating the number of occurrences in which B_1 performed $b_{1t} = b_k$ after having implemented $b_{1(t-1)} = b_j$ and $b_{1(t-2)} = b_i$; and $\rho^{B_1}_{ijk}$ are the prior parameters with $\rho_{ijk}^{B_1} > 0$ for $i, j, k = 1, \ldots, n$. In case the adversary is agent A_2,

$$p_1(a_{2t} = a_k \mid a_{2(t-1)} = a_j, a_{2(t-2)} = a_i, D_t) \sim Dir(\rho_{ij1}^{A_2} + h_{ij1}^{A_2}, \ldots, \rho_{ijm}^{A_2} + h_{ijm}^{A_2}),$$

with $a_{2t} \in \mathscr{A}$ and $h_{ijk}^{A_2}$ designating the number of occurrences in which agent A_2 performed $a_{2t} = a_k$ after having implemented $a_{2(t-1)} = a_j$ and $a_{2(t-2)} = a_i$; and $\rho_{ijk}^{A_2}$ are the prior parameters with $\rho_{ijk}^{A_2} > 0$ for $i, j, k = 1, \ldots, m$. The required data will be stored in a three-dimensional matrix, where the last row accumulates the sum of row values for each column. See [21] for additional explanation.

For the classical conditioning models, we have, for the human opponent B_1, a posterior Dirichlet distribution

$$p_1(b_{1t} = b_i \mid a_{1t} = a_j, D_t) \sim Dir\,(\beta_{1j}^{B_1} + h_{1j}^{B_1}, \ldots, \beta_{nj}^{B_1} + h_{nj}^{B_1}),\;\; b_{1t} \in \mathscr{B},$$

where, similarly, $h_{ij}^{B_1}$ designates the number of occurrences when the user implemented $b_{1t} = b_i$ after having observed our supported agent A_1 performing $a_{1t} = a_j$; and $\beta_{ij}^{B_1}$ are the prior parameters with $\beta_{ij}^{B_1} > 0$ for $i = \{1, \ldots, n\}$. The classical conditioning model for agent A_2 would be, analogously,

$$p_1(a_{2t} = a_i \mid a_{1t} = j, D_t) \sim Dir\,(\beta_{1j}^{A_2} + h_{1j}^{A_2}, \ldots, \beta_{mj}^{A_2} + h_{mj}^{A_2}),\;\; a_{2t} \in \mathscr{A},$$

where $h_{ij}^{A_2}$ designates the number of occurrences when the opponent performed $a_{2t} = a_i$ when our supported agent implemented $a_1 = a_j$; and $\beta_{ij}^{A_2}$ are the prior parameters with $\beta_{ij}^{A_2} > 0$ for $i = \{1, \ldots, m\}$. In these cases, the required data will be stored in a two-dimensional matrix.

2.5.3 Preference Model

Each agent will aim at satisfying five objectives as in [21], slightly modified here to account for social interactions. We assume that the agents use a multi-attribute utility function, see [4], adopting an additive form. In qualitative terms, the objectives are ordered in hierarchical importance for the robot (1) to achieve a sufficiently high energy provision; (2) to ensure that it performs under safe conditions; (3) to be considered as a member of the society; (4) to be accepted as such; and, finally (5) to achieve complete functionality by having its software updated to the latest version. Utility weights were assessed initially with the constraint $w_i > w_{i+1}$ to take into account the objectives hierarchy and tested for the sensibility of behavior of the robot. The first objective u_1(*energy*), as well as u_{22} (*temperature*), u_{23} (*light*), u_{24} (*noise*), u_{315} (*being touched*), u_{32} (*detection*) and u_{52} (*being updated*), remain unchanged from [21] as none of them depend on the interaction with other participants. The other sub-objectives will be extended as follows:

$$u_{21}(\mathit{attack}) = \begin{cases} 1, & \text{if no attack from any user or agent is inferred at } t \\ & \text{or at } t-1, \\ 0.5, & \text{if after an attack at } t-1, \text{ there was no attack at } t, \\ 0, & \text{otherwise,} \end{cases}$$

where attack refers to actions $b_5 = \mathit{attack}$, $b_6 = \mathit{offend}$ and $a_{13} = \mathit{offend}$.

$$u_{311}(\mathit{not\ ignored}) = \begin{cases} 0, & \text{if the agent is ignored by any user at } t \\ 0.5, & \text{if it was ignored at } t-1, \text{ but was not at } t \\ 1, & \text{otherwise,} \end{cases}$$

$$u_{312}(\mathit{being\ spoken}) = \begin{cases} 1, & \text{if an agent performs action speak at } t \\ 0.5, & \text{if an agent performed action speak at } t-1 \\ 0, & \text{otherwise,} \end{cases}$$

where *performing action speak* refers to detecting the user, or another agent, initiating a dialogue (i.e. $b_{10} = \mathit{speak}$ or $a_6 = \mathit{speak}$).

$$u_{313}(\mathit{asked\ to\ play}) = \begin{cases} 1, & \text{if the robot is asked to play by the user} \\ & \text{or by another agent at } t, \\ 0.5, & \text{if the robot was asked to play at } t-1, \\ 0, & \text{otherwise,} \end{cases}$$

where *asked to play* refers to detecting a request to play from the user ($b_{13} = \mathit{play}$), including the game's title, or by another agent ($a_7 = \mathit{ask\ for\ playing}$).

$$u_{314}(\mathit{being\ ordered}) = \begin{cases} 1, & \text{if the robot receives an order from any user at } t, \\ 0.5, & \text{if the robot received an order at } t-1 \text{ but not at } t, \\ 0, & \text{otherwise,} \end{cases}$$

where *being ordered* consists of detecting an order among a catalogue of verbal actions ($b_{12} = \mathit{order}$).

$$u_{41}(\mathit{play}) = \begin{cases} 1, & \text{if the robot inferred a user or another agent} \\ & \text{playing around at } t, \\ 0.5, & \text{if the robot was playing with somebody at } t-1, \\ 0, & \text{otherwise,} \end{cases}$$

where *playing around* is referred to actions $b_{13} = play$ and $a_5 = play$, respectively.

$$u_{42}(\mathit{flatter}) = \begin{cases} 1, & \text{if the robot is flattered by a user} \\ & \text{or by another agent at } t, \\ 0.5, & \text{if the robot was flattered by a user} \\ & \text{or by another agent at } t-1, \\ 0, & \text{otherwise,} \end{cases}$$

being $b_3 = \mathit{flatter}$ and $a_{12} = \mathit{flatter}$, the incumbent actions.

$$u_{43}(\mathit{stroke}) = \begin{cases} 1, & \text{if the robot receives a stroke from a user at } t, \\ 0.5, & \text{if the robot received a stroke from a user at } t-1, \\ 0, & \text{otherwise,} \end{cases}$$

$$u_{44}(\mathit{apologize}) = \begin{cases} 1, & \text{if the robot receives an apology from a user} \\ & \text{or an agent after an attack,} \\ 0.5, & \text{if the robot received an apology from a user} \\ & \text{or an agent at } t-1 \text{ after an attack,} \\ 0, & \text{otherwise,} \end{cases}$$

being $b_4 = \mathit{apologize}$ and $a_{14} = \mathit{apologize}$ the incumbent actions.

For its fifth objective, the robot considers its social adaptation. To do so, it evaluates whether it is considered as socially useful by its peers, and whether it has been updated recently. We represent this through

$$\begin{aligned} u_5(\mathit{social\ adaptation}) = {} & w_{51} \times u_{51}(\mathit{socially\ useful}) \\ & + w_{52} \times u_{52}(\mathit{being\ updated}), \end{aligned}$$

with $\sum_{i=1}^{2} w_{5i} = 1$, and weights ordered in importance as follows: $w_{51} \gg w_{52} > 0$. The aim of u_{51} is two-folded: on one hand, we want to evaluate whether the agent is somehow recognized as a member of the society, inferring the reactiveness of its opponents to its actions; on the other, we want to measure how good the reaction of its opponents is. Our implementation of these ideas is

$$u_{51}(\mathit{socially\ useful}) = \frac{\sum_{k=1}^{q} p(M_j^k)}{q} \times \mathit{inter},$$

where $p(M_j^k)$ is the probability used to represent the j-th agent estimation of how reactive the human opponents B_k were, see (2.5); and *inter* is the impact of the reaction of opponent B_k at t, with

$$
inter = \begin{cases} 1, & \text{if } b_t \in \text{ affective actions,} \\ 0.5, & \text{if } b_t \notin \text{ affective actions and } \notin \text{ aggressive actions,} \\ 0, & \text{otherwise,} \end{cases}
$$

where the set of affective actions is $\{b_1 = \mathit{recharge}, b_2 = \mathit{stroke}, b_3 = \mathit{flatter}, b_4 = \mathit{apologize}\}$, and the aggressive actions set is $\{b_5 = \mathit{attack}, b_6 = \mathit{offend}, b_7 = \mathit{move}, b_8 = \mathit{blind}, b_9 = \mathit{shout}, b_{10} = \mathit{discharge}, b_{12} = \mathit{ignore}\}$.

As explained in [21], agents operating in this way may end up being too predictable. This may be a shortcoming in certain applications leading to repetitive interaction with the agents and, consequently, it losing interest in the users. We may reduce such effect by choosing the next action in a randomized way, with probabilities proportional to predictive expected utilities. However, there are certain rules that must be satisfied before the randomization. For example, the action *ask for help* will be performed only if the robot feels insecure, implemented through the component utility function of the second objective being below 0.5; action *salute* will be performed only if the robot detects a new user in the scene; action *ask for charging* will be performed only if the battery level is below 20 %; and so on.

For each of the scenarios considered we performed 20.000 iterations, which took about 10 min, corresponding to 166 h of actual interaction.

2.5.4 Competitive Scenario

Through this experiment, we want to show how a level-1 agent competes strictly against a level-0 agent, first, and, secondly, against another level-1 agent. This corresponds to $\lambda_1 = 0$ in our previous discussion. Recall that, in both cases, there is a user interacting with the incumbent agents. In the first case, only one agent makes its decision based on the ARA framework considering its opponents as non-strategic thinkers, whereas in the second one, both of them do. We expect that when all the decision agents within a society perform a level-1 ARA analysis, they perform as under a fictitious play model in which players form beliefs about how opponents play and maximize expected utility, see [2]. This leads them, after a sufficiently long performing period, to a Nash Equilibria. In the experiment, we simulate that each agent will consider, alternatively, the user and the opposite agent, as its adversary. After updating their forecasting models, they would choose the action to be implemented and update their internal clock.

We may appreciate the reaction of a level-0 agent and a level-1 agent (rows in both cases), respectively, facing a user (columns) while behaving competitively against each other in Figs. 2.3 (level-0 agent) and 2.4 (level-1 agent), where the size of the boxes is proportional to the relative frequency of each agent action in response to the user action. We may appreciate, based on the frequency of the corresponding actions, that the level-1 agent behaves somewhat more coherently performing more often actions like *apologize*, *asking for help* or *complain*, when the user has performed

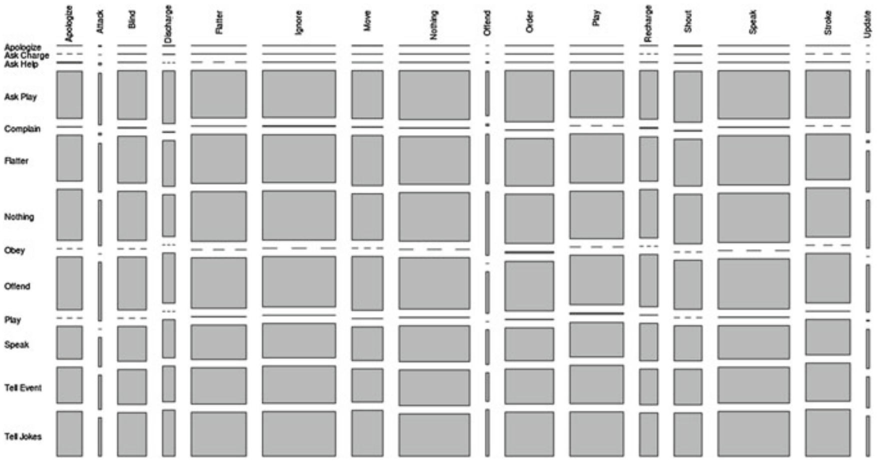


Fig. 2.3 Level-0 agent reacting to user actions. Box sizes proportional to the relative frequency of the agent action in response to the user action

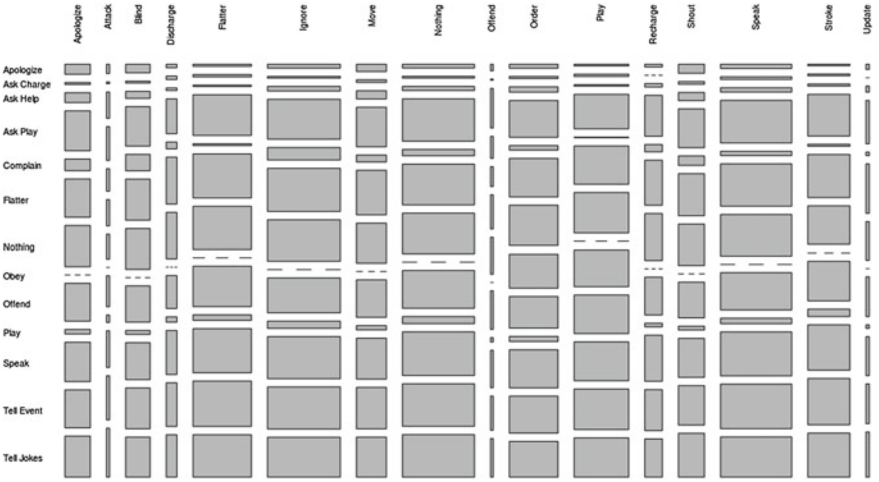


Fig. 2.4 Level-1 agent reacting to user actions. Box sizes proportional to the relative frequency of the agent action in response to the user action

an aggressive action like *attack*, *move* or *shout*, among others. On the other hand, we may appreciate that the behavior of the level-0 agent does not significantly change depending on the action of the user, corresponding to a random behavior, with the same distribution for each agent action given the action of the user.

Given that the user action and the environmental state are perceived simultaneously by all agents, any behavioral difference should be due to the strategic capacity of the corresponding agent. Both level-1 agents achieve similar utility levels (first quartile ≈ 0.426, median ≈ 0.472, third quartile ≈ 0.527) when competing against each other, as we expect them to be under a Nash Equilibria if they interact for a sufficient long time. The level-1 agent reaches approximately the same level of utilities facing a level-0 agent as before (first quartile ≈ 0.401, median ≈ 0.442, third quartile ≈ 0.494). However, the utilities obtained by the level-0 agent are lower (first quartile ≈ 0.343, median ≈ 0.395, third quartile ≈ 0.438). Recall that the utility function is scaled between 0 and 1.

Similarly, the level-1 agent who competed against a sophisticated agent computed slightly higher optimal utilities (first quartile ≈ 0.431, median ≈ 0.463, third quartile ≈ 0.507), than when facing a level-0 agent (first quartile ≈ 0.426, median ≈ 0.459, third quartile ≈ 0.506).

This experiment thus suggests that sophisticated agents obtain approximately the same utility levels independently of the adversary they are facing, but higher utilities than less sophisticated ones. Therefore, there is indeed an advantage in performing strategically.

2.5.5 Cooperative Scenario

Within the cooperative scenario, we assume that whenever the user performs an action within the user's interacting group of activities (*speak*, *order* and *play*), we consider it as a potentially cooperative situation with agents trying to satisfy the user. Both agents interact with the user and will establish communication through the CTTP, which collects all the information needed from the agents and compute the solution described in Sect. 2.4, which depends on the society's cooperativeness and competitiveness parameters. Each agent j has its own competitiveness parameter λ_{j2}. In this experiment, we shall assume that the competitiveness parameter will depend on the utility level obtained in the previous iteration through $\lambda_{j2} = 1 - u_j(a_{t-1}, b_{t-1}, e_{t-1})$, with $\lambda_{j1} = 1 - \lambda_{j2}$. In other words, if the agent satisfies its objectives, this will contribute positively to the society's cooperativeness parameter: the higher values of $u_j(a_{t-1}, b_{t-1}, e_{t-1})$, the closer it will be to 1, so that λ_{j2} will be smaller (and λ_{j1} bigger).

In this experiment we study which is the solution suggested by the CTTP and whether that solution improves, in utility and expected utility terms, the ARA level-1 solution, and how the implemented action is affected by the society's competitiveness values. For that purpose, we have computed the cooperative and the ARA level-1 solutions at each time-step, although the cooperative one is the only solution applied during the simulation. We have considered that the disagreement point will be where both agents implement the ARA level-1 solution, as in Sect. 2.5.4, whenever it is feasible, and, if not, the closest attainable expected utility vector. Figures within this Section correspond to agent A_1.

In Fig. 2.5, we may appreciate that the more competitive the society is, the lower expected utility each agent obtains from the alternatives, as the actions selected

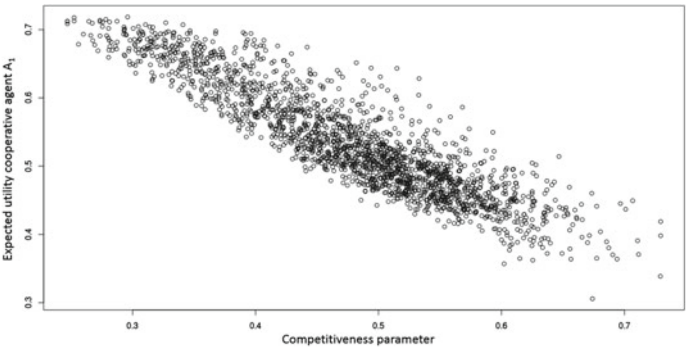
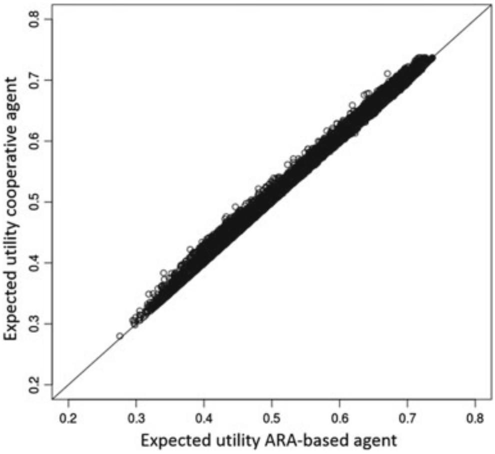


Fig. 2.5 Impact of society's competitiveness on agents' expected utility

Fig. 2.6 Expected utility attained by a cooperative (y-axis) and an ARA-based agent (x-axis)



get closer to the disagreement point, in consonance with now the competitiveness parameter is chosen.

Figure 2.6 suggests how a cooperative agent tends to achieve higher expected utilities than an ARA-based agent. Under the ARA framework the attained expected utilities are: first quartile ≈ 0.472, median ≈ 0.522, third quartile ≈ 0.6. Under a cooperative attitude they are: first quartile ≈ 0.478, median ≈ 0.53, third quartile ≈ 0.606. Based on this, we suggest that the cooperativeness parameter used within the solution described in (2.8) impacts (positively) on the agent decision making, increasing the utility values it expects to reach, outperforming those obtained by the ARA level-1 agent, without much additional computational load.

In terms of utility, we may observe that under the ARA framework the utilities obtained from the consequences are lower (first quartile ≈ 0.452, median ≈ 0.513, third quartile ≈ 0.589) than those obtained under a cooperative attitude, being the cooperative approach a better solution (first quartile ≈ 0.473, median ≈ 0.53, third quartile ≈ 0.6).

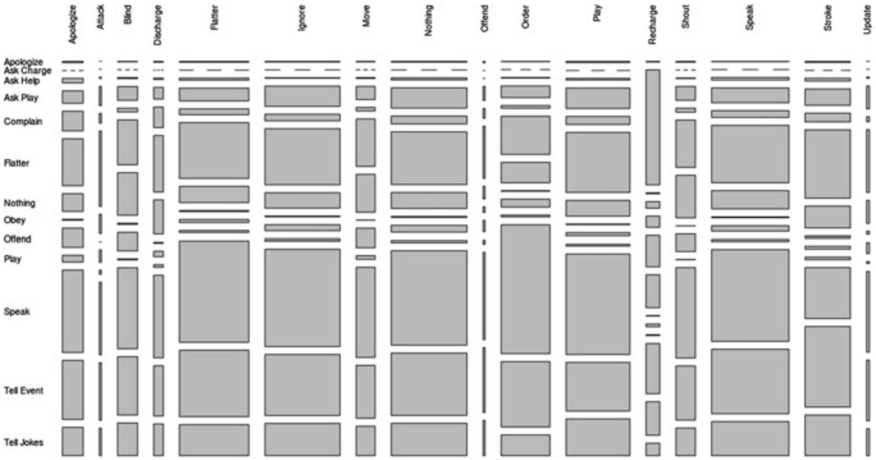


Fig. 2.7 Cooperative agent's actions depending on the user's action

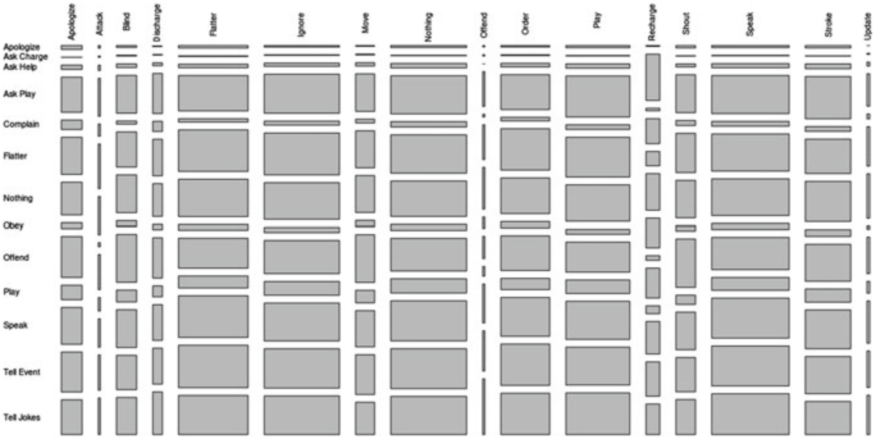


Fig. 2.8 ARA based agent's actions depending on the user's action

Finally, in Figs. 2.7 and 2.8 we may observe the different reaction of an agent (rows) within a cooperative situation and under the ARA framework, respectively, while interacting with the same user (columns). We appreciate that under cooperative situations, see Fig. 2.7, actions are not as uniformly distributed as under the ARA framework, which can be explained through the competitiveness parameter providing the cooperative agent with more adaptability to deal with interactive situations. Recall that in Figs. 2.7 and 2.8, the size of the boxes is proportional to the relative frequency of each agent action in response to the user action.

2.6 Discussion

We have described a social behavioral model for an autonomous agent, which imperfectly processes information from its sensors, facing several intelligent adversaries using multi-attribute decision analysis at its core, complemented by forecasting models of the adversaries. The preference model in [21] has been extended to include social adaptation. We have also explored the interaction among different agents, and users, under competitive and cooperative attitudes, depending on the social needs of our agent.

Regarding competitive attitudes, we have promoted the ARA solution over Nash equilibria concepts, as it avoids too strong common knowledge assumptions. Within cooperative scenarios a solution concept which aims at maximizing a distance from a disagreement point has been presented. The distance function used is parameterized with two parameters which respectively measure the degree of cooperativeness and competitiveness of our agent. Based on such parameters, the agent would move from a cooperative attitude towards a competitive one, or vice versa. Through a set of simulations performed with the AiSoy1 robot, we have demonstrated that the society's competitiveness parameter has indeed an impact on the actions implemented by the agents, and that, using the cooperative solution, the expected utility and the utility of the consequences that the agents receive tend to be higher than under the non-cooperative model.

Our ultimate interest for this type of models is the design of societies of robotic agents that interact among them and with one or more users. Based on our experiments, the proposed approach is amenable from the computational point of view and usable in real time applications. Note however, that we have limited memory to the two last periods and have planned just one period ahead. Longer memories and planning periods ahead, as well as continuous action spaces, would require much more powerful processing environments. To achieve such final goal, there are some open issues that we should still deal with, including those described next.

The implementation of the models presented in this paper corresponds to a first level, in the level-k thinking hierarchy, in that we only appeal to past behavior of the adversary, possibly as a response to our previous behavior. As future work, we aim at facing more sophisticated adversaries climbing up higher in the ARA hierarchy reaching at least a level-2, where we consider our opponents as level-1 strategic thinkers.

In order to optimize the models, we should further discuss which averaging procedure we should use to compute the society's attitude towards cooperation, and how those cooperative and competitive parameters would evolve depending on such procedures.

The field of cognitive processes has recently shown that emotions may have a direct impact on decision-making processes, see e.g. [3]. Advances in areas such as affective decision making, see [13], and neuro-economics, see [9], are based on this principle. Following this, a potential future work, concerning these models will be addressed towards providing a model for our autonomous agent that makes decisions influenced by emotional factors when interacting with humans and other agents, see e.g. [26]. With this, we aim at making 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. Supported by the AXA-ICMAT Chair in Adversarial Risk Analysis. We are grateful to discussion with Diego García, from AiSoy Robotics S.L., Jesus Ríos and David Banks.

References

1. AiSoyRobotics: AiSoy1 robot. http://www.aisoy.es/ (2008)
2. Brown, G.W.: Iterative solution of games by fictitious play. Act. Anal. Prod. Alloc. **13**, 374–376 (1951)
3. Busemeyer, J.R., Dimperio, E., Jessup, R.K.: Integrating emotional processes into decision-making models. In: Gray, W.D. (ed.) Integrated Models of Cognitive Systems, pp. 213–229. Oxford University Press, New York (2007)
4. Clemen, R.T., Reilly, T.: Making Hard Decisions with Decision Tools. Duxbury Press, Pacific Grove (2004)
5. Clyde, M., George, E.I.: Model uncertainty. Stat. Sci. **19**, 81–94 (2004)
6. Esteban, P.G.: Cooperation and Competition in Emotional Robot Societies. Ph.D. thesis, Rey Juan Carlos University, 2014
7. Esteban, P.G., Ríos Insua, D.: Supporting an autonomous social agent within a competitive environment. Cybern. Syst. Int. J. **45**(3), 241–253 (2014)
8. Gibbons, R.: A Primer in Game Theory. Harvester Wheatsheaf, Hertfordshire (1992)
9. Glimcher, P.W., Fehr, E., Camerer, C., Poldrack, R.A.: Neuroeconomics: Decision Making and the Brain. Elsevier, Amestradam (2008)
10. Harsanyi, J.C.: Subjective probability and the theory of games: comments on Kadane and Larkey's paper. Manag. Sci. **28**, 113–120 (1982)
11. Hoeting, J., Madigan, D., Raftery, A., Volinsky, C.: Bayesian model averaging: a tutorial. Stat. Sci **4**, 382–417 (1999)
12. Kadane, J.B., Larkey, P.D.: Subjective probability and the theory of games. Manag. Sci. **28**, 113–120 (1982)
13. Loewenstein, G., Lerner, J.S.: The role of affect in decision making. Handb. Affect. Sci. **3**, 619–642 (2003)
14. Menache, I., Ozdaglar, A.E.: Network Games: Theory, Models, and Dynamics. Morgan and Claypool Publishers, California (2011)
15. Myerson, R.: Game Theory: Analysis of Conflict. Cambridge Press of Harvard University Press, Massachusetts (1991)
16. Raiffa, H.: The Art and Science of Negotiation. Cambridge Press of Harvard University Press, Massachusetts (1982)
17. Raiffa, H., Richardson, J., Metcalfe, D.: Negotiation Analysis: The Science and Art of Collaborative Decision Making. Cambridge Press of Harvard University Press, Massachusetts (2007)
18. Ríos Insua, D., Kersten, G.E., Ríos, J., Grima, C.: Towards decision support for participatory democracy. Inf. Syst. e-Bus. Manag. **6**, 161–191 (2008)
19. Ríos Insua, D., Ríos, J., Banks, D.: Adversarial risk analysis. J. Am. Stat. Assoc. **104**, 841–854 (2009)
20. Ríos, J., Ríos Insua, D.: Adversarial risk analysis for counterterrorism modeling. Risk Anal. **32**, 894–915 (2012)

21. Rázuri, J.G., Esteban, P.G., Ríos Insua, D.: An adversarial risk analysis model for an autonomous imperfect decision agent. In: Guy, T.V. (ed.) Decision Making and Imperfection, pp. 163–187. Springer, New York (2012)
22. Stahl, D.O., Wilson, P.W.: On players models of other players: theory and experimental evidence. Games Econ. Behav. **10**, 218–254 (1995)
23. Thomson, W.: Nash's bargaining solution and utilitarian choice rules. Econom.: J. Econom. Soc. **49**(2), 535–538 (1981)
24. Thomson, W.: Cooperative models of bargaining. In: Aumann, R.J., Hart, S. (eds.) Handbook of Game Theory with Economic Applications, pp. 1237–1284. Elsevier, Amestradam (1994)
25. Thomson, W.: Bargaining and the theory of cooperative games: john nash and beyond, RCER working. Papers 554, University of Rochester - Center for Economic Research, (2009)
26. Villa, A.E., Missonnier, P., Lintas, A.: Neuroheuristics of decision making: from neuronal activity to EEG. In: Guy, T.V. (ed.) Decision Making with Imperfect Decision Makers, pp. 159–194. Springer, New York (2012)
27. Wooldridge, M.: An Introduction to Multiagent Systems. Wiley, New York (2008)
28. Yu, P.L.: A class of solutions for group decision problems. Manag. Sci. **19**, 936–946 (1973)

Springer

http://www.springer.com/978-3-319-15143-4

Decision Making: Uncertainty, Imperfection,
Deliberation and Scalability
Guy, T.V.; Karny, M.; Wolpert, D. (Eds.)
2015, XII, 184 p. 41 illus., 13 illus. in color., Hardcover
ISBN: 978-3-319-15143-4