Abridgement of HANA: a Human-Aware Negotiation Architecture ⋆

Angela Fabregues and Carles Sierra
{fabregues,sierra}@iiia.csic.es

IIIA - Institut d'Investigació en Intel·ligència Artificial
CSIC - Spanish Scientific Research Council,
UAB, 08193 Bellaterra, Catalonia, Spain

Abstract. HANA is an agent architecture suitable for multiple bilateral negotiations in realistic problems involving humans. The architecture deals with pre-negotiation and provides a new search and negotiation technique where search and negotiation go hand in hand: the former providing offers to propose, and the later providing commitments for pruning the search space, and information for fine-tuning the evaluation of offers. The architecture represents graded beliefs, dynamic desires and general intentions. It can be extended incorporating new behavioural models that can enrich the negotiation strategy with new information.

1 Introduction

The research field of negotiation has been studied from many different perspectives, among them: game theory [2, 3], psychology [4], business [5], neuroeconomics [6], or psychopharmacology [7] just to mention a few. The computational study of negotiations is denoted by automated negotiation. Most works on automated negotiation assume rational agents and static negotiation problems. However, humans are rationally bounded, and their negotiations are usually dynamic. It is often impossible to explore the complete negotiation space due to time limitations and the dynamics of the problem. By the time that an optimal solution is found, the solution is not optimal any more. This paper contributes to automated negotiation defining an agent architecture suitable to negotiate with humans.

We named our proposed architecture by the Human-Aware Negotiation Architecture (HANA). It allows multiple bilateral negotiations about actions, and deals with pre-negotiation looking for good enough sets of actions and offers. It is a modular architecture based on an ecological model of rationality. The mental state of the agent is represented as graded beliefs, dynamic desires and general intentions. We use a novel search and negotiation technique where search and negotiation go hand in hand: the former providing offers to propose, and the latter providing commitments for pruning the search space, and information for fine-tuning the evaluation of offers. Several negotiation strategies are provided that can be dynamically combined. The architecture is extensible, allowing the incorporation of new behavioural models.

The paper starts describing the background on multiple bilateral negotiation. Then, it introduces the problem (Section 3), and proceeds with a general description of the

⋆ This paper is an abridgement of [1], providing new background and discussion sections.

architecture (Section 4). The modules composing the architecture are described next. Those are: the interface, the world model (Section 5), the plan search (Section 6), and the negotiation (Section 7). The paper ends with a discussion (Section 8).

2 Background

The work on automated negotiation can be classified by the type of negotiation protocol being used. Common types are: multilateral protocols, typically organised as auctions among more than two agents; bilateral protocols, often with alternating proposals between two agents; and multiple bilateral protocols that are not as popular as the other two. Multiple bilateral negotiation protocols allow the negotiation of agreements among more than two agents at the same time. The main difference between multiple bilateral and multilateral protocols is that several agreements can be negotiated in multiple bilateral protocols while only one can be negotiated in multilateral protocols.

Multiple bilateral negotiations are common in business [8], however automated negotiation research lacks of this kind of negotiations. The main problem is that many bilateral negotiations take place in an “independent" manner, although the agreements reached affect a common agent. Thus, controlling those bilateral negotiations is crucial to avoid inconsistent commitments, and to reach the best possible outcome. Previous work on multiple bilateral negotiation combines two protocols. They define a negotiation process with two phases: a bilateral phase, and a multilateral phase. Examples are [9] and [10]. An alternative is to use sequential bilateral negotiations as an approximation to multiple bilateral negotiations [11]. This approach avoids the simultaneity of negotiations and simplifies the problem. It does not negotiate concurrently, however it takes into account a scenario with several agents, and does not consider being in an isolated bilateral negotiation. The outcome of negotiating with an agent affects the subsequent negotiations with that agent and others. There are works that simplify the problem assuming that there is only one agreement as an outcome of a negotiation with multiple opponents, [12–14].

Our interest in HANA is in domains where multiple potential agreements could be feasible. In [15], the authors share the same interest and envisage a negotiation support systems solution combining human judgement capabilities with autonomous agents. Similarly, other works on automated negotiation use an agent per negotiation counterpart and a control or management mechanism (maybe another agent) to control all those agents. For instance, in [14], an agent is used to coordinate the negotiating agents that conduct reasoning by using constraint-based techniques. They use two levels of strategies: the individual negotiation level, and the coordination level. After each negotiation cycle, the individual agents report back to the coordinator that evaluates their performance and issues new instructions accordingly. The idea of a MAS performing as a single negotiator is not new. [16] does already assume the existence of multiple agents. The authors do not only assume one agent per negotiating counterpart and a controller, they assume the existence of many other agents for several parts of the reasoning of the automated negotiator. Our approach skips this tendency of using several agents to build an automated negotiator, and use a single one with a single mental state and with the capability to perform concurrent tasks like, for instance, searching for good possible agreements and sending proposals. Our alternative to the use a MAS as an automated agent consists in executing those concurrent tasks using several threads sharing the same information and reasoning mechanisms. This alternative can be seen also in [17, 18]. Those works use a negotiation protocol completed by a normative system where, apart from proposing (*offer*), accepting (*accept*) and rejecting (*end*) agreements, agents can *confirm* and *decommit*. Confirm is used to confirm an acceptance. This is necessary in those works to avoid the acceptance of inconsistent agreements by the negotiating threads. The delay between accepting and confirming an agreement can be used by the negotiation counterpart to end the negotiation. However, when an acceptance is confirmed, any decommitment performed by the agents, meaning to break the agreement, implies a penalty. Our approach uses a far simpler protocol where acceptances do not need to be confirmed, and decommitment is not announced. The penalty of decommitment is not stipulated by the protocol, it is decided and performed by the other agents, for instance, not relying any more on the given agent.

Most negotiation models address the issues associated with the design of negotiation protocols and negotiation strategies. However, few of them deal with the preparation and planning for negotiation. According to [19]: “Peace requires a process of building constructive relationships in a civil society not just negotiating, signing, and ratifying a formal agreement." As relevant is the negotiation protocol or strategy, as is the structure of relevant information, the analysis of possible negotiating options, and the plan to follow. This task is denoted by *pre-negotiation* and is studied in other works like [20, 17, 21–23]. We do pre-negotiation in this paper with the definition of the HANA architecture that generates negotiating options from joint possible plans.

As described in [24], pre-negotiation addresses the operational and strategic process of preparing and planning for negotiation. This concerns the structuring of personal information, the analysis of the opponents, and the definition of the protocol and selection of the initial strategy. HANA assumes the use of a multiple bilateral protocol. Even though, the codification of the protocol as a normative system incorporated in the world model makes possible the use of other protocols as the world model is extensible. The main restriction in our architecture is on the negotiating objects that are assumed to be sets of actions refusing the possibility to argue about the suitability of a particular negotiation protocol. However, the rest of pre-negotiation activities are present in our architecture.

3 Resource negotiation problem

A Resource Negotiation Problem (RNP) is a particular resource allocation problem with negotiating agents. An RNP is a MAS problem with several agents $\alpha \in \mathcal{A}$ controlling the resources $r \in R$ of a common environment. Agents individually act applying particular operators $op \in Op$ to their resources, and negotiate to agree on particular combinations of actions to be performed at the same time. An action is the application of an operator to a resource by an agent $a = \langle \alpha, op, r \rangle \in A$. A plan is a set of action $p = \{a_1, a_2, ...\} \in P$. A plan is complete $\bar{p} \in \bar{P}$ if it contains an action for every resource in the environment. It is complete for a given agent $\bar{p}_\alpha \in \bar{P}_\alpha$ if it contains an action for every resource in the environment controlled by that agent. An environment $\omega \in W$ is a deterministic state transition system where the state is represented by the partition of resources among agents $W = 2^{\mathcal{A}\times R}$, and the transition function determines the next state given a complete plan: $\omega' = \mathbf{T}(\omega, \bar{p})$. Thus, the evolution of the environment is given by the actions that agents execute.

The negotiation in RNPs is bilateral. The negotiation objects are denoted by negotiation options $\delta \in \mathcal{O} \subseteq P$ and are plans with actions involving only two agents. A negotiation utterance $\mu = \langle \theta, \alpha, \beta, \delta \rangle \in M$ represents a message containing a negotiation option, a sender, a receiver, and an illocutionary particle θ that can be: to propose, to accept or to reject. All agents controlling resources with actions included in the negotiation option must participate in the communication: either as the sender or the receiver. Therefore, it is not possible to negotiate about actions to be performed by a third party. A negotiation dialogues is a sequence of utterances $\Psi = \langle \mu_0, \mu_1, ... \rangle$. The negotiation protocol establishes that all proposals must be replied to. It enforces this forbidding them to send a proposal to an agent that is waiting for our reply. Nevertheless, the agent is allowed to send a proposal to someone else. Therefore, the protocol permits concurrent negotiations among themselves. When a negotiation option is accepted, the involved agents commit to perform the actions included in the negotiation option.

4 Agent architecture

HANA is a software architecture to build agents capable of participating in RNPs. We introduce here its main modules and then we give details for each of them in subsequent sections. We refer to the agents designed according to HANA as *HANA agents*.[1] The architecture is graphically represented in Figure 1.

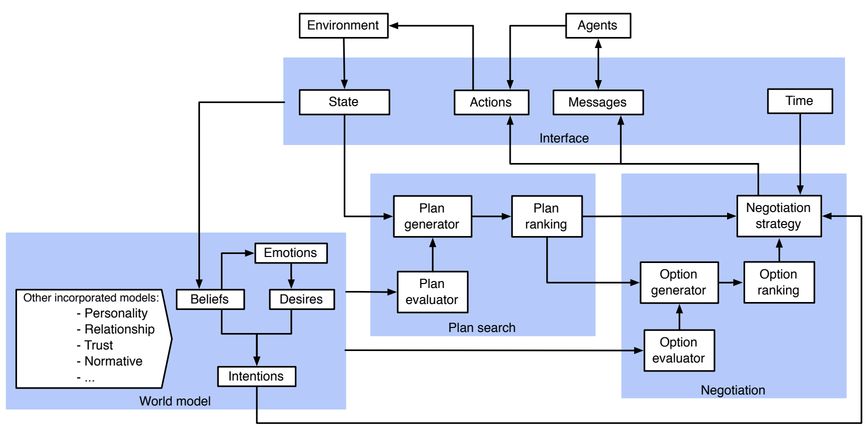

Fig. 1: Graphical representation of HANA. Arrows represent data flows. Coloured boxes represent the modules that form part of the agent, and white boxes are components of those modules.

[1] HANA is an acronym for Human-Aware Negotiation Architecture. It stresses the aim of negotiating with humans in realistic scenarios that motivated the design of this agent architecture.

In an RNP, agents can execute actions and exchange private messages. Thus, the first component of HANA is an *interface module* that situates the agents in their environment, that is: it allows the observation of the environment state, observe and execute actions, and exchange messages with other agents. In other words, this module contains the sensors and actuators of the agent. Which actions to execute and which messages to send are decided by the *negotiation strategy*.

The design philosophy behind HANA is to provide some means to negotiate as humans do, as the negotiation counterparts could be humans. In particular, there are two capabilities that we think realistic agents should show: dealing with emotions, and dealing with uncertainty [25]. The architecture incorporates emotions as this is an important part of the non-constructivist rationality approach, we need to understand emotional reactions of the other negotiators. Although the environment is fully observable, the actions to be executed by the other agents can only be guessed analysing the other agents' previous behaviour. To cope with this uncertainty, we decided to represent the world as a graded BDI model, that is with graded *beliefs*, *desires* and *intentions* following the g-BDI model [26].

The space of plans and negotiation options that an agent can execute and propose, respectively, is potentially huge. We assume that the space is large enough and the negotiation time short enough to preclude obtaining the optimal solution. That means that any architecture for this type of negotiation needs to give the means to look for good enough solutions. Moreover, the longer it takes to decide what to propose, the less probable is the proposal to be accepted. As time goes by, the agents reach agreements that increase the amount of commitments and reduce the set of options compatible with those commitments. Increasing acquired commitments increases, in turn, the probability that our desired plans will not be compatible any longer. Consequently, the architecture must allow agents to start negotiating from the very beginning of a negotiation round. Dealing with huge solution spaces is not inconvenient for human agents, e.g. in playing Chess or Go. Humans work with good enough solutions in their everyday lives. Time constraints, boredom, or tiredness make humans accept good enough solutions. To start negotiating from the very beginning, HANA proposes to perform a search and negotiation technique that assumes the plan search to go hand in hand with the negotiation. The *plan search module* executes an anytime algorithm that provides a periodically updated ranking of good enough plans. The ranking takes into account the commitments obtained by the *negotiation module*. And the negotiation module proposes options generated from the previously found good enough plans that contain actions to be executed by other agents. In this way, HANA agents can start the negotiation from the very beginning proposing options that, once negotiated, will provide new information —because the option will be accepted or rejected— to focus the search on the generation of new and better evaluated plans. As can be seen in Figure 1, the plan and option evaluators depend not only on the commitments but on the whole world module. Thus, those evaluation functions are also updated taking into account the intentions generated from new observations. The intentions trigger the decisions of the agent.

The execution of HANA consists of several concurrent processes for: the *interface* (to receive messages and observe the results of actions and the environment state), the *world model* (to update the world model given the perceived changes), the *plan search* (to continuously update the ranking of plans), and the *negotiation* (to generate options from plans and determine what to do next).

5 World model

During the last decade, many successful representation models have been based on BDI representations. BDI is based on the theory of human practical reasoning [27] and has well-founded logical semantics [28]. The work of Casali et al. [26] on what they denoted by *g-BDI*, gives a powerful generic representation for degrees of belief, degrees of desire (preferences) and degrees of intention (active goals). We adapt this work to our problem and incorporate the beliefs, desires and intentions as the main components of the world model.

The most important aspect of the evolution of the world is what an agent expects to happen in the environment due to the decisions of other agents. This is so because the natural evolution of environments is subject to shared knowledge on physical laws, and thus known by every agent. Therefore, the evolution of the world can be due to either actions, A, or utterances, M, of other agents. We denote those events (actions and utterances) by $\Phi = M \cup A$. The agent has at all times a belief degree assigned to every element of Φ meaning how certain is the agent that that event will happen. We decided to model these degrees as probabilities because the data for them comes from the previous interactions with the other agents and thus those data can be statistically processed. Axiomatics on how to represent probabilities are provided in [26].

Definition 1. *Given $\Phi = M \cup A$, a* belief *is a tuple $\langle \alpha, \varphi, \vartheta \rangle \in \mathcal{A} \times \Phi \times [0, 1]$ where ϑ is α's belief degree on φ happening. We denote by $\mathcal{B}$ the set of all possible beliefs and by $\mathcal{B}_\alpha \subseteq \mathcal{B}$ the set of possible beliefs of α.*[2]

For a given environment state ω, the belief degree of α on an action a happening is $B_\alpha(a)$. Recall that plans are considered sets of actions that are to be executed at the same time. The belief on the execution of a feasible plan $p = \{a_1, a_2, \ldots, a_n\} \in P^\omega$ is thus naturally modelled according to HANA as the belief on the conjunction of the execution of each action that is then modelled as the product. $B_\alpha(p) = B_\alpha(a_1 \wedge a_2 \wedge \cdots \wedge a_n) = \Pi_{a_i \in p} B_\alpha(a_i)$

When new observations of the environment are made, HANA agents update their beliefs. From the many possible *belief review functions* available in the literature [29–31] HANA uses a recency based belief revision defined as follows.

Definition 2. *Given an agent $\alpha \in \mathcal{A}$, a* belief review function*, denoted by $\sigma : 2^{\mathcal{B}_\alpha} \times 2^{\mathcal{B}_\alpha} \rightarrow 2^{\mathcal{B}_\alpha}$, is any function satisfying: (i) $\sigma(\mathcal{B}', \mathcal{B}'') = \mathcal{B}'''$, (ii) $\mathcal{B}'$ is the original belief set, (iii) $\mathcal{B}''$ is a new belief set, (iv) $\mathcal{B}^*$ is the resulting belief set before normalisation,*[3] *(v) $\mathcal{B}'''$ is the resulting normalised belief set, (vi) if $\langle \varphi, \vartheta \rangle \in \mathcal{B}'$, $\langle \varphi, \vartheta' \rangle \in \mathcal{B}''$ and $\vartheta \neq \vartheta'$ then $\langle \varphi, \vartheta' \rangle \in \mathcal{B}^*$, (vii) if $\langle \varphi, \vartheta \rangle \in \mathcal{B}'$ and $\langle \varphi, \_ \rangle \notin \mathcal{B}''$ then $\langle \varphi, \vartheta \rangle \in \mathcal{B}^*$, and (viii) nothing else belongs to $\mathcal{B}^*$.*

[2] We will note $\langle \alpha, \varphi, \vartheta \rangle \in \mathcal{B}$ as $B_\alpha(\varphi) = \vartheta$ when useful.

[3] To normalise, we follow the work done in [32], and compute the minimum relative entropy probability distributions with respect to the distributions in $\mathcal{B}'$ that consider the new beliefs in $\mathcal{B}''$ as constraints to satisfy, and that satisfy the protocol.

Desires, intentions and emotions are also updated via a similar simple update method that we omit here. Arrows in Figure 1 show the influence between the different motives: changes in the environment provoke updates in the belief set, that generates emotional updates, that update the desires. The new set of beliefs and desires determines new intentions.

Although only a few components in Figure 1 are interconnected to build up the world model of HANA agents (beliefs, desires, intentions and emotions), other models might be incorporated. The world model is based on multicontext systems [26] that are modular structures that allow for an easy interconnection of different (logical) models, using transition functions between them, to build even more complex agents. For instance, a trust model may impact on intentions, as the intention degree to satisfy a desire via a plan with an untrustworthy agent should be low. Also, a social model might impact on intentions, as we might want to have a higher intention degree on plans involving an agent with whom we would like to increase the level of intimacy [33].

As defined in Section 3, the agents must fulfil a negotiation protocol in order to be able to negotiate with other agents. The rules or constraints that the protocol provides can be incorporated in the agent as internal norms to follow. This is done, according to HANA with a high degree of desire on fulfilling the negotiation protocol, and some transition functions between this desire, and several beliefs that modify the degree of what we call *basic intentions*: *reply* δ (when the agent believes that it has received the proposal δ), *propose* (when there is time left in the negotiation round) and *executeActions* (when we are approaching the end of the negotiation round).

6 Plan search

The interplay between search and negotiation is the most important contribution of HANA. In most multi-issue negotiation problems the space of potential solutions is determined by the admissible values of issues. That is: potential solutions are elements of the cartesian product of the admissible values for each issue [34]. Differently, in RNPs the space of potential solutions is defined as the combination of feasible actions that are compatible. Only certain subsets of the space of actions constitute feasible solutions, and finding which ones are feasible is not straightforward. Good and bad solutions are not placed together nicely as in continuously valued attributes (e.g. if a certain low price is good, nearby low prices will be similarly good). Sometimes a small change in a plan makes it go from excellent to catastrophic. Moreover, the space of potential solutions in real domains is frequently huge. What HANA brings in to address this type of negotiation problem is a search and negotiation technique that enables the negotiation to start as soon as possible over reasonably good solutions.

The outcome of the search process is a continuously refreshed ranking of candidate plans. The *plan ranking* is made by the *plan generator* thanks to a utility function that represents the preferences of the agent and that is implemented within a component of the architecture called *plan evaluator*, see Figure 1. Preferences are determined by the world model, and thus, they for instance take into consideration personality traits and relationships between agents. They do not evaluate only their individual position.

Every agent α in a state transition system Ω must decide what actions to perform, that is, what complete plan $\bar{p}_\alpha$ to perform. Remember that given a state ω, next state ω' is computed by a state transition function $\mathbf{T} : W \times P \to W$. $\mathbf{T}(\omega, \bar{p})$ is defined for complete plans, $\bar{p} \in \bar{P}^\omega$, that are those that can be obtained from the union of complete plans for every agent controlling resources in the current state: $\bar{p} = \bigcup_{\beta \in \mathcal{A}} \bar{p}_\beta$. To decide what plan to perform, α must know what is the utility that every plan would provide. If α knew the plans of the other agents, $Q = \bigcup_{\beta \in \mathcal{A} \setminus \{\alpha\}} \bar{p}_\beta$, it could compute the utility of α performing the complete plan $\bar{p}_\alpha$ using its utility function, $\mathcal{U}_\alpha : W \to [0, 1]$, as: $\mathcal{U}_\alpha(\omega, \bar{p}_\alpha) = \mathcal{U}_\alpha(\mathbf{T}(\omega, Q \cup \bar{p}_\alpha))$.

However, whilst agents may have a clear idea of their preferences, and hence can build up a utility function, it is usually impossible to know what other agents' plans will be. Therefore, instead of using the deterministic transition function, $\mathbf{T} : W \times P \to W$, α must use a probabilistic state transition function.

Definition 3. *Given a transition system $\Omega = \langle \mathcal{A}, R, Op, W, P, \mathbf{T}, \omega_0, W_f \rangle$, a* probabilistic transition function*, denoted by $\mathbb{T}(\omega'|\omega, p) \in \mathbb{P}(W)$ is any conditional probability distribution over W given $\omega \in W$ and $p \in P$, such that for every complete plan $\bar{p} \in \bar{P}^\omega$ then $\mathbb{T}(\mathbf{T}(\omega, \bar{p})|\omega, \bar{p}) = 1$ and $\mathbb{T}(\omega'|\omega, \bar{p}) = 0$ for all $\omega' \neq \mathbf{T}(\omega, \bar{p})$.*

In RNPs, the state transition function $\mathbf{T} : W \times P \to W$ is fixed and common to all the agents. Instead, a probabilistic transition function has to be particular for each agent as it necessarily depends on the interpretation of other agents' past behaviour. Therefore, we will denote by $\mathbb{T}_\alpha(\omega'|\omega, p)$ the probabilistic transition function of agent α. Then, the utility of a plan, complete or not, for an agent can be estimated as $E[\mathcal{U}_\alpha(\omega, p)] = \sum_{\omega_i \in W} \mathbb{T}_\alpha(\omega_i|\omega, p) \times \mathcal{U}_\alpha(\omega_i)$ The complexity here relies on the evaluation of $\mathbb{T}_\alpha(\omega'|\omega, p)$.

We can identify the problem of learning the probabilistic state transition function for all complete plans for a given agent $\alpha \in \mathcal{A}$ as a Markov Decision Process (MDP), [35]. Learning $\mathbb{T}$ requires a wealth of data that is usually not available, and an initially random behaviour that may produce very negative outcomes in RNPs. Moreover, there is a required feature for any MDP problem that is not verified in our case: the Markov property. The transition function could depend on the past as other agents could learn from previous states and modify their decision function. Contrarily, we propose an alternative that is to infer the probability state transition function from beliefs on the execution of plans as defined in Section 5. From belief degrees on particular actions happening, we can compute the belief degree of complete plans. Also, beliefs easily integrate other sources of information that are missing in an MDP, such as emotions or previous commitments. That is, the belief degree on an action happening may be determined, for instance, by knowing that the other agent is of a revenge type or that the other agent reached an agreement with another agent whom we trust and told us so. Moreover, as new sources of information can be easily incorporated into the world model this makes the architecture highly flexible and modular. For all these reasons we define the expected utility not for a plan in particular, but for a set of beliefs hold in a particular state as follows:

Definition 4. *We define the* expected utility *for $\alpha \in \mathcal{A}$ holding the set of beliefs $\mathcal{B}' \subseteq \mathcal{B}_\alpha$ in state $\omega \in W$ as:*

$$E[\mathcal{U}_\alpha(\omega, \mathcal{B}')] = \sum_{\omega_i \in W} \left(\frac{\sum_{\substack{\bar{p} \in \bar{P}^\omega \\ \mathbf{T}(\omega,\bar{p})=\omega_i}} B'(\bar{p})}{\sum_{\bar{p} \in \bar{P}^\omega} B'(\bar{p})} \times \mathcal{U}_\alpha(\omega_i) \right) \tag{1}$$

The previous definition does not require that α has made up his decision of what actions to perform. That is, the equation can be used at the beginning of the negotiation process —when α is still uncertain on what to do, and also when all the bilateral negotiation processes have been finished and α knows what to do. The expected utility of a plan p is computed at any time assuming that the plan will be executed: $E[\mathcal{U}_\alpha(\omega, \sigma(\mathcal{B}', \{\langle \alpha, a, 1\rangle | a \in p\}))]$[4]. Actually, the richer the world model the more accurate the utility functions can be. We measure the level of information in a belief set as the average of Shannon's entropy among the probability distributions of actions to be operated on resources. Notice that, the higher the uncertainty the higher the entropy and thus, the less information.

Definition 5. *The uncertainty on a set of beliefs $\mathcal{B}' \subseteq \mathcal{B}_\alpha$ given the set of predicates Φ that partition it, is measured as:*

$$\mathcal{H}(\mathcal{B}') = -\frac{1}{|\Phi|} \sum_{\phi \in \Phi} \left(\frac{1}{\ln |\phi|} \sum_{\langle \alpha, \varphi_i, \vartheta_i \rangle \in \phi} \vartheta_i \ln \vartheta_i \right) \tag{2}$$

The uncertainty on actions to be executed is usually high at the beginning of a negotiation round, as an agent has not enough information about each agent decisions, to 0 when the complete plan is actually performed and observed by all agents. Negotiation is the means to reduce the uncertainty on the belief model. By reaching agreements on negotiation options, agents commit themselves to the execution of their actions in the negotiating option, and thus they reduce the uncertainty by making equal to zero the probability of executing incompatible actions on the same resource.

The task of the plan search module is to find good enough plans to be executed by the agent, but also to provide good enough plans to negotiate with other agents. Plans to be executed are complete plans for the agent α, that is, plans containing actions involving all the resources controlled by the agent in the current environment state. The plans to negotiate with other agents are extensions of those complete plans containing actions to be performed by other agents. These plans are denoted by *joint plans*, $\hat{p} \in \hat{P}_\alpha$.

The bilateral nature of the interactions force options to contain actions to be done by, at least, two agents. In HANA, joint plans involving more than two agents can be negotiated (either concurrently or sequentially) proposing several options generated from the joint plan. As introduced before, plans can be evaluated by their expected utility. Notice that this measure assumes that the plan will be executed. When joint plans are evaluated, we can also assume that the HANA agent will execute its part of the plan. Even though, we must be cautious about the actual execution of the actions in

[4] Plans are executed at the end of the negotiation round when the certainty on the actions to be executed are commonly high.

the plan assigned to other agents. We define the *confidence* of a plan in order to measure the degree of belief on the execution of a joint plan assuming that the HANA agent will perform its part of the plan.

Definition 6. *Given a set of beliefs $\mathcal{B}' \subset \mathcal{B}_\alpha$ hold by agent $\alpha \in \mathcal{A}$, and a feasible plan $p \in P^\omega$, α's* confidence *on p is: $\mathcal{C}_\alpha(\mathcal{B}', p) = B(\{\langle \beta, op, r\rangle | \langle \beta, op, r\rangle \in p \text{ and } \beta \neq \alpha\})$*

Note that for all $\bar{p}_\alpha \in \bar{P}^\omega_\alpha, \mathcal{C}_\alpha(\mathcal{B}', \bar{p}_\alpha) = 1$ because $B(\emptyset) = B(true) = 1$.

The output of the plan search is a *plan ranking*. The confidence measure can be used by HANA agents to rank joint plans. HANA agents can rank the joint plans by their utility (how good are they for the agent) filtering out those joint plans that do not reach a minimum level of confidence (the agent does not think that other agents will perform their actions in the plan). This minimum level of confidence may increase as time goes by and the negotiation round deadline approximates in order to focus on joint plans where confidence is high.

To generate plans we need a search algorithm that is: (i) capable of search in a huge space of solutions —as required by most real scenarios, (ii) anytime —as required by the time bounds, (iii) capable of generate several solutions instead of just one —we are looking for several plans, and (iv) guided by a dynamic heuristic —as the set of beliefs evolves with the agent interaction. We decided to implement HANA's plan generator with an evolutionary algorithm that constantly optimises the set of plans in the ranking. Concretely, we use a genetic algorithm (GA) as these algorithms allow the efficient exploration of large spaces of solutions and produce successive populations of solutions that are increasingly better adapted to the environment even when it is changing along the search process. For us, each single solution, a chromosome in the GA, represents a complete or joint plan for the agent. The idea is to generate the plan ranking from the current population of solutions taking all or a subset of the best ones, preferably the latter. This population is updated generation after generation by the crossover, mutation and selection operators. It is important to guarantee the feasibility of the generated plans when applying crossover and mutation. The evaluation of a chromosome is done by the fitness function that computes the expected utility of the plan represented by the chromosome. *Fitness proportionate selection* is used to give more chances to good plans to take part in crossovers. HANA's genetic search allows the setting of the genes mutation probability. It is used to focus the search on the joint plans looking for other agent actions that can nicely extend the best complete plans. In fact, to represent plans of diverse size, we use chromosomes with a size equal to the size of complete plans, and let some genes have a void value meaning that the resource corresponding to that gene has no assigned action. The probability of void values per gene can also be adjusted. The initial population does not need to be randomised. We set the initial population and stop the search at any time saving the current population. In this way, we can resume the computation later on. It is possible to use elitism to keep the best plans alive generation after generation. Elitism in the plan generation provides a minimum of stability needed to avoid an erratic performance of the agent during negotiation. The idea is to keep the plan search running all the time, however it can be stopped and restarted again.

7 Negotiation

The negotiation module uses the ranking of plans and the world model to decide how to negotiate, and what actions to perform. That is, what messages to send to other agents, and what actions to execute over the environment. The world model and the plan search have independent processes that make the world model data and the plan ranking evolve along time. The negotiation module is controlled by another process that takes a snapshot of the previous modules' data structure. The snapshot is a *negotiation state*.

Definition 7. *A* negotiation state *is a tuple $s = \langle \alpha, \omega, t, \mathcal{B}_\alpha^t, \mathcal{D}_\alpha^t, \mathcal{I}_\alpha^t, P_\alpha^t \rangle$, where: $\alpha \in \mathcal{A}$ is an agent, $\omega \in W$ is an environment state, t is a time instant, $\mathcal{B}_\alpha^t$, $\mathcal{D}_\alpha^t$, and $\mathcal{I}_\alpha^t$ are the beliefs, desires and intentions of α at time t, and $P_\alpha^t : P^\omega \mapsto [0, 1]$ is a plan ranking.*

We denote the set of all possible negotiation states by S. Taking a snapshot, the negotiation process can use the data from the world model and the plan ranking and perform a negotiation step while the plan search is looking for even better plans[5]. The workflow of the negotiation process is as follows: (i) takes a snapshot of the negotiation state, (ii) generates a ranking of negotiation options, (iii) executes the agent's intentions included in the world model, and (iv) goes to (i) to continue with a new negotiation state.

The negotiation process uses the option generator to build a ranking of negotiating options. Options are generated from the plan ranking P_α^t as combinations of actions in joint plans $\hat{p} \in \hat{P}_\alpha^\omega$ that are included in the plan ranking $P_\alpha^t(\hat{p}) \neq \perp$. Options are evaluated by the option evaluator that computes the next expected negotiation state assuming the acceptance of a given option $\delta \in \mathcal{O}_\alpha^\omega$. The simplest way to generate the ranking of options, $\mathcal{O}_\alpha^t : \mathcal{O}_\alpha^\omega \mapsto [0, 1]$ is as $\mathcal{O}_\alpha^t(\delta) = f(\mathit{next}(s, \delta))$, where: $s \in S$ is the current negotiation state, $f : S \mapsto [0, 1]$ is a negotiation state evaluation function, $\mathit{next} : S \times \mathcal{O}_\alpha^\omega \mapsto S$ computes the next expected negotiation state, and $\exists \hat{p} \in \hat{P}_\alpha^\omega$ such that $\delta \subseteq \hat{p}$ and $P_\alpha^t(\hat{p}) \neq \perp$. An alternative is to apply a filter and generate the option ranking using only the joint plans with value over a threshold $\nu > 0$. That is, using every plan $\hat{p} \in \hat{P}_\alpha^\omega$ such that $P_\alpha^t(\hat{p}) > \nu$.

HANA provides several evaluation functions for negotiation states. Other functions can be used. In general, the richer the world model the more sophisticated the evaluation functions can be. The following functions assume the basic world model:

– *Quality of information*. The higher the quality of the information that we can reach in a negotiation state the better. A well informed state contains joint plans that can reduce the uncertainty about the other agents' actions. The higher the uncertainty reduction the better. A natural way to evaluate the quality of information is to define it as 1 minus the average uncertainty of Definition 5.

$$f_H(s) = \max_{p \in P_\alpha^t} (1 - \mathcal{H}(\sigma(\mathcal{B}_\alpha^t, \{\langle \alpha, a, 1 \rangle | a \in p\}))) \tag{3}$$

[5] Note that the changes in the world model and the plan ranking that are done after taking the snapshot are considered in the next iteration of the negotiation process.

– *Independence*. The more independent an agent is the better. If an agent can reach a high utility by its own means the better the negotiation state is. This measure depends on the complete plans for the agent that have been found so far, $\bar{P}^t_\alpha = \{p | p \in \bar{P}_\alpha$ and $P^t_\alpha(p) \neq \bot\}$. A state is as good as the best state the agent can reach by its own means, this is the maximum expected utility to be obtained by assuming we choose one of the complete plans for agent α:

$$f_{\mathcal{U}C}(s) = \max_{\bar{p} \in \bar{P}^t_\alpha} E[\mathcal{U}_\alpha(\omega, \sigma(\mathcal{B}^t_\alpha, \{\langle \alpha, a, 1 \rangle | a \in \bar{p}\}))] \tag{4}$$

– *Opportunity*. The more utility to be obtained with joint plans the better. Finding joint plans that give high utility is actually the reason of the whole negotiation process. Any state that has joint plans with high expected utility is a good state. This measure is similar to the previous one but using joint plans $\hat{P}^t_\alpha = \{p | p \in \hat{P}_\alpha$ and $P^t_\alpha(p) \neq \bot\}$:

$$f_{\mathcal{U}J}(s) = \max_{\hat{p} \in \hat{P}^t_\alpha} E[\mathcal{U}_\alpha(\omega, \sigma(\mathcal{B}^t_\alpha, \{\langle \alpha, a, 1 \rangle | a \in \hat{p}\}))] \tag{5}$$

– *Confidence*. The more confidence in the available plans the better. Having a high confidence in the plans found during the search the less uncertainty on what will happen.

$$f_{\mathcal{C}}(s) = \max_{p \in P^t_\alpha} \{\mathcal{C}_\alpha(\mathcal{B}^t_\alpha, p) | p \in P^\omega, P^t_\alpha(p) \neq \bot\} \tag{6}$$

Each of these different measures allows the evaluation of the negotiation states and thus rank available options. When to use each measure is what determines an agent's negotiation strategy. The other key element of the negotiation strategy is the *aspiration level*, i.e. the minimum evaluation value that the agent has for options to be acceptable. The options above the aspiration level should be accepted. Otherwise, rejected. At the beginning of a negotiation round, agents would request a high aspiration value. As time goes by and the deadline approaches, agents become less demanding, and they decrease their expectations in order to reach some agreements that improve, even in a low amount, their negotiation state. HANA allows for the definition of the way the aspiration level decreases as the next definition shows.

Definition 8. *Given a negotiation state s, a deadline t_{max}, and current time t, the* aspiration level*, denoted $\mathcal{A}(s,t)$, is defined as:*

$$\mathcal{A}(s,t) = g_{min}(s,t) + \left(\frac{t_{max} - t}{t_{max}}\right)^\tau \cdot (1 - g_{min}(s,t))$$

where $\tau \in [0,1]$ is the aspiration decay rate and $g_{min}(s,t)$ is the minimum value that can be guaranteed.

The negotiation strategy is then determined by fixing values for $g_{min}(s,t)$. HANA allows to define these functions as linear combinations of the measures defined before. That is, $g_{min}(s,t) = w_1(t) \cdot g_1(s) + w_2(t) \cdot g_2(s) + ... + w_n(t) \cdot g_n(s)$ where every $g_i(s) \in [0,1]$ is a measure over the state s and $\sum_{i<n} w_i(t) = 1$. Next we discuss a few negotiation strategies:

– *Conservative*. An agent can guarantee a minimum utilitarian value with its own actions that corresponds to $g_{min}(s) = f_{UC}(s)$. This strategy is convenient at the end of a negotiation round as it concedes maximally towards the guaranteed minimum.
– *Informative*. A convenient strategy at the beginning of a negotiation round is to increase the agent's information quality. This facilitates the exploration of the space of options and reduce uncertainty in the negotiation. The more information an agent has the more probable its future proposals will be accepted. This can be achieved by $g_{min}(s) = f_H(s)$.
– *Dynamic*. A combination of the previous two strategies starting with informative and ending with conservative: $g_{min}(s,t) = w_1(t) \cdot f_H(s) + w_2(t) \cdot f_{UC}(s)$ where $w_1(t) = \frac{t_{max}-t}{t_{max}}$ and $w_2(t) = 1 - w_1(t)$.

8 Discussion

In this paper, we face the challenge of automated negotiation with humans focusing our attention in several aspects that were not studied in depth in the past, like pre-negotiation and multiple bilateral negotiation. We describe a negotiation problem, RNP, with several potential applications that may benefit humans in their everyday life. The use of a negotiation protocol allowing multiple bilateral negotiations is crucial to that end, as they are common in business [8]. Also the fact that negotiations in this problem are about actions, and repeat along time facilitating the establishment and evolution of relationships among agents. Deadlines are assumed for negotiations as they take place before actions are performed.

An architecture for agents to take part in RNP is defined. HANA follows a heuristic approach and deals with pre-negotiation. The architecture defines how observed information must be represented. It solves the problem of supplying the negotiation strategy with plans and negotiating options assuming a huge space of possible solutions. To that end, it uses an anytime algorithm that runs in parallel to the negotiation strategy providing an up to date ranking of plans and options. The architecture takes into account the cost of exploring the space of possible solutions. Several negotiation strategies are proposed for the HANA agents. These strategies can be combined resulting in a dynamic negotiation strategy adjustable to the current available information and left time until the deadline is reached. The evolutive nature of the architecture is empowered by a search and negotiation technique. This technique corresponds to the use of the best ranked plans found by the plan search to generate the negotiation options being supplied to the negotiation strategy. And also, to the use of the information obtained from the negotiation to prune the search space and fine tune the evaluation of plans. The architecture is extensible allowing the incorporation of behavioural models that can trigger the agent intentions and affect the negotiation strategy that is guided by intentions. In fact, the world model of the agent is based on a graded BDI to simplify the incorporation of external models. Degrees are used to represent the uncertainty on the world that is basically related to the future actions to be performed by the other agents. A good usage of the search and negotiation technique reduces that uncertainty facilitating a successful performance.

The principal difference between HANA and other architectures is its extensibility —facilitating the incorporation of other research models, and its applicability to the industry. This is not only a theoretical work providing a new search and negotiation technique. It is also a practical work, [24], that takes into account the complexity of searching for proposals, and the uncertainty in the environment. It includes pre-negotiation that is necessary as refers to the study of negotiation opponents, the generation of possible negotiating options, and the planning of a strategy for the negotiation. Contrarily to other works allowing multiple bilateral negotiation, HANA is designed as a single negotiating agent (in [14, 36] a negotiator is itself a MAS), and it is able to firmly commit to the performance of actions. It does not use protocols including decommitment to avoid possible agreement overlap like in [17]. For all those reasons we state that this is a suitable negotiation architecture for software agents to deal with human counterparts.

Acknowledgments

Research supported by the Agreement Technologies CONSOLIDER project under contract CSD2007-0022 and INGENIO 2010, by the Agreement Technologies COST Action, IC0801, and by the Generalitat de Catalunya under the grant 2009-SGR-1434.

References

1. Fabregues, A., Sierra, C.: Hana: a human-aware negotiation architecture. Decision Support Systems (In press)
2. Raiffa, H.: Negotiation Analysis: The Science and Art of Collaborative Decision Making. Harvard U.P. (2002)
3. Kraus, S.: Strategic Negotiation in Multiagent Environments. MIT Press (2001)
4. Adams, J.S.: Inequity in social exchange. In Berkowitz, L., ed.: Advances in experimental social psychology. Volume 2. New York: Academic Press (1965)
5. Bazerman, M.H., Loewenstein, G.F., White, S.B.: Reversal of preference in allocation decisions: judging an alternative versus choosing among alternatives. Administration Science Quarterly (37) (1992) 220–240
6. Coricelli, G., Nagel, R.: Neural correlates of depth of strategic reasoning in medial prefrontal cortex. Proc. of the National Academy of Sciences **106**(23) (2009) 9163–9168
7. Eisenegger, C., Naef, M., Snozzi, R., Hienrichs, M., Fher, E.: Sequence of testosterone vs. placebo on ultimatum game behavior in women. Nature (463) (2010) 356–359
8. Bichler, M., Kersten, G., Strecker, S.: Towards a structured design of electronic negotiations. Group Decision and Negotiation **12** (2003) 311–335
9. Raisch, W.: The eMarketplace: Strategies for Success in B2B eCommerce. McGraw-Hill, Inc., New York, NY, USA (2002)
10. Shakun, M.: Multi-bilateral multi-issue e-negotiation in e-commerce with a tit-for-tat computer agent. Group Decision and Negotiation **14** (2005) 383–392
11. Pinho Jr., O., Ramalho, G., Paula, G., Tedesco, P.: Sequential bilateral negotiation. In: Advances in Artificial Intelligence - SBIA 2004. Volume 3171. (2004) 526–535
12. Van de Walle, B., Heitsch, S., Faratin, P.: Coping with one-to-many multi-criteria negotiations in electronic markets. In: Proc. of the 12th Int. Workshop on Database and Expert Systems Applications. (2001) 747–751
13. Nguyen, T.D., Jennings, N.R.: Coordinating multiple concurrent negotiations. In: Proc. of the 3rd Int. Conf. on Autonomous Agents and Multiagent Systems. (2004) 1064–1071

14. Rahwan, I., Kowalczyk, R., Pham, H.H.: Intelligent agents for automated one-to-many e-commerce negotiation. In: Proc. of the 25th Australasian Conf. on Computer science (ACSC '02). Volume 4. (2002) 197–204
15. Vahidov, R.: Situated decision support approach for managing multiple negotiations. In: Negotiation, Auctions, and Market Engineering. Volume 2. (2008) 179–189
16. Kraus, S., Lehmann, D., Ephrati, E.: An automated diplomacy player. In: Heuristic Programming in Artificial Intelligence: The 1st Computer Olympia. (1989) 134–153
17. Nguyen, T.D., Jennings, N.R.: Managing commitments in multiple concurrent negotiations. Electronic Commerce Research and Applications **4**(4) (2005) 362–376
18. Williams, C.R., Robu, V., Gerding, E.H., Jennings, N.R.: Negotiating concurrently with unknown opponents in complex, real-time domains. In: Proc. of the 20th European Conf. on Artificial Intelligence. (August 2012) 834–839
19. Saunders, H. In: Prenegotiation and Circum-negotiation: Arenas of the Peace Process, Washington D.C., United States (1996) 419–43
20. Munroe, S.: Motivation and Autonomy for Pre-Negotiation. PhD thesis, University of Southampton (September 2005)
21. Lopes, F., Mamede, N., Novais, A.Q., Coelho, H.: A negotiation model for autonomous computational agents: Formal description and empirical evaluation. Journal of Intelligence Fuzzy Systems **12**(3,4) (December 2002) 195–212
22. Li, C., Giampapa, J., Sycara, K.: Bilateral negotiation decisions with uncertain dynamic outside options. In: Proc. of the 1st IEEE Int. Workshop on Electronic Contracting. WEC '04, Washington, DC, USA, IEEE Computer Society (2004) 54–61
23. Zeng, D., Sycara, K.: Bayesian learning in negotiation. Int. Journal of Human-Computer Studies **48**(1) (January 1998) 125–141
24. Lopes, F., Wooldridge, M., Novais, A.Q.: Negotiation among autonomous computational agents: principles, analysis and challenges. Artificial Intelligence Review **29**(1) (2008) 1–44
25. Minsky, M.: The emotion machine: from pain to suffering. In: Proc. of the 3rd Conf. on Creativity & cognition (C&C '99), ACM (1999) 7–13
26. Casali, A., Godo, L., Sierra, C.: A graded bdi agent model to represent and reason about preferences. Artificial Intelligence **175** (2011) 1468–1478
27. Bratman, M.E.: Intention, Plans, and Practical Reason. Cambridge University Press (1999)
28. Rao, A.S., Georgeff, M.P.: Decision procedures for BDI logics. Journal of Logic and Computation **8**(3) (1998) 293–342
29. Alchourròn, C.E., Gärdenfors, P., Makinson, D.: On the logic of theory change: Partial meet contraction and revision functions. Journal of Symbolic Logic **50** (1985) 510–530
30. Hansson, S.O.: A Textbook of Belief Dynamics: Solutions to Exercises. Kluwer Academic Publishers, Norwell, MA, USA (2001)
31. Pilotti, P., Casali, A., Chesñevar, C.: An approach to automated agent negotiation using belief revision. In: Proc. of the 12th Argentine Symposium on Artificial Intelligence (40th JAIIO), Córdoba, Argentina (2011) 202–212
32. Fabregues, A., Madrenas, J., Sierra, C., Debenham, J.: Supplier performance in a digital ecosystem. In: Proc. of the IEEE Int. Conf. on Digital Ecosystems and Technologies (IEEE-DEST 2009), Istanbul (June 2009) 466–471
33. Sierra, C., Debenham, J.: The logic negotiation model. In: Proc. of the 6th Int. Conf. on Autonomous Agents and Multi-agent Systems (AAMAS 2007). (2007) 1026–1033
34. Faratin, P., Sierra, C., Jennings, N.R.: Negotiation decision functions for autonomous agents. Robotics and Autonomous Systems **24**(3–4) (1998) 159–182
35. Kaelbling, L.P., Littman, M.L., Cassandra, A.R.: Planning and acting in partially observable stochastic domains. Artificial Intelligence **101** (May 1998) 99–134
36. Kraus, S.: Designing and building a negotiating automated agent. Computational Intelligence **11** (1995) 132–171