# Applying game theory to automated negotiation

Ken Binmore and Nir Vulkan

*The ESRC Centre for Economic Learning and Social Evolution, University College London, Gower Street, London WC1E 6BT, UK*

With existing technology, it is already possible for personal agents to schedule meetings for their users, to write the small print of an agreement, and for agents to search the Internet for the cheapest price. But serious negotiation cranks the difficulty of the problem up several notches. In this paper, we review what game theory has to offer in the light of experience gained in programming automated agents within the ADEPT (Advance Decision Environment for Process Tasks) project, which is currently being used by British Telecom for some purposes.

## 1. Introduction

An *artificial agent* is a program that acts independently in furtherance of its user's interests. Some simple search agents can already be found on the internet (like BargainFinder and ShopBot) and others are being developed (by companies such as AgentSoft, Microsoft and IBM). The usage of agent technology is widely expected to increase and follow a similar path to that of graphical user interfaces (GUI) – first it will be an optional extra, but over time, products without intelligent agents will no longer be viable. The type and number of agents will increase as their intelligence and autonomy develop. However, progress will depend on the ability of agents to communicate with each other, share limited resources and negotiate agreements. Bearing in mind that today's co-ordination languages, such as Stanford's KQML, are rich enough to enable automated agents to interact meaningfully, this becomes particularly important.

Machines that "make deals" are familiar to computer scientists, but in the future they will no longer be a part of the same system, perhaps not even designed by the same people. Interacting agents will try to satisfy only their user's goals and will no longer have a notion of global utility. Agents will, therefore, find themselves in situations where they have an incentive to lie, or act tough, or exploit other strategic avenues not usually associated with machines.

Although the issues described above are relatively new to the AI community, they closely resemble the models studied by economists and game theorists. The sub-field of microeconomics known as game theory is widely acknowledged to provide the best available set of tools for the design of multi-agent architectures (for a good review of the topic see Rosenschein and Zlotkin [13]). However, because existing theory is sometimes misunderstood or needs further development for computer applications, researchers have been reluctant to use game theory and those that do, tend to miss out on the more recent advances, which we believe are likely to prove most useful in such settings. Following Varian [18] and others, we seek to bridge the gap between the strategic approach of microeconomics and distributed AI by describing some of the problems and opportunities involved in applying game theory and mechanism design to automated agents. Our ideas are based on experience obtained in working with researchers from the Distributed AI unit at QMW on the negotiation modules of their much publicized ADEPT system. Our goal is to give an overview of some of the challenges raised by such implementations, without getting into project-specific details.

### *1.1. The ADEPT project*

In the ADEPT (Advance Decision Environment for Process Tasks) project negotiating agents were developed for business process management applications. The first versions were jointly developed by the Distributed AI unit at QMW college, and British Telecom. BT plans to use ADEPT in several of its processes, like the "Provide Customer Quote Business Process": In this process, before designing a network to provide some services to a customer, a quotation is provided. Up to six departments are involved in the process where each of those departments is being replaced by an agent. Agents negotiate prices, a deadline for the completion of services (with penalties if these deadlines are violated), and the quality of the network services.

Unlike most existing Internet search agents, the ADEPT agents are computationally strong, and are able to make many different kinds of decisions. Each agent has a huge memory, a strong CPU at its disposal and access to all relevant (private) information. Each agent is built from several modules, each with its own responsibilities and tasks. The Situation Assessment Module (SAM) makes the decision to start negotiating (or to renegotiate), while the Interaction Management Module (IMM) is in charge of the actual process of negotiation. The IMM is the "game theorist" of the agent, and while higher level modules are able to interfere in extreme circumstances (such as an unexpected change in priorities caused by new information), it alone has control over the decisions being made at any stage of the negotiation. The IMM is divided into a Mega-level Domain Independent Reasoner (DIR) where strategies are chosen (the words "tactic" often substitutes for what is called a "strategy" in game theory), and a specific DIR process which is active during negotiation. See Jennings et al. [10] and Sierra et al. [17] for more details.

As far as we know, the ADEPT agents are the "strongest" agents amongst existing multi-agent applications. While the range of bargaining situations they face is very complex from the point of view of economic modeling, their structure lends itself to implementations of game-theoretic ideas and models.

## 2. Basic differences

In economic theory distinction is made between models in which we analyse the optimal behavior of individuals or firms *given* the underlying mechanism (or rules of the game), and models in which we study optimal mechanism design *given* that agents behave optimally. In the current early stages of multi-agent design these two approaches are being developed simultaneously. One cannot compare two different protocols (mechanisms) without specifying the behaviour of the interacting agents. Similarly, one cannot design optimising agents without some information about the protocols governing their interaction. However, we think it clarifies the underlying philosophy to maintain a clear distinction between the design of protocols and the design of the agents who operate within the rules specified by the protocol.

In current systems like ADEPT, bargaining automated agents are programmed with rules-of-thumb distilled from intuitions about good behavioral practice in human negotiations. The danger is that the programmer may not be fully aware of the circumstances to which human behavioral practice is adapted, and hence use behavioral rules that are capable of being badly exploited by new agents that have been programmed to take advantage of the weaknesses of the agents currently in plan. When protocols that have been deliberately constructed to take advantages are available within the artificial environment of a computing system, the risks of creating the opportunity for such destabilising invasions by new agents are particularly large.

Economists believe that their approach provides an escape route from these difficulties. In principle, an agent should be designed to *optimize* on behalf of the decision maker whose role it usurps. The revelation principle of mechanism design applies also to agent design, and so the designer of a properly engineered agent can tell his client that his programming takes care of all the *strategic* problems involved in bargaining optimally. This leaves the client to report *truthfully* on his preferences and his information. This may not always be easy for the client to understand. For example, evidence from the recent Guttman and Maes electronic-agent marketplace experiment at MIT shows that users consistently *lied* to agents that they had designed for themselves, because they thought they could get more by giving the impression of being tougher than they are. Similar problems have occurred with ADEPT. With a properly designed agent, it would always be a mistake to tell such lies. If it is optimal to pretend to be tough, the agent will do all the pretending necessary.

The need to get true data in an appropriate form from the client poses both psychological and theoretical problems in creating a suitable interface. In particular, the client may not have thought about what his fundamental preferences are in sufficient detail and must therefore be prompted with a suitable set of questions. Computer scientists who want to employ the methodology advanced here must also learn enough economics to distinguish between what economists call direct and indirect utilities. I may prefer one action to another as a means to an end. My utility over the actions is therefore *indirect,* because it is derived from looking at my preferences over the final ends. What an agent needs from a client is coherent data about his *direct* utility over final ends. This will remain *fixed* as his information changes. Where a client is unable to supply the data in an adequate form, the agent will need to make an attempt to construct direct utility functions from whatever data is available.

In principle, once each agent has the necessary data from his client, there is no need for any simulation of the bargaining process. Game theory should provide a prediction of the outcome that would follow the use of optimal strategies that can be employed immediately. For example, Rosenschein and Zlotkin [13] are right to draw attention to the fact that the Nash bargaining solution will sometimes provide a suitable prediction – although they are over-optimistic about the range of environments in which it can be used in an unmodified form.

The prediction that game theorists make must constitute an *equilibrium* of the chosen protocol, otherwise new exploitative agents will be able to invade and disrupt the system. However, *cognoscenti* will be aware that the problem of equilibrium *selection* arises in a particularly aggravated form when studying bargaining with incomplete information. Fortunately, operating within a computer context torpedoes most of the difficulties that arise when trying to model players as people:

(1) For some protocols, the system itself can *choose* an equilibrium in an unproblematic manner.

(2) When the choice of an equilibrium selection norm would itself give rise to bargaining problems among the players, the equilibrium refinement theories of game theory can be given new life, because we have a specific *model* of a player in a computer context. We therefore have a noncontroversial means of interpreting the counterfactuals involved when observing that optimizing players stay on the equilibrium path in a game because of what *would* happen if they were to deviate.

(3) Where analytic approaches fail, algorithmic methods for computing fixpoints corresponding to equilibria can be realistically employed.

We therefore have hopes that introducing game-theoretic methods into automated negotiations will not only provide sound design principles for computer scientists but will also create an area of applications for game theory that eliminates many of the problems that have hindered advances in some directions.

## 3. Classification of environments based on extensive form

One of the main reasons that game theory may be useful in the artificial agent context is that it provides us with a classification of interacting situations based on their extensive forms. It provides us with the tools to say that two, seemingly different, situations are strategically equivalent because they can be translated into the same game form. In open architectures, like ADEPT, the interacting agent can find itself in many different bargaining environments. A service provider, for example, could find itself the sole provider in some settings, one of two in others, or one of six. All are very different strategic settings, despite the apparent similarities. It is unlikely that the same behavioral algorithm for an agent will prove equally useful in all of these settings. Game theorists are sensitive to such differences (in a manner unusual in computer science), and it is an interesting challenge to see if we could teach our automated agents to become similarly sensitive.

The existence of multiple-game situations raises some interesting questions for the design of automated agents. As long as information gathering is not observable, we would like our agent to know as much as possible about the game it is about to play (see Hurkens and Vulkan [9]). But how do we do that? How can we possibly envisage the situations the agent might face? In the real-world people use rules of thumb to classify situations, which do well on the average, and in the absence of better information we would be wise to equip our agents with such rules. In other words, agents will have to decide what to pay attention to. The challenge we are facing is to design agents who are more than just a dictionary of optimal strategies for a stereotyped set of game-types, but instead a "thinking machine" in the genuine AI sense of these words – one that learns what it needs to pay attention to in its environment. Our impression is that economic theory and game theory might benefit hugely from attempting such an exercise[1].

It may help to appreciate a game theorist's sensitivity to an agent's environment, to observe that the fact that agents may gather information about their environment is also a fact about the environment. If agents' information gathering is observable, it has strategic value. More specifically, suppose that some of the features of the game to be played are determined by a random move and that agents are able to obtain (partial) information about the future state of the world (possibly with a cost). Then, if information gathering is observable an agent may use it to signal its choice of strategy in the actual game to be played. For example, an agent might choose not to be informed at all, because it will imply a continuation equilibrium where he is better off than in any (or some) of the states where it is informed. A designer of agents must therefore consider the strategic effects of the public actions of their agents. For example, if agents can credibly signal that they are *not* going to gather information, then this may be used to deter potential competition (see Hurkens and Vulkan [9] and Vulkan [19] for more details, or see Matthews [11] for a similar study within the context of auctions).

In the version of ADEPT which is being used by BT sensitivity to the environment is largely confined to the number of agents who simultaneously offer or request the same service. However, the distinction is service-specific, and agents are able to determine with certainty the number of other players before negotiation begins, although it is becoming clear that new versions of the system will need to account for situations in which the agent cannot tell, without further inquiry, the kind of game he is playing. However, even the case in which the number of interacting agents is given raises strategic issues that are largely neglected by ADEPT.

## 4. One-to-one bargaining

The following two-person bargaining model is based on the ADEPT format. Players 1 and 2 bargain over which possible agreement in a set A to implement.

[1] In a similar setting, Hurkens and Vulkan [9] showed that an equilibrium where each of the agents is paying attention to different features of the game is a likely outcome of such meta games.

We model an agreement as a pair $a = (a1, a2)$ of real numbers. Each player has a reservation value $r(i)$ so that only agreements satisfying $a \geqslant r$ are viable (if player 2 is the only potential customer for an object that player 1 has to sell, then $r(1)$ and $r(2)$ are the players' respective valuations of the object to themselves. If player 2 pays player 1 an amount $m$ for the object, then $a = (r(2) - m, m - r(1))$. Each player is also assumed to care about the time at which agreement is reached. The cost to players of a delay of length $t$ in reaching an agreement is taken to be $c(i)t$. In ADEPT, at least one player is assumed to have a finite deadline $d(i)$. Agent $i$'s utility for an agreement $a$ at time $t$ is then given by $a(i) - c(i)t$ when $t \leqslant d(i)$. If a player breaks off the negotiation to take up his best outside option, or else a deadline is breached, then each receives a breakdown payoff $b(i) - c(i)t$ (if player 2 is only able to buy a suitable object from player 1, then $b(2) = 0$ and $b(1) = r(1)$).

In the general case, a player's parameters will only be known to himself. The other player's beliefs about the parameters are calculated from a prior probability distribution. A major task for an agent programmed to investigate his environment will be to learn what he can about such distributions from observing the play of his opponents and of agents involved in other publicly observable games. He must also seek to predict what they know about the distribution, and what they know about what he knows – and so on. However, we short-circuit such considerations here by assuming that the relevant distribution is common knowledge (it can be deduced from a publicly observable event).

The problem will often be simplified because it is commonly known that some of the parameters take extreme values. For example, it may be commonly known that $c(1) = c(2) = c$, where $c$ is vanishingly small, or that $r(1) = 0$ because the object is a service and hence worthless to the seller. However, neither $r(2)$ nor $d(2)$ are likely to be commonly known. If they were, player 1 would be in a perfect position, because he could wait until player 2's deadline and then make a take-it-or-leave-it demand of $r(2) - \varepsilon$. If $b(2) = 0$, player 2 would then be left with a choice between $\varepsilon > 0$ and zero. If she is rational, she would accept the former.

In the model, negotiation follows a process of offer and counteroffer first successfully analysed in the infinite horizon case by Rubinstein [14]. After an offer is made, the current responder has three choices: (a) she can accept, (b) she can refuse and opt out, (c) she can refuse but continue to negotiate. In the last case, an interval of length $\Delta > 0$ elapses before she can make her counteroffer. In principle, this process can continue forever.

The apparent problem created by a bargaining process with a potentially infinite horizon is illusory, since it can be implemented by introducing a random stopping time at the expense of complicating the model only slightly. It is dangerous to terminate the process at a fixed stopping time that the agents know or can learn as in the current ADEPT system. The reason is that the effect is the same as introducing an artificial commonly known deadline that may be exploited as described above.

Perhaps the most interesting case is when $\Delta$ is made arbitrarily small. If players are allowed to delay making a counteroffer, they can use this opportunity for strategic signalling of their strength (as in Admati and Perry [1]). As in most models of bargaining with incomplete information (e.g., Rubinstein [15,16], Gul and Sonnenschein [8]), the set of (sequential) equilibria in the general case is large and complex. This is nearly always true in models of two-sided uncertainty (Chatterjee and Samuleson [5,6]). In particular there are multiple equilibria in some of which agreements may be delayed for many periods (this remains true even if we use the refined solution concept of perfect sequential equilibrium, as studied by Grossman and Perry [7]). Strategic delay was first studied by Admati and Perry [1]. Equilibria involving strategic delay are "type revealing", i.e., agents delay until the opponent cracks and reveals his type. Once this has been established agreement can be reached immediately. Of course, the number of rounds before agreement depends on the number of types. Admati and Perry only consider one-sided uncertainty. In a model of two-sided uncertainty with strategic delays, it is possible to obtain equilibria where the true types of players are never revealed. This happens because, under certain assumptions, if a player knows that its opponent is lying about his type, its best response is to do the same.

In general, we have "more" uncertainty than in any of the models successfully analysed by orthodox game theory methods. But we also have major advantages facilitating a successful analysis that are not available outside a computer context. First, we have control over the protocol, which can be altered where necessary to remove difficulties. Introducing some randomization into the model can often simplify matters considerably. Second, we have a specific model of an agent that allows us to deal with the equilibrium selection problem by legitimizing the refinement criteria that are so difficult to justify when applied to human players (section 2).

## 5. One-to-many negotiations

Within ADEPT, the structure of most concern involves a single requirer of a specific and limited service, who has to decide which of many agents who offer the service will provide it. Binmore's [2] adaptation of Rubinstein's bargaining model to the multi-person case shows that the problem reduces to an auction in the perfect information case, in which the service is provided by the agent willing to do it most cheaply. When information complexities are present, traditional bargaining may still retain a role. For example, an automated agent in an open architecture might not be certain about the number of providers or the quality of the service they advertise. He might therefore bargain in parallel to the use of an auction.

One-to-many negotiations may also be embodied in a much wider market context involving search. When the products being traded are individualistic, as in the housing market, such a scenario is inevitable. One must then envisage negotiations taking place while the negotiators continue to search. If one finds a new partner, the potential for auctioning arises. This idea of bargaining and search with incomplete information is quite well developed (Chatterjee and Lee [4]).

However, we follow the standard view among economists that an auction is an effective way of resolving the "one-to-many" bargaining problem. The case for implementing auctions in a world of automated agents is even stronger than for people. This is true for two main reasons: First, fast agreements with less communication are preferred not only directly (via the agents' utility functions), but also indirectly: Distributed systems, like ADEPT or the Internet, can support only a limited amount of communication at any given time. Intra-agent communication can, therefore, be seen as carrying a negative externality. Secondly, some of the better auction structures require that players be able to compute complex strategies. With automated agents, this is not a problem, since calculation is cheap.

## 6. Additional implementation problems

The *implementation* itself may create problems which are specific to the trading mechanism and distributed system in question. For example, in the original version of a Dutch auction announcements are made publicly. On a distributed system where information might reach different users with some delay, this becomes a problem (this problem was first reported in [12] in Rodríguez et al.'s implementation of the Spanish fish market). If the question of timing and common knowledge of bids is not resolved, it will become possible to benefit from arbitrage. Timing is only one example for the problem faced by those who implement trading mechanism on distributed systems. Problems can also arise regarding issues such as security, identity and others. In fact, the scale and the exact nature will only become known during the implementation process. Moreover, it is often not possible (or it is cost-ineffective), to change the technology so that it would "fit the model". Instead, an applied theorist could investigate the implications on the model of whatever problems which may arise. If incentives exist for individuals to behave in such a way that the mechanism could become inefficient, it is mostly possible to change some details of the mechanism so that such behavior is no longer desirable for the participants.

## 7. Final remarks

Agent technology is likely to have a huge impact on the future of the Internet and electronic commerce. The growing numbers of applications where automated agents interact amongst themselves without user intervention provides us with a new field where game theory and economic theory can be applied. Economic agents have been criticized in the past as being more like computer programs than real human beings, so it is interesting to see just how useful our models could be in the design of agents and protocols. Our initial experience shows that existing models still need to be considerably modified to become useful in these new settings. In this paper we tried to underline the theoretical difficulties and opportunities that arise from these implementations, the kind of changes to existing models that are required and how we believe progress can be made.

## Acknowledgements

Financial support from the ESRC and the Leverhulme trust is gratefully acknowledged. The ADEPT system described in this paper was developed jointly by BT and the Distributed AI unit at the department of Electronic Engineering, QMW college, University of London. We thank Nick Jennings, Carles Sierra and Peyman Farain for their time and co-operation.

## References

[1] A.R. Admati and M. Perry, Strategic delay in bargaining, Rev. Econom. Stud. 54 (1987) 345–364.

[2] K. Binmore, Bargaining and coalitions, in: *Game Theoretic Models of Bargaining*, ed. A.E. Roth (Cambridge Univ. Press, Cambridge, 1985) pp. 269–304.

[3] K. Binmore, Modeling rational players, Parts I and II, Economics and Philosophy 3 (1987) 179–214; 4 (1985) 9–55.

[4] K. Chatterjee and C.C. Lee, Bargaining and search with incomplete information about outside options, mimeo (1996).

[5] K. Chatterjee and L. Samuleson, Bargaining with two-sided incomplete information: An infinite horizon model with alternating offers, Rev. Econom. Stud. 54 (1987) 175–192.

[6] K. Chatterjee and L. Samuleson, Bargaining under two-sided incomplete information: The unrestricted case, Oper. Res. 36 (1988) 605–618.

[7] S.J. Grossman and M. Perry, Sequential bargaining under asymmetric information, J. Econom. Theory 39 (1986) 120–154.

[8] F. Gul and H. Sonnenschein, On delay in bargaining with one-sided uncertainty, Econometrica 56 (1988) 601–611.

[9] S. Hurkens and N. Vulkan, Information acquisition and entry, Discussion paper 96-23, University College London (1996).

[10] N.R. Jennings, P. Farain, M.J. Johnson, T.J. Norman, P. O'Brein and M.E. Wiegand, Agent-based business process management, Internat. J. Cooperative Inform. Systems 5(2–3) (1996) 105–130.

[11] S. Matthews, Information acquisition in discriminatory auctions, in: *Bayesian Models in Economic Theory*, eds. M. Boyer and R.E. Kihlstrom (Elsevier, Amsterdam, 1984).

[12] A.J. Rodríguez, P. Noriega, C. Sierra and J. Padget, FM96.5 – A Java-based electronic auction house, in: *2nd Internat. Conf. on the Practical Application of Intelligent Agents and Multi-Agent Technology*: *PAAM '97* (1997).

[13] J. Rosenschein and G. Zlotkin, *Rules of Encounter* (MIT Press, Cambridge, MA, 1994).

[14] A. Rubinstein, Perfect equilibrium in a bargaining model, Econometrica 50 (1982) 207–211.

[15] A. Rubinstein, A bargaining model with incomplete information about time preference, Econometrica 53 (1985) 1151–1172.

[16] A. Rubinstein, Choice of conjectures in a bargaining game with incomplete information, in: *Game Theoretic Models of Bargaining*, ed. A.E. Roth (Cambridge Univ. Press, Cambridge, 1985) pp. 99–114.

[17] C. Sierra, P. Faratin and N.R. Jennings, A service-oriented negotiation model between autonomous agents, in: *Proc. of Modeling Autonomous Agents in a Multi-Agent World (MAAMAW)* (1997).

[18] R.H. Varian, Economic mechanism design for computerised agents, in: *USENIX Workshop on Electronic Commerce*, New York (July 1995).

[19] N. Vulkan, Uncertainty, information acquisition and economic equilibria, Ph.D. thesis, University College London (1996).