ELSEVIER

Artificial Intelligence 94 (1997) 79–97

Artificial Intelligence

Negotiation and cooperation in multi-agent environments *

Sarit Kraus [a,b,1]

[a] *Department of Mathematics and Computer Science, Bar Ilan University, Ramat Gan, 52900 Israel*
[b] *Institute for Advanced Computer Studies, University of Maryland, College Park, MD 20742, USA*

Abstract

Automated intelligent agents inhabiting a shared environment must coordinate their activities. Cooperation—not merely coordination—may improve the performance of the individual agents or the overall behavior of the system they form. Research in Distributed Artificial Intelligence (DAI) addresses the problem of designing automated intelligent systems which interact effectively. DAI is not the only field to take on the challenge of understanding cooperation and coordination. There are a variety of other multi-entity environments in which the entities coordinate their activity and cooperate. Among them are groups of people, animals, particles, and computers. We argue that in order to address the challenge of building coordinated and collaborated intelligent agents, it is beneficial to combine AI techniques with methods and techniques from a range of multi-entity fields, such as game theory, operations research, physics and philosophy. To support this claim, we describe some of our projects, where we have successfully taken an interdisciplinary approach. We demonstrate the benefits in applying multi-entity methodologies and show the adaptations, modifications and extensions necessary for solving the DAI problems. © 1997 Elsevier Science B.V.

Keywords: Distributed Artificial Intelligence; Multi-agent systems; Cooperation; Negotiation

1. Introduction

One of the greatest challenges for computer science is building computer systems that can work together. The integration of automated systems has always been a challenge,

* This is an extended version of a lecture presented upon receipt of the Computers and Thought Award at the 14th International Joint Conference on Artificial Intelligence in Montreal, Canada, August 1995.

[1] Email: sarit@cs.biu.ac.il or sarit@umiacs.umd.edu.

PII S0004-3702(97)00025-8

but as computers have become more sophisticated, the demands for coordination and cooperation have become more critical. It is not only basic level components such as printers, disks, and CPUs, but also high-level complex systems that need to coordinate and cooperate.

Examples of such intelligent systems include:

- automated agents that monitor electricity transformation networks [32];
- teams of robotic systems acting in hostile environments [5];
- computational agents that facilitate distributed design and engineering [54];
- distributed transportation and planning systems [25,56];
- intelligent agents that negotiate over meeting scheduling options on behalf of people for whom they work [67]; and
- Internet agents that collaborate to provide updated information to their users.

In these environments, even when coordination is not required, cooperation may improve the performance of the individual agents or the overall behavior of the system they form.

Problems of coordination and cooperation are not unique to computer systems, but exist at multiple levels of activity in a wide range of populations. People pursue their own goals through communication and cooperation with other people or machines. Animals interact (with limited language), cooperate with each other, and form communities. Particles interact with each other and compose different types of material and phases of matter. Although most computers currently act in multicomputer environments, the interaction among them is generally restricted, and they interact under strict rules. Negotiation or other sophisticated interactions rarely occur among computers. In general, the levels of negotiation, bidding, voting, and other sophisticated interactions that characterize natural coordinating systems are absent.

Recent research in Distributed Artificial Intelligence (DAI) aims to increase the power, efficiency, and flexibility of intelligent automated systems (agents) by developing sophisticated techniques for communication and cooperation among them. In my research, I have addressed the challenge of building coordinated and collaborated intelligent agents by combining AI techniques with methods and techniques from various fields that study multi-entity behavior.

I argue that an interdisciplinary approach is beneficial for the development of coordinated and cooperative intelligent agents. Because these fields, which study multi-entity behavior, are not concerned with agent design, one might think that they are not relevant for DAI. Our experience is quite the contrary. It is true that these fields do not solve AI problems, but they have thought about a wide range of issues that are important to the design of intelligent agents, and they provide techniques, sometimes with proven properties or methods for proving properties that are useful to adopt for designing agents. DAI researchers still have a lot of work left in order to adapt these methods for their needs; however, they do not need to start from scratch. In this paper, we show by example the advantages and the challenges of building on other work.

The amount of work done in the related fields is overwhelming. Thus, a major challenge in taking an interdisciplinary approach is determining which technique to use. There are several parameters that influence the choice of the appropriate techniques for a DAI application:

(1) *The level of cooperation among the agents*: cooperative agents which work toward satisfying the same goal versus agents which are self-motivated and try to maximize their own benefits.[2] There are intermediary cases where self-motivated agents join together to work toward a joint goal.

(2) *Regulations and protocols*: environments where the designers of the agents can agree on regulations and protocols for the agents' interaction versus situations with no pre-defined regulations and protocols.

(3) *Number of agents*: a very large number of agents (hundred or more) versus a few agents which communicate and coordinate their actions.

(4) *Type of agents*: systems of automated agents versus systems composed of people and automated agents.

(5) *Communication and computation costs*: the availability and cost of communication among the agents and their computation capabilities and costs.

Any DAI task can be characterized according to these dimensions. This characterization guides the choice of the multi-entity technique that can be applied to the specific task.

Consider the development of automated agents for buying and selling items on the Web, such as clothes and furniture. Suppose there are several enterprises, each with several kinds of goods which they sell to users or to other enterprises. Each enterprise has intelligent seller and buyer agents. The job of the seller agent is to sell the enterprise's goods to other enterprises through their buyer agents or to users. The job of a buyer agent is to obtain from other enterprises the goods that are missing from the stock of its enterprise. Several different DAI problems may arise in such a framework:

(A) In the interaction between two automated agents belonging to different enterprises, the agents are self-motivated, but may benefit from cooperation. The designers of the agents may agree upon regulations for the interaction, the number of agents of each interaction is limited, and they can communicate and have computation capabilities.

(B) A seller agent of an enterprise may try to sell some goods to a person. In this case, the person will prefer a non-structured interaction, and it is more difficult to set regulations and protocols for the interaction in advance.

(C) Two agents of the same enterprise may work together toward the same goal: increasing the benefits of their enterprise. In this case, the agents are cooperative, regulations and protocols can be set in advance, the number of agents is limited, they are automated, and they can communicate.

In each of these three cases, there is a different multi-entity technique that should be applied.

In this paper, we will examine different DAI tasks and will discuss the application of game-theoretic techniques (Section 2), physics models (Section 3), operations research methods (Section 4), and informal models of cooperation and coordination (Section 5) to DAI environments.

[2] Research in DAI is divided into two basic classes: *Distributed Problem Solving* (DPS) and *Multi-Agent Systems* (MA) [6]. Cooperative agents belong to the DPS class, while self-motivated agents belong to the MA class.

2. The application of game-theoretic techniques to multi-agent environments

Researchers in DAI have considered problems related to task allocation and resource sharing where the agents are self-motivated, as in the following examples:

- situations where airplanes belonging to different airlines need to share the limited resources of the same airport, and it is necessary to find a mechanism that will give priority to planes with less fuel on board [61];
- an electronic market populated with automated agents which represent different enterprises and buy and sell (e.g., [8,17,74]);
- transportation centers that deliver packages and may cooperate to reduce expenses [64];
- information servers that form coalitions for answering queries [36]; and
- intelligent agents that negotiate over meeting scheduling options on behalf of people for whom they work [67].

Using the five criteria presented in the Introduction to characterize these examples, we observe that in these examples the agents are self-motivated and try to maximize their own benefits. The designers of the agents may agree in advance on regulations and protocols for the agents' interaction. In each interaction the number of agents is usually small (less than a dozen agents); there are only automated agents which can communicate and have computational capabilities.

In such situations we recommend the application of game-theoretic techniques. Game theory studies mathematical models of conflict and cooperation between people. The models of game theory are highly abstract representations of classes of real life situations that involve individuals who have different goals or preferences [49]. The active entity in all game-theoretic models is a *player*. Game-theoretic models are divided into two main types: "noncooperative" models, in which the sets of possible actions of *individual* players are primitives, and "cooperative" models, in which the sets of possible joint actions of *groups* of players are primitive [53].

The abstract models of game theory can be used as a basis for the agents' interaction protocols, when the designers of the agents agree to use them. Automated agents in a DAI framework can be modeled by players of game-theoretic models. Since it is assumed in game theory that the players are self-motivated, so should the agents be in the environments where these techniques are applied. DAI addresses both situations in which each individual agent acts by itself (e.g., [17,61,67]), thus calling for the application of noncooperative models, and situations in which tasks require that groups of agents work together (e.g., [36,64]), thus calling for cooperative models. In the first case, game-theoretic models are appropriate when there is only a handful of agents, and, in the second case, game theory may be applied to a few dozens of agents. The use of game-theoretic techniques requires substantial computations, and, communication capabilities are also usually needed. The examples described in the beginning of the section, as well as other situations that satisfy these criteria, are cases where applying game-theoretic techniques by DAI researchers should be considered. We have applied both noncooperative and cooperative game-theoretic models to DAI situations. We briefly describe these attempts here.

2.1. *Use of a strategic model of negotiation for resource sharing and task distribution*

We first consider situations where a small number of self-motivated agents need to share resources or can benefit from task distribution. In these situations, inter-agent cooperation can be enhanced using negotiation strategies that enable agents to communicate their respective desires and to compromise in order to reach mutually beneficial agreements.

The game-theoretic strategic approach to the bargaining problem [3] provides a useful foundation for designing such capabilities in systems. In this approach, agents' negotiating maneuvers are moves in a noncooperative game, and the rationality assumption of the negotiators [30] is expressed by using the *Nash Equilibrium* concept. [4] Our main goal in this research was to define an acceptable protocol for the interactions among the agents and to identify strategies for the agents participating in the negotiation. These methods are applicable in the following example.

Example 1 (*Data allocation in multi-agent environments*). There is a set of several (more than two) information servers in the environment which are connected by a communication network. Each server is located in a different geographical area and receives queries from clients in its area. In response to a client's query, a server sends back information stored locally or information stored in another server, which it retrieves from that server. [5] The information is clustered in datasets. [6]

When a set of new datasets arrives, each new dataset has to be allocated to one of the servers by mutual agreement among all of them. However, each server has its own interests and wants to maximize its own utility, and thus the servers may be in conflict concerning where to locate the new datasets. Furthermore, the servers have no common interest and no central controller which can be used to resolve such conflicts. We propose that these conflicts will be resolved via negotiations. In particular, we propose a strategic negotiation model that takes into account the passage of time during the negotiation process itself in order to solve this problem.

Our negotiation protocol is a process that may include several iterations. We assume that servers can take actions in the negotiation only at certain times in the set *Time* = $\{0, 1, 2, \ldots\}$ that are fixed in advance and ordered (randomly). In each period $t \in$ *Time*, if the negotiation has not terminated earlier, a server whose turn it is to make an offer at time t, will suggest a possible allocation for all the datasets considered, and each of the other servers may either accept the offer or reject it or opt out of the negotiation. If an offer is accepted by all the servers, then the negotiation ends, and this offer is implemented. If at least one of the servers opts out, then the negotiation ends. If no server has opted out, but at least one of the servers has rejected the offer, the negotiation

[3] Introductory books on game theory that discuss approaches to bargaining include [21,47,49,53].

[4] A pair of strategies (σ, τ) is a Nash Equilibrium if, given τ, no strategy of Agent 1 results in an outcome that Agent 1 prefers to the outcome generated by (σ, τ), and similarly for Agent 2, given σ.

[5] A specific example of such a distributed knowledge system is the Data and Information System component of the Earth Observing System (EOSDIS) of NASA [50]. It is a distributed system which supports the archiving and distribution of data at multiple and independent data centers.

[6] A dataset corresponds to a *cluster* in information retrieval, and to a *file* in the file allocation problem.

proceeds to period $t+1$, and the next server makes a counter-offer, the other servers respond, and so on.

Using this negotiation mechanism, we showed that the servers have simple and stable negotiation strategies that result in efficient agreements without delays. We have proved that our methods yield better results than the static allocation policy currently used for data allocation for servers in distributed systems.

The main question is, in general, what is the advantage of using game-theoretic models for such problems, and what must be done in order to adapt them to DAI environments. The strategic bargaining theory provides general frameworks for modeling negotiation, but to apply them to the design of agents, we needed to address five problems: choosing a strategic bargaining model which is applicable for the specific DAI problem; matching the DAI scenarios with the game-theoretic definitions of the chosen model; identifying equilibrium strategies; developing low complexity techniques for searching for appropriate strategies; and providing utility functions.

For example, for the data allocation problem described in Example 1, we have chosen Rubinstein's model of Alternative Offers [62].[7] The main property of this model is that it takes into consideration the passage of time during the negotiation. This is useful for environments of Example 1 since for a server participating in the negotiation process, the time when an agreement is reached is very important.[8] The model of Alternative Offers provides formal definitions of players, possible agreements, the protocol of alternative offers, and the notion of strategies. In order to apply these concepts to the data allocation problem, we had to match the world state and formal definitions and modify them. For example, in the data allocation scenario, a player is a server and an agreement is a distribution of datasets to information servers.

Game theory proposes different notions of equilibria that capture different aspects of stability. Given specific assumptions about the environments, game theory researchers identify strategies that are in equilibrium. In order to address the third need mentioned above, when applying game-theoretic techniques to DAI environments, we formalized the assumptions that are appropriate for our environments. For example, in the data allocation scenario, all agents sustain a loss over time, there is a finite (but large) set of agreements, and there are some agreements which are better for all agents than opting out of the negotiations. In most of the cases, these assumptions are different from the assumptions that are considered in game theory, and therefore we needed to identify the equilibrium strategies under the DAI assumptions.

The third problem mentioned above arises in DAI situations where the designer of the system cannot provide the automated agent with a negotiation strategy in advance. For example, in the data allocation scenario, finding possible dataset allocations can be done only after the specifications of the datasets are known to the agents and thus cannot

[7] See [52] for a detailed review of the bargaining game of Alternative Offers.

[8] There are two reasons for this. First, there is the cost of communication and computation time spent on the negotiation. Second, there is the loss of unused information: until an agreement is reached, new documents cannot be used. Thus, the servers wish to reach an agreement as soon as possible, since they receive payment for answering queries.

be supplied in advance by the designers. Construction of the strategies which are in equilibrium can rely on theorems proven in advance, but can be done only when the set of possible agreements can be defined. For such situations, there is a need to develop low complexity computational techniques for searching for appropriate strategies by the automated negotiators. The issue of the complexity involved in finding strategies is not discussed in the game theory literature.

Another issue that is rarely discussed in game theory is the source of a utility function or a set of preferences that is needed for any decision-making. In game theory, one aspect of a definition of a game is the players' utility functions or preferences, and it is assumed that each player knows its utility function (and has some knowledge of the utility function of its opponents). A designer of an automated agent is required to *provide* the agents with a utility function or a preference relation. Without doing so, the game-theoretic techniques cannot be used for automated agents. In the data allocation scenario, we have developed a complex utility function which takes into consideration factors such as storage costs, retrieval costs, distances between servers, etc. Only then can we apply game-theoretic techniques.

More details on our work on the strategic model of negotiation and the definition of utility functions can be found in [39,44,45,65]. In the process of developing and specifying the strategic model of negotiation, we have examined bilateral negotiations, as well as multi-agent environments (more than two agents), single encounters and multiple encounters, situations characterized both by complete and incomplete information, and the differing impact of time on the payoffs of the participants [39,45]. Recently, we have also considered problems where there are two attributes to the agreements [65]. While some combinations of these factors can result in minor delays in reaching an agreement, the model nevertheless reveals an important capacity for reaching agreement in early periods of the negotiation.[9]

2.2. *The game theory approach to coalition formation*

By creating coalitions that allow them to share resources and cooperate on task execution, autonomous agents may be able to increase their benefits. Cooperative game-theoretic models can be used to do this for self-motivated agents, each of which has tasks it must fulfill and resources it needs to complete these tasks. Although the agents can act and reach goals by themselves, it may be advantageous to join together.

For example, taxi drivers may own different types of cabs and therefore may have different costs, different transportation capabilities, and different resulting payoffs. Each taxi driver would like to increase his own benefits, but it may be in the driver's interest to cooperate and form coalitions in order to achieve greater and more complex transportation capabilities. Game-theoretic coalition formation theories can be used in the development of automated agents that represent these drivers as they form coalitions.

Game theory [11,28,34,51,57] provides a good framework with concepts of a coalition and coalitional value and different notions of stability, but to use it, we have had

[9] In [44], it was shown how the strategic model can be used in applications such as a hostage crisis simulation.

to address three tasks: the development of explicit protocols for interaction among the agents; the development of algorithms for coalition formation; while simultaneously taking into account communication costs and limited computation time. Most of the work in game theory does not treat these issues, but only predicts how the players will distribute the benefits, given a coalition configuration.

In [68,72] we addressed the three tasks mentioned above and presented algorithms for coalition formation and payoff distribution in general environments. We focused on a low complexity Kernel-oriented [12] coalition formation algorithm. The properties of this algorithm were examined via simulations. These have shown that the model increases the benefits of the agents within a reasonable time period, and more coalition formations provide more benefits to the agents.

3. Applying classical mechanics to large scale agent systems

There are situations where cooperation among a large number of agents (hundred or more) is needed. For example, the World Wide Web (WWW) consists of millions of users and is still growing. Another example is the employment of hundreds of simple, inexpensive autonomous mobile devices to achieve military and civilian goals in ground, air, and underwater environments [22]. In such situations [31,73], the agents work together toward satisfying a large set of joint goals, and the designers of the agents can agree in advance on regulations and protocols for the agents' interaction.

The negotiation and coalition formation methods presented in the previous section are suitable for environments with a relatively small number of agents. But, in very large agent-communities, these negotiation methods are typically too computationally complex and time-consuming. Furthermore, with hundreds of agents, direct communication connections between all of the agents may be impossible or too costly to establish.

Physical models of particle-dynamics have proved useful in such settings. They use mathematical formulation either to describe or to predict the properties and evolution of different states of matter. In particular, we developed efficient techniques for cooperation among hundreds of agents by adopting methods of classical mechanics used by physicists to tackle the problem of finding the properties of interaction among many particles. Although there are many differences between particles and computational systems, we have shown that the classical mechanics approach yields a model that enables feasible cooperation in very large agent-systems; the approach has a low computational complexity, which is crucial for the functioning of such systems. We have applied the classical mechanics-based methods to the following freight transportation example [16,63,75].

Example 2 (*Freight transportation system*). The system of freight transportation consists of many carriers (e.g., messengers on motorcycles) which belong to the same company, operating in a big city. Each carrier has a freight carrying capability that is given in units of volume and has a given location. The tasks that the carriers must fulfill are freight transportation tasks. We deal here with freight (e.g., packages) that should be moved from various locations to other locations. There are many freight transportation tasks to perform, and the carriers would like to perform them as soon as possible, while at the same time minimizing the company's expenses.

In the above example and in the other DAI environments that we consider, there is a large set of agents and a large set of goals they need to satisfy. Each agent has capabilities and should move toward satisfying goals. The first step in applying the classical mechanics model to DAI is the match between particles and their properties, agents and their capabilities, and goals and their properties. The next step is to identify the state of matter for modeling a community of agents and goals. The mathematical formulation that is used by physicists either to describe or to predict the properties and evolution of particles in these states of matter serve as the basis for the development of algorithms for the agents. However, several modifications of the classical mechanics model are necessary to provide an efficient algorithm for automated agents.

In the physical world, mutual attraction between particles causes motion. The reaction of a particle to the field of potential will yield a change in its coordinates and energies. The change in the state of the particle is a result of the influence of the potential. For DAI, the agents calculate the attraction and move according to the results of these calculations. That means, in our model, that each agent calculates the effect of the potential field on itself by solving a set of differential equations. According to the results of these calculations, it moves to a new state in the goal-domain. If it reaches a goal, it will proceed to a goal-satisfaction process. In cases where too many agents fit the requirements of the same goal, some are prevented from reaching the goal, through the property of mutual rejection between dynamic particles. We model the goal-satisfaction process by a collision of dynamic particles with static particles. Because the properties of particle collisions are different from the properties of goal-satisfaction, several adjustments were made to develop efficient algorithms for agent systems.

For example, in the freight transportation system of Example 2 each piece of freight is modeled by a static particle and each carrier is modeled by a dynamic particle, since carriers move toward the task's location. The volume of carriers' freight carrying capabilities and the volume of each piece of freight are modeled by particle masses, and their locations by particle locations.

The interaction between a carrier and a piece of freight is modeled by the mutual potential function of the modeling particles. It is calculated with respect to the distance between them. The potential functions derivatives yield forces which act on a dynamic particle and direct it. That is, the advancement towards a piece of freight is modeled by the movement of a dynamic particle towards a static particle. Repulsion between two dynamic particles which model two different carriers will influence the freight-task distribution among the carriers and will prevent two carriers from proceeding to a piece of freight which can be moved by one carrier. The performance of a freight-transportation task is modeled by the collision between a static particle, which models the task, and a dynamic particle, which models the agent.

In [70], we provide a detailed algorithm to be used by a single agent within the system. The algorithm leads to agent-goal allocation, and it converges to a solution where the fulfillment of goals is accomplished either by single agents or by groups of agents via cooperation. The computational complexity is low, and no explicit communication is necessary. In addition to these properties, we have proven that the algorithm we provide performs relatively close to the optimum.

The physics approach has several advantages. While common DAI algorithms must be checked for their validity either by a formal proof or by simulations, the models that are based on physics techniques can rely on theoretical and experimental results that are already known from physics. According to these results, one can predict the evolution of the modeled agent-system, since it will evolve in the same manner as a corresponding physical system. The local interactions, which enable one to derive the global behavior of the system, assure a low computational complexity of the model. In very large-scale agent-systems, this approach provides a model that promises emergent cooperative goal-satisfaction activity. In addition, the properties of the system as a whole can be analyzed, using concepts from statistical mechanics. The employment of such concepts enables us to derive the properties of a system through the properties of its components.

4. Applying operations research techniques

Many DAI researchers have considered situations of cooperative automated agents: for example, several workstations working together on fulfilling tasks [48], multi-agent for integration of design, manufacturing and shop floor control activities [4], and cooperative shipping companies [18]. In such situations, all the agents work together toward the satisfaction of a joint goal; the designer of the automated agents can develop, in advance, protocols for cooperation between the agents; the number of agents is not large; and the agents can communicate and have computation capabilities.

We recommend, in such situations, the consideration of operations research techniques. Researchers in operations research seek to determine how best to design and operate an organizational system, usually under conditions of scarce resources [76].

Autonomous agents working in DPS environments can be considered as an organizational system, and thus algorithms that were developed for human organizations in operations research may be applied to DAI environments. This is suitable for environments with a few dozen agents with large computation capabilities, because the computational complexity of the operations research techniques is usually high, and their efficiency decreases with the size of the organization to which they are applied.

We have applied operations research techniques which were developed for the set covering and set partitioning problems for coalition formation in DPS environments [69,71]. Given a set of agents and a set of tasks which they have to satisfy, we consider situations where each task should be attached to a group of agents which will perform the task. An example is a transportation company, similar to the example in the previous section. The company supplies transportation services via a number of trucks, lift trucks, cranes, boats, and planes. The drivers belong to a cooperative and share the benefits equally, and thus try to maximize the overall benefits of the company. There may be occasions in which one vehicle cannot perform a given transportation task by itself. In such cases, cooperation is necessary. Therefore, several drivers will form groups, and each group will fulfill a transportation task cooperatively. If the transportation company has many drivers, a distributed task allocation mechanism may be advantageous.

As we mentioned above, task allocation among agents may be approached as a problem of assigning groups of agents to tasks, and, therefore, the partition of the agents into subgroups becomes the main issue, and our problem becomes similar to the Set Partitioning Problem (SPP). Set partitioning entails the partition of a set into subsets, and the set partitioning problem is finding such a partition that has a minimal cost.[10] The SPP has been dealt with widely in the context of NP-hard problems [23], and approximation algorithms were developed in operations research [2,3,9,10,24]. Among them we can find the algorithm of Chvatal [10], which has a logarithmic ratio bound.[11]

The details of the algorithm that we developed, which is based on the operations research methods for the SPP is specified in [69]. Although the general task allocation problem is computationally exponential, the algorithm above is polynomial and yields results which are close to the optimal results and bounded by a logarithmic ratio bound. Another advantage of the algorithm, which is crucial in the case of a distributed system, is the distribution of the algorithm. We distribute the calculations in a natural way. That is, the distribution is an outcome of the algorithm characteristics, since each agent performs mostly those calculations that are required for its own actions during the process. In addition, our distribution method prevents most of the possibly overlapping calculations, thus saving unnecessary computational operations.

The algorithm is an anytime algorithm. If halted before normal termination, it still provides the system with several coalitions that have already formed. Since the first coalitions to be formed are the better ones, the results, when halted, are still of good quality. The anytime property of such an algorithm is important for dynamic environments, wherein the time-period for negotiation and coalition formation processes may be changed during the process.

In another paper [55], we considered the problem of distributed dynamic task allocation by a set of cooperative agents. We modeled the agents, using a stochastic closed queueing network, which is a well known operations research technique.

In both cases, we have developed polynomial algorithms that provide near optimal results. From our experience, we realized that in order to apply operations research techniques to DAI, there are several steps that must be taken. First, there is the need to find a problem that was considered in operations research which is close to the DAI problem and to make a detailed match between the problems. For example, in the coalition formation problem described above, we realized that it is close to the SPP or SCP problems. Then, there is the need to adjust the operations research algorithm to the DAI environment. In particular, most of the operations research algorithms are centralized, and, since we deal with autonomous agents, we seek distributed algorithms. In addition, there is the need to develop utility functions that can be used by the agents. In operations research it is assumed that cost function is provided as part of

[10] Coalition formation where coalitions may overlap can be approached as a Set Covering Problem (SCP).

[11] An approximation algorithm for a problem has a ratio bound $\rho(n)$ if $\rho(n)$ is smaller than the ratio between the optimal cost and the approximated cost.

the problem (as in game theory). In our model, we need to provide the agents with efficient techniques to calculate them (see also [64]). For example, in [69, 71] we had to develop the cost function and coalitional values in the context of task allocation and to provide a distributed algorithm to compute them. This notion of coalitional value is different from the notion of game-theoretic coalitional value, since here the value depends on the coalitional configuration and on the task allocation.

Although adjusting the operations research techniques to DAI situations required some effort, we determined that the benefits from using these well-developed methods, and techniques for evaluating them, may help in reaching efficient algorithms for the DAI environment.

5. The application of informal models of behavioral and social sciences to automated agents

There are situations where automated agents need to interact with other agents in non-structured environments; for example, an information server which works to form a multi-media document for answering a complex query of a user, agents that help train people in negotiation [44], and agents that sell goods on the World Wide Web [8]. In such situations, the agents are self-motivated, and usually the automated agents need to interact with people. The number of agents in the environment is not large, and communication is possible.

In such situations, we found that formalizing and implementing informal models of behavioral and social sciences can be beneficial. Behavioral and social sciences study human cooperation and coordination and develop frameworks and models of organizations and communities (e.g., [20,46,59,60]). In non-structured and unpredictable environments, heuristics for cooperation and coordination among automated agents, based on successful human cooperation and interaction techniques, may be useful.

We have applied informal models to different types of environments, and we will discuss one of them below. Applying informal models to DAI can be done in two ways:

(a) using the informal models as motivation for the development of heuristics for the cooperative activities of the automated agents;

(b) formalizing the informal models (e.g., using logic) and then applying them to a DAI environment.

In both cases, there is a need to carry out simulations in order to evaluate the performance of the techniques, since the informal models usually do not formally analyze the behavior of the systems. The main advantage in using these models is that we build upon experience and expertise that were developed over the years in the specific type of interactions, rather than starting from scratch and using only our own experience. Our success in the developments of specific applications, in particular automated negotiators [41,42], supports this claim.

There are two main approaches in the social sciences to the development of theorems relating to negotiation. The first approach, which we used in Section 2.1, is the formal theory of bargaining. This formal game-theoretic approach provides clear analyses of various situations and precise results concerning the strategy a negotiator should choose.

However, it requires making restrictive assumptions, and the agents need to follow strict negotiation protocols which are not possible in some real world environments.

The second approach, which we refer to as the negotiation guides approach, comprises informal theories which attempt to identify possible strategies for a negotiator and to assist him in achieving good results (see, for example, [13,19,29,33,35]). These negotiation guides do not accept the strong restrictions and assumptions presented in the game-theoretic models. Applying these methods to DAI is more difficult than using the first approach, since there is no formal theory nor strategies that can be used. However, these methods can be used in domains where people interact with each other and with automated agents, and situations where automated agents interact in environments without pre defined regulations. These informal models can serve as guides for the development of negotiation heuristics [41] or as a basis for the development of a logical model of negotiation [42].

In [37,41], we developed a general structure for a self-motivated *Negotiating Automated Agent* acting in environments where cooperation between the agents may be beneficial, but where conflicts among the agents can arise. There are no strict regulations and protocols for the negotiation, there is no mediator, and central controllers do not exist. Thus agreements are *not* enforced, and agents may break their promises. The agents have incomplete information concerning the other agents' goals and tasks, and an agent can provide the other agents with false information.

As a testbed, a specific domain was chosen, the *Diplomacy* game, which is rich enough to include most aspects of negotiation.[12] Given a (restricted version of) natural language which covers this domain, our agent, Diplomat, was confronted with human agents and even demonstrated an advantage over its human negotiation partners.

The framework of Diplomat consists of five modules: the *Prime Minister*, that directs the Diplomat's activities; the *Ministry of Defense*, that is responsible for the planning; the *Foreign Office*, that negotiates with the other players; the *Headquarters* that executes the basic tasks of Diplomat; and the *Intelligence Agency*, that is responsible for collecting information about the environment and the other players. These modules are implemented by a dynamic set of local agents that work together, communicate, and exchange messages to achieve the common general tasks of Diplomat.

In the design of Diplomat and in choosing the negotiation heuristics it uses, we used different general informal negotiation guides. For example, as we mentioned above, Diplomat consists of different modules for planning—i.e., the Ministry of Defense—and negotiations—i.e., the Foreign Office. The development of different modules for negotiation and planning is a characteristic of a good negotiator, according to Fisher and Ury's model [19]. They suggest that a good negotiator should do much "inventing", that is, find out new ideas that are not already among the negotiation issues. The separation of the planning and negotiation into two modules enables the Ministry of Defense to find as many solutions to the problem as possible, without taking into account whether or not they are acceptable to the other side. The ideas will not be conveyed to the other

[12] Diplomacy is a board game marketed by Avalon Hill Company and played on the map of Europe during the years just prior to World War I. Coalitions and agreements among the players significantly affect the course of the game.

side until the Foreign Office decides to do so. Therefore, their consideration by the Ministry of Defense can do no harm.

There are several heuristics that Diplomat uses to decide how to make suggestions to another agent. For example, when considering a cooperation agreement with another agent, Diplomat designs several possible strategies and compares them to choose the strategy that will be a basis for the agreement. Since a negotiator wants to "win", one may suspect that the only criterion that will guide him while comparing and choosing between strategies will be his own benefits derived from the strategies. However, as has been suggested by the literature on human negotiation, this is not the case. The reason for that phenomena is that in order for the agreement to last, it should be beneficial to *all* parties involved. Otherwise, a neglected partner may be tempted to reach a more appealing agreement, even without informing the negotiator. For that same reason, the other partner should be convinced that the agreement is profitable to Diplomat (see [19]); otherwise he will suspect that the negotiator will later break the agreement.

In order to test Diplomat, we arranged several Diplomacy games, and our findings (see [41]) show that Diplomat played well in the games in which it participated. We believe that its success is due to the integration of the heuristic techniques we developed for the construction of negotiator agents and well developed informal theories of negotiation.[13]

6. Conclusions

In this paper we argue that applying multi-entity techniques, such as game theory and physics, to DAI, is beneficial. We described several attempts to apply methodologies from diverse fields to DAI problems. A summary of the multi-entity techniques that we used and their application in DAI is given in Table 1. The last column uses the parameters presented in the introduction to characterize the problems that we considered. For example, we applied game theory in environments where the agents are automated and self-motivated, but it is possible that the agents will follow some agreed-upon protocols (Sections 2.1 and 2.2). We demonstrated that classical mechanics models are useful for task distribution in very large sets of cooperative agents (Section 3). We applied operations research techniques such as queueing networks for task distribution among a relatively small set of cooperative agents (Section 4). We used the less formal social science models of cooperation when there were no strict protocols for the cooperation (Section 5), or when communication was not possible [15,43]. Further, we demonstrated that ideas drawn from philosophy can be the basis for the development of SharedPlans among agents [26,27].

[13] We have applied other informal models to DAI situations. In [42], we developed a formal logic that forms a basis for the development of a formal axiomatization system and the implementation of a logic-based negotiator [14] based on persuasion models [1]. In [26,27], we have applied philosophical informal models of cooperative activity [7] for situations where teams composed of people and computers plan and work together toward satisfying a shared goal. In [15,43], we used the notion of focal point introduced by Schelling [58,66], for multi-agent cooperation without communication.

Table 1
Summary of multi-entity techniques and their application in DAI. In the last column, SMA stands for self-motivated agents, and CA indicates cooperative agents which work toward satisfying the same goal (see Section 1). R&P indicates that the designers can agree on regulations and protocols for agents interaction. s#, m# and l# stands for environments with small (handful), medium (few dozen), or large number (hundreds) of agents, respectively. AUTO indicates environments with only automated agents, and AUTO&PE stands for systems composed of people and automated agents. That communication is possible is indicated by COMU.

Multi-entity techniques	DAI	Papers	Characterization
Game theory			
Strategic bargaining models	Negotiation for task distribution & Resource allocation in MA	[44,45] [39]	SMA, s#, R&P, AUTO, COMU
Theories of coalition formation	Coalition formation in MA	[68,72]	SMA, m#, R&P, AUTO, COMU
Principle-agent models	Contracting tasks in MA	[38,40]	SMA, s#, R&P, AUTO, COMU
Physics			
Classical mechanics	Goal satisfaction in very large DPS environments	[70]	CA, l#, R&P, AUTO
Operations research			
SPP & SCP	Coalition formation in DPS	[69,71]	CA, m#, R&P, AUTO, COMU
Queueing networks	Task allocation in DPS	[55]	CA, m#, R&P, AUTO, COMU
Behavioral sciences			
Negotiation guides	Diplomatic negotiation	[37,41]	SMA, m#, AUTO&PE, COMU
Persuasion models (logic)	Argumentation	[14,42]	SMA, s#, AUTO&PE, COMU
Focal points (logic & decision theory)	Cooperation without communication	[15,43]	CA, m#, AUTO, R&P
Philosophy (logic)	Collaborative plans	[26,27]	CA&SMA, m#, AUTO&PE, COMU

There are two main aspects of a multi-entity environment that determine its usefulness to a DAI problem and its effect on the amount of work required for the adaptation of techniques developed for it to the DAI problems. The first criterion is the similarity between the entities and the automated agents. The second criterion is the level of formalization that is used by researchers of the multi-entity domains.

For example, people are more similar to automated agents than are particles. Therefore, in all the multi-entity techniques that were developed for humans environments, it was not difficult to match the entities in the environment and the participants in the multi-agent domains. For example, it is clear that players in game-theoretic frameworks can model automated agents. It is less clear which types of particles in the classical mechanics framework serve as models for agents and that collisions are a good way to model goal-satisfaction.

The second criterion has to do with the fact that we need to provide our automated agents with formal and well-designed algorithms. With respect to this, it is easier to use techniques from formal multi-entity models than techniques that were not formalized by their developers. For example, even though people and automated agents have much in common, with respect to cooperation, it is quite difficult to develop an algorithm for agent cooperation based on the informal ideas, procedures, and rules that are presented by social scientists and philosophers. Much effort is required to formalize these procedures and rules and to produce an implementable algorithm for the automated agents. On the other hand, after going through the process of modeling a community of agents using a classical mechanics framework, the usage of the formal techniques of classical mechanics is not so difficult. There is a need to modify the formal procedures and to adjust them to the multi-agent requirement, but there is no need to create the formal procedure from scratch.

Acknowledgments

I would like to thank the many people who, over the years, have collaborated with me: C. Baral, E. Blake, P. Bonatti, E. Ephrati, A. Evenchik, D. Etherington, M. Fenster, B. Grosz, M. Harris, J. Hendler, K. Holley, J. Horty, D. Lehmann, G. Lemel, M. Magidor, J. Minker, M. Nirkh, D. Perlis, T. Plotkin, J. Rosenschein, A. Schwartz, O. Shehory, Y. Shoham, S. Subrahmanian, K. Sycara, B. Thomas, J. Wilkenfeld, and G. Zlotkin. Our joint work influenced my thinking on cooperation and coordination.

I would like to thank Barbara Grosz, Martha Pollack, Jonathan Wilkenfeld, Onn Shehory and Orna Schechter, each of whom also provided help and support while I was preparing the Computers and Thought lecture and this paper. Special thanks to Dr. Shifra Hochberg for editorial assistance.

This work was supported by the NSF under Grants No. IRI-9423967 and IRI-9311988 and the Israeli Ministry of Science, Grants No. 6288 and 4210.

References

[1] H. Abelson, *Persuasion* (Springer, New York, 1959).

[2] E. Balas and M. Padberg, On the set covering problem, *Oper. Res.* 20 (1972) 1152-1161.

[3] E. Balas and M. Padberg, On the set covering problem: an algorithm for set partitioning, *Oper. Res.* 23 (1975) 74-90.

[4] S. Balasubramanian and D. Norrie, A multi-agent intelligent design system integrating manufacturing and shop-floor control, in: *Proceedings 1st International Conference on Multiagent Systems* (1995) 3-19.

[5] T. Balch and R.C. Arkin, Motor schema-based formation control for multiagent robot teams, in: *Proceedings 1st International Conference on Multiagent Systems* (1995) 10-16.

[6] A.H. Bond and L. Gasser, An analysis of problems and research in DAI, in: A.H. Bond and L. Gasser, eds., *Readings in Distributed Artificial Intelligence* (Morgan Kaufmann, San Mateo, CA, 1988) 3-35.

[7] M.E. Bratman, Shared cooperative activity, *Philosophical Review* 101 (1992) 327-341.

[8] A. Chavez and P. Maes, Kasbah: an agent marketplace for buying and selling goods, in: *Proceedings 1st International Conference on the Practical Application of Intelligent Agents and Multi Agents Technology*, London (1996) 75-90.

[9] N. Christofides and S. Korman, A computational survey of methods for the set covering problem, *Math. Oper. Res.* 21 (1975) 591-599.

[10] V. Chvatal, A greedy heuristic for the set-covering problem, *Math. Oper. Res.* 4 (1979) 233-235.

[11] M.S.Y. Chwe, Farsighted coalitional stability, *J. Economic Theory* 63 (1994) 299-325.

[12] M. Davis and M. Maschler, The kernel of a cooperative game, *Naval Res. Logist. Quart.* 12 (1965) 223-259.

[13] D. Druckman, *Negotiations* (Sage, Beverly Hills, CA, 1977).

[14] A. Evenchik, Inference system for argumentation in negotiation between automatic agents, M.Sc. Thesis, Department of Mathematics and Computer Science, Bar-Ilan University, Ramat Gan, Israel (1995).

[15] M. Fenster, S. Kraus and J. Rosenschein, Coordination without communication: experimental validation of focal point techniques, in: *Proceedings 1st International Conference on Multiagent Systems* (1995) 102-116.

[16] K. Fischer and N. Kuhn, A DAI approach to modeling the transportation domain, Technical Report RR 93-25, Deutsches Forschungszentrum für Künstliche Intelligenz GMBh (1993).

[17] K. Fischer, J.P. Müller, I. Heimig and A. Scheer, Intelligent agents in virtual enterprises, in: *Proceedings 1st International Conference on the Practical Application of Intelligent Agents and Multi Agents Technology*, London, 1996.

[18] K. Fischer, J.P. Müller, M. Pischel and D. Schier, A model for cooperative transportation scheduling, in: *Proceedings 1st International Conference on Multiagent Systems* (1995) 109-116.

[19] R. Fisher and W. Ury, *Getting to Yes: Negotiating Agreement without Giving in* (Houghton Mifflin, Boston, MA, 1981).

[20] R.C. Ford, R.B. Armandi and C.P. Heaton, *Organization Theory: An Integrative Approach* (Harper and Row, New York, 1988).

[21] D. Fudenberg and J. Tirole, *Game Theory* (MIT Press, Cambridge, MA, 1991).

[22] D.W. Gage, Command control for many-robot systems, *Unmanned Systems* (Fall, 1992) 28-34.

[23] M.R. Garey and D.S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness* (Freeman, New York, 1979.

[24] R.S. Garfinkel and G.L. Nemhouser, The set-partitioning problem: set covering with equality constraints, *Oper. Res.* 17 (1969) 848-856.

[25] L. Glicoe, R. Staats and M. Huhns, A multi-agent environment for department of defense distribution, in: *Proceedings IJCAI-95 Workshop on Intelligent Systems*, Montreal, Que. (1995).

[26] B. Grosz and S. Kraus, Collaborative plans for group activities, in: *Proceedings IJCAI-93*, Chambery, France (1993) 367-373.

[27] B.J. Grosz and S. Kraus, Collaborative plans for complex group activities, *Artificial Intelligence* 86 (1996) 269-357.

[28] S. Guiasu and M. Malitza, *Coalition and Connection in Games* (Pergamon, Oxford, 1980).

[29] Lavinia Hall, ed., *Negotiation: Strategies for Mutual Gain* (Sage, Beverly Hills, CA, 1993).

[30] J.C. Harsanyi, *Rational Behavior and Bargaining Equilibrium in Games and Social Situations* (Cambridge University Press, Cambridge, 1977).

[31] T. Hogg, Social dilemmas in computational ecosystems, in: C.S. Mellish, ed., *Proceedings IJCAI-95*, Montreal, Que. (Morgan Kaufmann, San Mateo, CA, 1995) 711-718.

[32] N.R. Jennings, Controlling cooperative problem solving in industrial multi-agent systems using joint intentions, *Artificial Intelligence* 75 (1995) 1-46.

[33] R. Johnson, *Negotiation Basics* (Sage, Beverly Hills, CA, 1993).

[34] J.P. Kahan and A. Rapoport, *Theories of Coalition Formation* (Lawrence Erlbaum, Hillsdale, NJ, 1984).

[35] C.L Karrass, *The Negotiating Game: How to Get What You Want* (Thomas Crowell Company, New York, 1970).

[36] M. Klusch and O. Shehory, A polynomial kernel-oriented coalition formation algorithm for rational information agents, in: *Proceedings ICMAS-96*, Kyoto, Japan (1996).

[37] S. Kraus, Planning and communication in a multi-agent environment, Ph.D. Thesis, Hebrew University, Jerusalem, 1988 (written largely in Hebrew).

[38] S. Kraus, Agents contracting tasks in non-collaborative environments, in: *Proceedings AAAI-93*, Washington, DC (1993) 243-248.

[39] S. Kraus, Beliefs, time and incomplete information in multiple encounter negotiations among autonomous agents, *Ann. Math. Artif. Intell.* (1997).

[40] S. Kraus, An overview of incentive contracting, *Artificial Intelligence* 83 (1996) 297-346.

[41] S. Kraus and D. Lehmann, Designing and building a negotiating automated agent, *Comput. Intell.* 11 (1995) 132-171.

[42] S. Kraus, N. Nirkhe and K.P. Sycara, Reaching agreements through argumentation: a logical model, in: *Proceedings DAI-93* (1993).

[43] S. Kraus and J.S. Rosenschein, The role of representation in interaction: discovering focal points among alternative solutions, in: E. Werner and Y. Demazeau, eds., *Decentralized Artificial Intelligence*, Vol. 3 (Elsevier, Amsterdam, 1992) 147-165.

[44] S. Kraus and J. Wilkenfeld, A strategic negotiations model with applications to an international crisis, *IEEE Trans. Systems Man Cybernet.* 23 (1993) 313—323.

[45] S. Kraus, J. Wilkenfeld and G. Zlotkin, Multiagent negotiation under time constraints, *Artificial Intelligence* 75 (1995) 297-345.

[46] W.A. Kraus, *Collaboration in Organizations: Alternatives to Hierarchy* (Human Sciences Press, New York, 1984).

[47] R.D. Luce and H. Raiffa, *Games and Decisions* (John Wiley and Sons, New York, 1957).

[48] T.W. Malone, R.E. Fikes, K.R. Grant and M.T. Howard, Enterprise: a marketlike task schedule for distributed computing environments, in: B.A. Huberman, ed., *The Ecology of Computation* (North-Holland, Amsterdam, 1988) 177-205.

[49] R. Myerson, *Game Theory* (Harvard University, 1991).

[50] NASA, EOSDIS home page, http://www-v0ims.gsfc.nasa.gov/v0ims/index.html, 1996.

[51] J. von Neumann and O. Morgenstern, *Theory of Games and Economic Behavior* (Princeton University Press, Princeton, NJ, 1947).

[52] M.J. Osborne and A. Rubinstein, *Bargaining and Markets* (Academic Press, San Diego, CA, 1990).

[53] M.J. Osborne and A. Rubinstein, *A Course in Game Theory* (MIT Press, Cambridge, MA, 1994).

[54] C. Petrie, M. Cutlosky and H. Park, Design space navigation as a collaborative aid, in: *Proceedings 3rd International Conference on Artificial Intelligence in Design*, Lausanne (1994).

[55] T. Plotkin and S. Kraus, A queueing network approach to distributed task allocation for cooperative agents (submitted, 1996).

[56] M. Pollack, T. Znati, E. Ephrati, D. Joslin, S. Lauzac, A. Nunes, N. Onder, Y. Ronen and S. Ur, The dipart project, in: *Proceedings ARPI Annual Meeting*, Tucson, AZ (1994).

[57] A. Rapoport, *N-Person Game Theory* (University of Michigan, Ann Arbor, MI, 1970).

[58] E. Rasmusen, *Games and Information* (Basil Blackwell Ltd., Cambridge, MA, 1989).

[59] M.I. Reed, *The Sociology of Organisations: Themes, Perspectives and Prospects* (Harvester Wheatsheaf, New York, 1992).

[60] S.P. Robbins, *Organization Theory* (Prentice Hall, Englewood Cliffs, NJ, 1990).

[61] J.S. Rosenschein and G. Zlotkin, *Rules of Encounter: Designing Conventions for Automated Negotiation Among Computers* (MIT Press, Boston, MA, 1994).

[62] A. Rubinstein, Perfect equilibrium in a bargaining model, *Econometrica* 50 (1982) 97-109.

[63] T. Sandholm, An implementation of the contract net protocol based on marginal cost calculations, in: *Proceedings AAAI-93*, Washington, DC (1993) 256-262.

[64] T.W. Sandholm and V.R. Lesser, Coalitions among computationally bounded agents, *Artificial Intelligence* 94 (1997) 99-137 (this issue).

[65] O. Schechter, Sharing resources through negotiation in multi-agent environments, Master's Thesis, Bar-Ilan University, Ramat-Gan, Israel (1996).

[66] T.C. Schelling, *The Strategy of Conflict* (Oxford University Press, New York, 1963).

[67] S. Sen and E.H. Durfee, A formal study of distributed meeting scheduling, in: *Group Decision and Negotiation Support Systems* (1997).

[68] O. Shehory and S. Kraus, Coalition formation among autonomous agents: strategies and complexity, in: *From Reactions to Cognition* (Springer, Berlin, 1995) 57-72.

[69] O. Shehory and S. Kraus, Task allocation via coalition formation among autonomous agents, in: *Proceedings IJCAI-95*, Montreal, Que. (1995) 655-661.

[70] O. Shehory and S. Kraus, Emergent cooperative goal-satisfaction in large scale automated-agent systems, in: *Proceedings ECAI-96*, Budapest (1996) 544–548.

[71] O. Shehory and S. Kraus, Formation of overlapping coalitions for precedence-ordered task-execution among autonomous agents, in: *Proceedings ICMAS-96* (1996) 330–337.

[72] O. Shehory and S. Kraus, A kernel-oriented model for coalition-formation in general environments: Implementation and results, in: *Proceedings AAAI-96*, Portland, OR (1996) 134–140.

[73] Y. Shoham and M. Tennenholtz, On the emergence of social conventions: modeling, analysis and simulations, *Artificial Intelligence* 94 (1997) 139–166 (this issue).

[74] M.B. Tsvetovatyy and M. Gini, Toward a virtual marketplace: architectures and strategies, in: *Proceedings 1st International Conference on the Practical Application of Intelligent Agents and Multi Agents Technology*, London (1996) 597–614.

[75] M.P. Wellman, A market-oriented programming environment and its application to distributed multicommodity flow problems, *J. Artif. Intell. Res.* 1 (1993) 1–23.

[76] W.L. Winston, *Operations Research: Applications and Algorithms* (PWS-Kent Publishing Company, Boston, MA, 1987).