Mechanism Design for Communication in Cooperative Systems

Claudia V. Goldman
Department of Computer Science
University of Massachusetts
Amherst, MA 01003
clag@cs.umass.edu

Shlomo Zilberstein
Department of Computer Science
University of Massachusetts
Amherst, MA 01003
shlomo@cs.umass.edu

ABSTRACT

Distributed systems are characterized by having partial observability of the global state during execution. Nevertheless, when these systems comprise cooperative agents, they should attain global objectives. Planning for these decentralized systems is a very complex task. Exchange of local information through communication can alleviate this complexity by allowing the agents to be synchronized from time to time. Due to costs associated with real-world communication, agents may not be able to continuously obtain full observability of the system. We examine mechanisms that result in the decomposition of the global problem into local simpler problems that are applied each time the agents exchange information. The communication policies are computed with respect to a given mechanism and policy of action. This paper presents a framework to study these mechanisms and evaluation criteria to compare them. We also review related work on mechanism design and compare the approaches.

1. INTRODUCTION

Decentralized control is essential when a group of collaborating decision-makers can perform disjoint actions, with each having only partial information about the global state. Of particular interest are activities in which global objectives should be attained without the possibility of continuous communication. We are interested in this type of decentralized cooperative multi-agent systems (MAS) composed of multiple, resource-bounded decision-makers that share a common set of objectives. In principle, these agents are able to share their knowledge, although communication has a cost associated due to the risk in revealing the information to competing agents, or due to the bandwidth necessary to transmit the information or due to the complexity of computing the information to be transmitted. Therefore, it may be impossible or undesirable for these decision-makers to share all their knowledge all the time.

Game Theoretic and Decision Theoretic Agents Workshop at AAMAS'03, July 14–18, 2003, Melbourne, Australia.
.

In decentralized cooperative MAS, agents are indeed able to freely share information at the off-line planning stage due to the cooperative characteristic of the system. However, during the on-line stage of execution these agents will be acting in a decentralized manner, where full observability of the whole system state cannot be assumed, and where communication of information (e.g., regarding each agent's local state) incurs some cost. Controlling decentralized cooperative MAS includes considering these behaviors while planning off-line.

Controlling multi-agent systems that arrive at coordinated and cooperative behaviors has been extensively studied in the distributed artificial intelligence literature. We take a formal approach which is mainly concerned with optimal policies of action and of communication. It is different from centralized approaches taken to solve problems in multi-agent systems where both the off-line planning stage and the on-line stage are controlled by a central entity, or by all the agents in the system, who are assumed to have full observability (e.g., MMDPs [3]). It is also different from the distributed cooperative MAS that were studied at the other extreme, i.e., both off-line planning and on-line execution are done in a completely distributed manner (e.g., [22, 10, 18]). Examples of non-cooperative distributed systems were studied by Littman [16] and by Hu and Wellman [14].

The research done so far has assumed a known and fixed language of communication when communication was relevant [6, 13]. For example, a known standard for agents communication is the Knowledge and Query Manipulation Language (KQML [7]) where the messages format and the message-handling protocol are determined at the system design stage. Two exceptions are the work done by Gmytrasiewicz et al. [9] and by Wang and Gasser [28].

Gmytrasiewicz et al. [9] study the evolution of a communication language between knowledge-based and rational agents through a process of negotiation. Their assumption is that the agents have conflicts of interest because each agent would prefer a communication language that is easier and less expensive to use from its individual perspective. A failure in translating a message from the agent's own representation language to the agent communication language is a signal to start a negotiation with the other agent. The negotiation process assumes that agents can point at objects to negotiate over labels to these objects or over relations between them (expressed by the set of objects that follow that relation). The agents negotiate over the cost of implementing these new knowledge structures and the expected values of exchanging this information.

Wang and Gasser [28] study how agents collectively learn a single concept. They show the convergence of a perceptron-algorithm adapted to mutual learning of a concept by multiple agents. This single-learned concept does not have any semantics and its learning was not related to any planning process. We are interested in the exchange of messages that have a certain meaning and affect the agents' plans.

The formal framework underlying the study of communication and decentralized control is based on decentralized partially-observable Markov Decision Process (Dec-POMDP) [2]. We extended this Dec-POMDP framework to include a language of communication with the corresponding cost. Following the complexity analyzes done by Bernstein et al. [2] and by Pynadath and Tambe [23], the worst case complexity is in the NEXP class. Assuming that the agents can communicate and fully synchronize their partial views of the global state at free cost, transforms the Dec-POMDP problem into an MDP scenario which is known to be tractable. We are interested in computing optimal policies of action *and* optimal policies of communication for a decentralized controlled cooperative MAS. The framework in which we study decentralized control is described in Section 2. Our recent research [12] has taken a meta-level approach to communication in the framework of decentralized control. In order to reduce complexity, we have assumed that a mechanism can be applied on the global problem to decompose it into simpler local problems which can be solved independently by each agent. We also assume that agents exchange information in order to achieve cooperation. In this paper, we study mechanism design for communication in collaborative multi-agent systems. How does the information obtained through communication affect each agent's policy of action and near term behavior? This paper provides a formal framework to study these mechanisms, as well as criteria to evaluate them. In Section 3, we present a formal analysis of the mechanisms we propose for communication in cooperative multi-agent systems. Mechanism design is a known research area in game theory and Economics where a center designs the rules of a game for self-interested agents [21]. Recently, social-laws [25, 11] and negotiation mechanisms [24] were developed for multi-agent systems as mechanisms for social-coordination. These approaches are explained and compared in Section 4. We conclude in Section 5.

2. DECENTRALIZED CONTROL WITH COMMUNICATION

The formal framework in which we study the problem of decentralized control with communication is a decentralized partially-observable Markov Decision Process with communication, Dec-POMDP-Com, defined below. For simplicity of exposition, the framework is presented for two agents, but it can be extended to any number of agents.[1]

$M^{com} =< S, A_1, A_2, \Sigma, C_\Sigma, P, R, \Omega_1, \Omega_2, O, T >$ where:

- S is a finite set of global states. s_0 denotes the initial state of the system.
- A_1 and A_2 are finite sets of control actions.
- Σ is the alphabet of messages. $\sigma_i \in \Sigma$ denotes an atomic message sent by agent i. A special message that belongs to Σ is the null message, ϵ_σ. This message is sent by an agent that does not want to transmit anything to the other agents. The agents do not incur any cost in sending a null message.
- C_Σ is the cost of transmitting an atomic message. $C_\Sigma(\epsilon_\sigma) = 0$, $C_\Sigma : \Sigma \rightarrow \Re$.
- P is the transition probability function. $P(s, a_1, a_2, s')$ is the probability of moving from state s to state s' when agents 1 and 2 perform actions a_1 and a_2.
- R is the reward function. $R(s, a_1, \sigma_1, a_2, \sigma_2, s')$ represents the reward obtained by the system as a whole, when agent 1 executes action a_1 and sends message σ_1, and agent 2 executes action a_2 and sends message σ_2 in state s resulting in a transition to state s'.
- Ω_1 and Ω_2 are finite sets of observations.
- O is the observation function. $O(s, a_1, a_2, s', o_1, o_2)$ is the probability of observing o_1 and o_2 (respectively by the two agents) when in state s agent 1 takes action a_1 and agent 2 takes action a_2, resulting is state s'.
- T is a positive integer representing the horizon. In general, the Dec-POMDP-Com can be defined with an infinite horizon.

A Dec-POMDP-Com is *jointly fully-observable* if there exists a mapping $J : \Omega_1 \times \Omega_2 \rightarrow S$ such that whenever $O(s, a_1, a_2, s', o_1, o_2)$ is non-zero then $J(o_1, o_2) = s'$. In other words, joint full-observability means that the *combination* of the agents' partial views of the system state comprises the system state. Still, each agent has partial observability of the system state. We denote by Dec-MDP-Com a Dec-POMDP-Com with joint full-observability. A Dec-POMDP-Com is *jointly synchronized* if both agents have the same knowledge about the global state, and none of the agents separately has more knowledge than this. Communication is the only means to obtain information about other agents' observations.

We describe the interaction among the agents as a process in which agents perform an action, then they observe their environment, and then send a message that is instantaneously received by the other agent. We assume no delays in the system. Each agent's behavior is determined by its local policy δ composed of two policies: a policy of action δ^A, and a policy of communication δ^Σ.

DEFINITION 1. *A local policy for action for agent i, δ_i^A is a mapping from local histories of observations $\overline{o_i} = o_{i_1}, \ldots, o_{i_t}$ over Ω_i and histories of messages $\overline{\sigma_j} = \sigma_{j_1}, \ldots, \sigma_{j_t}$* received *($j \neq i$) since the last time the agents were synchronized to actions in A_i. $\delta_i^A : S \times \Omega^* \times \Sigma^* \rightarrow A_i$*

DEFINITION 2. *A local policy for communication for agent i, δ_i^Σ is a mapping from local histories of observations $\overline{o_i} = o_{i_1}, \ldots, o_{i_t}$ and o, the last observation perceived after performing the last local action, over Ω_i and histories of messages $\overline{\sigma_j} = \sigma_{j_1}, \ldots, \sigma_{j_t}$* received *($j \neq i$) since the last time the agents were synchronized to messages in Σ. $\delta_i^\Sigma : S \times \Omega^* o \times \Sigma^* \rightarrow \Sigma$*

DEFINITION 3. *A joint policy $\delta =< \delta_1, \delta_2 >$ is defined to be a pair of local policies, one for each agent, where each δ_i is composed of the communication and the action policy for agent i.*

[1]The model is similar to the COM-MTDP model[23], although Pynadath and Tambe did not present an algorithm for solving the problem of decentralized control with communication.

2.1 The Language of Communication

The model presented allows for a general setting of decentralized control, when the semantics or the syntax of the messages may not be shared by the agents and when the semantics are fixed or implicit. Learning to communicate is another separate line of research that we are looking at. The semantics are fixed when they are determined by the designer. They are implicit when they are given by the context in which a message is transmitted. The context of a message sent by agent j is a pair consisting of 1) the last synchronized state when the agents exchanged their information, and 2) the sequence of observations observed by agent i when agent j sent this message to him. A local policy defines implicitly the triplets of synchronized states, observation sequences and messages, and therefore the table of instantiations of the δ mapping gives us all the possible meanings for the language Σ.

The problem we are interested in here - how to design mechanisms for communication - is relevant for any type of messages that will be exchanged when agents communicate (e.g., informative, commitments, signals of reward or punishment and more complex world information). Even when agents do share the semantics and the syntax of messages they need to have conventions about how to interpret these messages and how to combine this information with their own local information to derive near-term policies of action.

In this paper, we assume that the language of communication is fixed ahead of time, and it is shared by all the agents. In particular we focus on *informative messages* which affect the decision of the hearer when he chooses its next action. The policy that instructs an agent to perform a control action, δ_i^A, is a function of the last synchronized state, the last observation o and all the r messages received so far from agent j. These messages affect directly the decision of agent i. Informative messages, on the other hand, do not affect the outcome of the action chosen by the hearer because the probability transition function, $P(s, a_1, a_2, s')$, depends only on the actions performed by the agents, and it does not depend on the messages transmitted. One example of a language with fixed semantics that is composed of informative messages is the language of observations (i.e., $\Sigma_i = \Omega_i$), where the agents communicate their observations.

3. MECHANISMS FOR COMMUNICATION

We are concerned with mechanism design for communication-based control of decentralized cooperative processes. The first algorithm that optimally solves the sub-class of transition-independent and factored decentralized control problems has been recently published by Becker et al. [1]. However, no algorithm is known that optimally solves both the decentralized control and communication problem where agents are allowed to exchange local information to attain a synchronized view of the system. Goldman and Zilberstein [12] developed the analytical expression for the value of a state in a Dec-POMDP-Com from following a joint policy δ for T steps. Then, an algorithm that solves the problem in question needs to find the optimal joint policy δ^* for action and for communication such that $\delta^* = argmax_\delta V_\delta^T(s_0)$, where s_0 is the initial global state of the Dec-POMDP-Com. The value function $V_\delta^T(s_0)$ is given as follows where: 1)f_j^i are the first i messages sent by agent j with length equal to i. 2) $\overline{P_\delta}(s, \overline{o_1}, \overline{\sigma_2}, \overline{o_2}, \overline{\sigma_1}, s')$ is the probability of transitioning from a state s to a state s' following the joint policy $\delta =< \delta_1, \delta_2 >$ while agent 1 sees observation sequence $\overline{o_1}$ and receives sequences of messages $\overline{\sigma_2}$, and agent 2 sees $\overline{o_2}$ and receives $\overline{\sigma_1}$ of the same length. 3) The observation and the message sequences are of length at most $T-1$. The actions depend on a sequence of observations of length i, and the messages depend on sequences of observations of length $i+1$ because the message is sent after an action was performed and the resulting observation was observed.

$$V_\delta^T(s) = \sum_{<\overline{o_1},\overline{o_2}>} \sum_{q\in S} \sum_{s'\in S} \overline{P_\delta}(s, \overline{o_1}, f_2^i, \overline{o_2}, f_1^i, q).$$
$$P(q, \delta_1^A(s, \overline{o_1}, f_2^i), \delta_2^A(s, \overline{o_2}, f_1^i), s').$$
$$R(q, \delta_1^A(s, \overline{o_1}, f_2^i), \delta_1^\Sigma(s, \overline{o_1}o_1, f_2^i), \delta_2^A(s, \overline{o_2}, f_1^i), \delta_2^\Sigma(s, \overline{o_2}o_2, f_1^i), s')$$

In order to reduce the complexity of solving the general problem, we propose to design mechanisms for decentralizing the control by allowing the agents to synchronize their local information from time to time through communication.[2] These mechanisms enable the agents to operate separately for certain periods of time. The policy of communication is designed at the meta-level of the Dec-POMDP-Com, and it instructs the agents when they should exchange information (e.g., information about each agent's observations). We focus on Dec-POMDP-Coms with four characteristics:

1. The state space S is *factored* : every combination of local states taken from S_1 and S_2 respectively determines a single global state.
2. The transition probability P is *transition independent*, i.e., $P(s_i'|s_i, a_i) = P(s_i'|s_i, s_j, a_i, a_j)$ when agents i and j took actions a_i and a_j at states s_i and s_j respectively.
3. The Dec-POMDP-Com is jointly-fully observable, that is a Dec-MDP-Com.
4. The observations are independent: $O = O_1 \times O_2$.

A mechanism for decentralizing control MDC is a mapping from a decentralized process to two single-agent Markov processes. In particular, we define a mechanism MDC that decomposes a Dec-MDP-Com into two single-agent (local) MDPs built as follows:

$$MDC :< S, A_1, A_2, \Sigma, C_\Sigma, P, R >\rightarrow$$

$$[< S_1, A_1, \Sigma, C_\Sigma, P_1, R_1 >, < S_2, A_2, \Sigma, C_\Sigma, P_2, R_2 >]$$

Our Dec-POMDP-Com framework and mechanisms are general to capture models with discounted infinite horizon where the agents' objective is to maximize their joint reward. It also captures scenarios with goal-oriented agents, where even if the horizon is infinite, there is a probability of one of achieving a goal state. The cost of communication C_Σ may include in addition to the actual cost incurred

[2] We are interested in mechanisms that are applied each time the agents exchange information by communication. Nevertheless, the mechanism could also be considered by designers of systems where synchronization is not attained necessarily by direct exchange of communication (e.g., synchronization may be attained when a common enviromental feature's value can serve as a signal for the agents). We call our mechanisms, mechanisms for communication because they are applied when communication occurrs and the problems they create are based on this synchronized information attained by this communication.

by the communication, the cost resulting from the complexity of computing the decomposition (i.e., by applying the mechanism) as well as the cost resulting from the complexity of computing the agents' local policies. R_1 and R_2 are the local reward functions of the corresponding MDPs. One of the main concerns of the mechanism is to find a "good" decomposition of the joint reward R into two local rewards such that the local policies of action will lead the agents to maximize the global reward R. In the simplest case, the Dec-POMDP-Com is reward independent, i.e., R_1 and R_2 are clearly summed up to output R. In the general case, the Dec-POMDP-Com is not reward independent, and R is given by a function of the local rewards that may add in a non-additive way (e.g., sub-additive or super-additive) depending on both agents doing complementary or redundant actions.

An approach to decentralized control with mechanisms for communication —. We assume that the agents are fully-synchronized when they start operating. Then, the mechanism MDC is applied on the global problem (with the synchronized information) resulting in two local problems (each one is a single agent MDP). Since we assume that the policy of communication of each agent is at the meta-level of control, any agent may initiate communication while solving this local problem. These policies of communication trade-off the cost of communication with the value of the information obtained. Whenever the agents exchange their local information, they become fully-synchronized. Then, they apply the mechanism and work on their possibly new local problems. The agents' policies of communication are computed assuming a certain mechanism for decentralization is given. The mechanism decomposes the decentralized MDP problem into two local MDP problems: each agent's local policy of action δ_i^A is computed based on its *local* reward function R_i, and the policy of communication δ_i^Σ is computed based on the *joint* reward R and based on the agents local policies of actions given a mechanism.

Figure 1 shows how both policies of action and communication are computed off-line, and then executed on-line. The optimal policy of action is found by solving the MDP resulting from applying the mechanism on the decentralized problem. **SolveComm** is a function that computes the policy of communication (either an approximation or an optimal policy).[3] The communication policy is computed based on this local policy of action, evaluated with the joint reward of the system. During the on-line stage, each time that an agent communicates based on this policy, both agents become fully-synchronized. At that point the mechanism is applied again and the corresponding optimal policy of action will be executed.

Off-line
SolveDec(Dec-MDP-Com,$Agent_i$){
/$*$ Dec-MDP-Com=
$< S = S_1 \times S_2, A_1, A_2, P = P_1 \times P_2, R, \Sigma, C_\Sigma >$ $*$/
  MDP_i = MDC(Dec-MDP-Com)
  $\delta_i^{A^*} \leftarrow$ **Solve**(MDP_i)
  $\delta_i^{\Sigma} \leftarrow$ **SolveComm**($P, R, \Sigma, C_\Sigma, \delta_i^{A^*}$)
}

On-line
Execute(Dec-MDP-Com){
  Do{
    MDP_i = MDC(Dec-MDP-Com)
    While ($\delta_i^{\Sigma} = \emptyset$){
      Execute $\delta_i^{A^*}$
    }
    Communicate s_i
  Until `Done`
}

Figure 1: Off-line and On-line application of a mechanism for communication

3.1 Evaluation Criteria

In this section we characterize the possible mechanisms for communication. We distinguish between characteristics of a mechanism that are relevant to the decomposition itself, and those of a mechanism that depend on a policy of communication assumed. In the related areas (reviewed in Section 4), efficiency is captured as maximizing the utility of the system. When designing mechanisms for communication, we should also consider the relation between temporary goals adopted or the near-term accumulated rewards and the global goal or optimization function of the system. We emphasize here the cooperative characteristic of the systems we are interested in and the fact that agents are exchanging information to achieve a joint goal or to optimize a joint global reward function.

The first group of features that characterize any mechanism is as follows:

- **Stationary** — A mechanism is stationary if for any state on which it is applied, it always results in the same decomposition of sub-problems. The result of applying a mechanism is not time-dependent.
- **Feasible** — A mechanism is feasible if it is applicable, i.e., the states, language and goal states (if relevant) do exist in the agents' specifications. The mechanism should be implementable on any possible global state.
- **Computational complexity** — The computation of the MDC mapping should be practical for resource-bounded agents so that they can actually communicate and coordinate their actions. There is a trade-off between the complexity of computing a mechanism and the joint reward of the system. There may not be a simple way to split the Dec-POMDP into two separate processes.
- **One-Pass** — After the exchange of information has occurred, (which lead to the application of the mechanism) a one-pass mechanism does not require any additional information for the agents to fully determine their local MDPs. For example, if each MDP has a single local goal to work on, then the mechanism that decomposed the global problem into these MDPs is indeed one-pass. Otherwise, agents may need to negotiate over which MDP each one should solve. A mechanism is, in general, a way to decompose one problem into two local problems. If this decomposition is ambiguous then the mechanism is not one-pass.

The remaining properties depend on the mechanism as

[3]Sometimes, it is not necessary to actually solve the local MDPs in order to compute the policy of communication. This can be done analytically knowing how the mechanism works (e.g. see [12]).

well as on the communication policy.

- **Complete** — If the Dec-POMDP-Com has a set of goal states, then a mechanism is complete if there exists a communication policy such that it guarantees that the agents reach one of these goals whenever it is possible.
- **Bounded** — Given a problem domain with goals then a mechanism is bounded if there exists a communication policy that can be applied at most k times for some given k.
- **Efficient** — A mechanism MDC_1 is more efficient than another mechanism MDC_2 if there exists a communication policy such that the joint reward attained by the system when MDC_1 is implemented is larger than the joint reward attained by the system when MDC_2 is implemented for any communication policy. The policy of communication takes into account the cost of computing the mechanism and solving each MDP. A mechanism is *optimal* for a certain problem if it is more efficient than any other mechanism.

In the infinite-horizon case, every complete mechanism is bounded for some k. If T is the finite horizon of the Dec-POMDP-Com, then every complete mechanism is bounded for some $k \leq T$.

In [12], we have introduced the Meeting under Uncertainty scenario, where two agents that cannot recognize each other need to meet at some location in a two-dimensional grid. Each agent's actions include movements to the east, west, north and south from its current location. Each action succeeds with a given probability P, and it fails with probability $1-P$ leaving the agent at the same location where it performed the action. Whenever the agents communicate, the information exchanged consists of each agent's observation that is given by their location coordinates.

A mechanism that can be applied to this decentralized control problem enables each agent to adopt a local goal and optimally move towards it based on its MDP. We have implemented the mechanism that lead the agents to adopt a local goal that is located at the middle of the agents' Manhattan distance. This mechanism is stationary, feasible, has low computation complexity (each agent computes the location at the middle of the Manhattan distance between them with the information acquired by communication), and it is a one-pass. Once this goal location is determined, each agent can solve its own MDP and reach that location. We show in [12] one myopic-greedy policy of communication for which this mechanism is complete and bounded, each agent indeed can reach its local goal, and the mechanism and the policy of communication are applied a finite number of times. We claim that letting the agents meet at the middle of their Manhattan distance in a two dimensional grid without obstacles is the most efficient mechanism. The joint expected time to meet was computed for any pair of distances possible between the two agents. The minimal value is attained when these distances are equal. The joint expected time to meet, Θ, when agent 1 is at distance d_1 from the meeting location, agent 2 is at distance d_2 from that location, and there is a cost of 1 for each time unit that passes and the agents have not met, is given by the following formula: (P is the transition probability of the Dec-POMDP-Com)

$$\Theta(0,0) = 0$$

$$\Theta(d_1,0) = P(-1+\Theta(d_1-1,0)) + (1-P)(-1+\Theta(d_1,0))$$

$$\Theta(0,d_2) = P(-1+\Theta(0,d_2-1)) + (1-P)(-1+\Theta(0,d_2))$$

$$\Theta(d_1,d_2) = P^2(-1+\Theta(d_1-1,d_2-1)) + P(1-P)(-1+\Theta(d_1-1,d_2))+$$

$$+(1-P)P(-1+\Theta(d_1,d_2-1)) + (1-P)^2(-1+\Theta(d_1,d_2))$$

When the grid is of size 10×10, the minimal expected time to meet is obtained when $d_1 = d_2 = 9$ and the expected value is -12.16.

The value of a centralized fully-observable solution can serve as an upper bound to the optimal decentralized solution. We have also started to identify sufficient conditions under which an optimal mechanism exist.

4. MECHANISM DESIGN - RELATED WORK

Mechanism design was originally studied in Game Theory to design games that yield outcomes with certain characteristics. Later, research in Computer Science has looked at adapting this approach to achieve social coordination and optimization of social welfare in distributed systems. We are interested in mechanisms that result in near-term behaviors that produce good approximations to the optimal control of a decentralized cooperative system. Here, we review the classic approach to mechanism design, and then follow with later studies done for computational systems.

4.1 The Economic Approach

Mechanism design or implementation theory is studied in Game Theory [21] in order to find rules for a game with certain characteristics. The players in this game, have each a preference function over the outcomes of the game. Given a choice rule from profiles of preferences to a subset of feasible outcomes, the question is whether a game can implement this choice rule in a such a way that a certain solution concept is attained (e.g., the Nash equilibrium is reached). The players are self-interested and therefore information about their own preferences is kept private. A designer of the game looks for a mechanism that will produce the desired outcome (e.g., a Nash equilibrium) when the players reveal some part of their information as input to the designer. Notice that following our approach, each time that the agents apply the mechanism MDC, they are faced with a problem to solve. In the economic approach the mechanism itself solves the problem.

An algorithmic view to mechanism design is found in [20]. The mechanism designer sets the algorithm for interaction among the agents and a payment structure that motivates the agents to participate in the interaction. Again, this literature is concerned with agents that are self-interested and may hold privately known information about their preferences. Thus, the main question handled by a designer of a mechanism is to combine the private preferences of the players into an outcome state that corresponds to the "social choice". Since Nisan and Ronen took an algorithmic approach to mechanism design, they were interested in *polytime computable mechanisms*, i.e., mechanisms whose output and payment functions are computable in polynomial time.

Following this economic approach agents are self-interested and are not willing to reveal their private information that may help their competitors to take advantage of their actions. Therefore, an important notion in this approach is whether the mechanism is *truthfully-implementable*, i.e., whether the implementation will induce the agents to report their true types and preferences. An example of a truthful implementable mechanism is the known Clarke mechanism [5], where a set of compensation rules is given as an incentive structure leading the agents to reveal their true preferences as their optimal strategies. In cooperative decentralized systems, it is clear that any mechanism for communication is truthfully-implementable. However, an interesting feature of these mechanisms is that the whole system may benefit if one of the agents does not send its actual observation, but a function of it. The hearer may take an action following the message received, and move eventually to a new state. Observing this new state, may lead the agent to take an additional action that was not the original aim of the message sent, but it is the result of an effect the sender of the message had on the resulting state. So, even though agents are truthful they may benefit by changing the content of the messages set by the designer. Notice that the contents of the messages are not set by the mechanism. The mechanism assumes that the agents are programmed knowing which are the messages they can exchange. Another interesting research area is to study the design of languages of communications for decentralized control.

Based on Wellman's definition of multi-agent systems [29], the role of a mechanism in a cooperative system (with global objectives) is "to coordinate local decisions and disseminate local information in order to promote these global objectives." We study mechanisms that will lead each agent to face possibly new local problems which are simpler to compute. At the meta-level the agents also compute a policy of communication that enables the agents to coordinate their local information attained by their local policies. The achievement of the global objectives of the system is a result of a process which interleaves the application of the mechanism which leads to near-term local behaviors and the exchange of information through communication from time to time that will induce the application of the mechanism and so forth.

4.2 Social Laws

The areas related to mechanism design in distributed artificial intelligence are social laws and negotiation mechanisms. Social laws were defined as mechanisms of coordination. Two approaches were studied. Shoham and Tennenholtz [25] define social laws as constraints on the agents' actions. Goldman and Rosenschein [11] define social laws as extensions to the agents' local plans of actions.

Shoham and Tennenholtz study social laws as mechanisms for coordination that will induce agents to avoid conflicts between their actions. A social law is a predicate over a local state prohibiting some of the agent's actions that it is indeed capable of performing. Once the social law is imposed on a multi-agent scenario, its effects are transparent to the agents. A modified multi-agent system is created in which only the permitted actions and the corresponding transitions are allowed. When designing mechanisms for communication, the agents are actively applying the convention and eventually may solve new local problems. In our case, agents' plans of actions can be affected due to messages received on-line. In the social-laws analysis, once the law is imposed, the agents find a plan of action that is not going to change anymore because of the law. They consider a special set of states denoted by the focal states. A useful social law induces legal plans such that every plan execution that includes a focal state s_1 will also include another focal state s_2, for any focal states. A legal plan is one that does not choose prohibited actions. For a given multi-agent system and a given set of focal states, it was proved that finding a useful social law or announcing that it does not exist is NP-complete [25].

More recently [26], a *rational social-law* was defined assuming that the agents play a game g, and that a social law sl induces a sub-game g_{sl} of g that includes only the actions that are not prohibited by sl.[4] In a game theoretical sense, rational agents are captured as utility maximizers. As such, the solutions that will be preferred by the agents in such settings will be either the maximin strategies, Nash equilibrium, or Pareto Optimal strategies(see [26] for more details about these solutions). For any such solution concept variable V, $V(g)$ will denote the value of that variable in the game g.

DEFINITION 4. *(adapted from [26]) Let g be a game, V a game variable (i.e., maximin strategy, Nash equilibrium or Pareto optimality), and $<$ and ordering on the possible values of this variable. A social law sl is rational with respect to g and V if $V(g) < V(g_{sl})$.*

Goldman and Rosenschein [11] define social laws as extensions to the agents' local plans of actions. Social laws are intended to transform the world state (global state) for the benefit of the whole system. Each agent is assigned a level of cooperation value that determines how much effort the agent will invest in extending its own plan in order to follow the law. This parameter can lead different agents to either not follow the law or to do it at different levels. The social law in this case was studied as a simpler (less complex) means to reach coordination instead of computing the optimal multi-agent joint plan. Mechanisms for communication are also introduced to reduce the complexity of solving the complete decentralized control problem with communication in the sense that local policies are affected by information obtained by exchange of information and eventually lead to approximations of the global optimal behavior. This paper deals with local optimization problems that are affected by information received from other agents. Social laws are more strict in the sense that either they prohibit the execution of certain actions, or demand the execution of longer plans in order to follow the social rule. In our case, the information exchanged by the agents together with the mechanisms imply two local problems that need to be optimally solved by each one of the agents with their own set of actions. The mechanism that determines the local problems depends on the communication of different messages and on their timing.

The implementation of social laws in the two approaches aforementioned is aimed at improving the coordination level of the multi-agent system. Flexible social laws were studied by Briggs and Cook [4]. Agents are allowed to choose from laws with various levels of strictness starting with the most strict and moving to more lenient laws when they cannot

[4] A *social convention* [26] is a social law that restricts the agents' behaviors to one particular strategy.

succeed in finding a plan. These social laws follow the approach taken by Shoham and Tennenholtz as restrictions to the agents' actions to reduce the chance of interaction between the agents. Mechanisms for communication are not intended to avoid interactions, the assumption is that for certain costs of communication, the exchange of information is indeed beneficial. The mechanisms for communication will induce the agents to interpret the messages received in order to better coordinate and thus attain increased joint utility. The works explained so far assume that the social laws are designed off-line. Another line of research study the emergence of these conventions [27, 26].

In the cases described above, the motivation for the agents to follow these social laws is implicit in the fact that the agents comprise a cooperative system, and therefore it is in their benefit to implement the law. In the next section, we describe work done on self-interested agents who need to be motivated to follow a social mechanism.

4.3 Negotiation Mechanisms

Self-interested agents need to be motivated to follow a certain interaction mechanism (similarly to the economic approach). Negotiation mechanisms were developed by Zlotkin and Rosenschein [24]. They suggested a negotiation protocol over possible joint deals. This protocol can either end in reaching the conflict deal (i.e., no cooperation is beneficial and each agent ends up performing its initially locally assigned deals) or the negotiation ends with an agreement that is some division of the set of both agents' deals. In this process the goal-oriented agents are interested in achieving their *pre-set local* goals at the minimum cost through possible cooperation and resolution of conflicts if they exist. The goal of this line of research is to find distributed consensus mechanisms such that agents that follow simple and stable strategies will obtain efficient (Pareto Optimal [17][5]) outcomes.

This work does not deal with sequential decision making. The agreements could be over a set of many tasks that eventually will be performed in a sequence, but the negotiation process is over all the possible deals as one decision. All the possible deals are already known when the negotiation mechanism is applied. In our case, we are interested in applying the communication convention as part of the control process in order to optimally behave and communicate.

The negotiation mechanism studied in [24] is monotonic and therefore ensure convergence to a deal. They also studied incentive-compatible mechanisms, and show that whether an agent will benefit from lying or not depends on the domain characteristics (e.g., concave/sub-additive/modular Task Oriented Domains). For example, it was proved that any lie (i.e., hidden, phantom or decoy) is not beneficial in any encounter of two agents in concave[6] Task Oriented Domains with any optimal negotiation mechanisms over all-or-nothing deals.

4.4 Evaluation Criteria

Mechanism designers in Economics (e.g., [19, 21]) are interested in *stable* mechanisms so that the self-interested agents will not be able to manipulate them for their own benefit. Thus, mechanisms are sought to implement a solution concept such as dominant strategies or Nash equilibrium for example. Economists are also interested in *truthfully-implementable* mechanisms that will induce the players to report their true preferences to the system designer. A good mechanism should have the following characteristics: *strategy-proof*, *efficient* and *budget-balanced* [15]. A mechanism is strategy-proof if the agents are motivated to participate in it and will reveal their true preferences. A mechanism is efficient if its output state maximizes the utility of the system (i.e., the social-welfare is optimized taking into account the individual selfish utilities of the agents). A mechanism is budget-balanced if the total monetary transfer from the agents to the center (the system designer) is non-negative. Kfir-Dahav et al. show [15] that the procedures of the Clarke mechanism used to optimize the social welfare are NP-hard and they suggest a heuristic that will maintain the strategy-proof and budget-balanced features at the expense of social-welfare efficiency.

According to the economic approach, an optimal social-law is one that attains maximal utility at the system level. When social-laws are applied to multi-agent systems, this optimality definition lacks the features of flexibility and usefulness. Useful social-laws enable the agents to work individually in a mutually-compatible manner [8]. Optimality is then seen as maximal flexibility of the agents to react to unpredictable changes in their environment while maintaining the ability to reach their original goals. The two features for evaluating social laws based on [8] are minimality and simplicity: *Minimal* social-laws restrict the agents' set of actions to the extent that it is necessary. Since restricting these actions limit the freedom of the agent to plan its behavior, a "good" social-law needs to balance this freedom with the need to avoid collisions between the agents' actions. Still, the agent needs some freedom to be able of finding a feasible plan, that is the law should impose the minimal number of constraints as possible on the agent's actions. A *simpler* social law is easier to implement.

The features discussed in [11] for cooperative state-changing rules include the following. A rule is: 1) *guaranteed* if there is certain that it will not increase global work. 2) *reversible* if its effects can be undone. 3) *redundant* if performing the extra work will cause the agent to remain in the same state. 4) *resource-dependent* if following the rule implies the use of consumable resources. 5) *state-dependent* if the rule can be apply only in a certain state.

Negotiation mechanisms [24] were evaluated based on the following criteria: 1) *Symmetric distribution*, i.e., no agent is to have a special role in the negotiation mechanism. In our case, since the agents are cooperative we do not risk having manipulating malevolent agents. However, the designer may want a more capable agent to have more influence on another agent when it communicates (e.g., by sending an instruction message). 2) *Efficiency*, i.e., the solution arrived at through negotiation should be efficient (e.g., satisfy the criterion of Pareto Optimality). 3) The strategies should be *stable* (e.g., strict Nash equilibrium where no single agent can benefit by changing its strategy, though a group might). 4) *Simplicity* , i.e., there should be low communication cost to the mechanism as well as relatively low computational complexity.

[5]A deal is Pareto optimal if it cannot be improved for one agent without decreasing the utility of another agent from the same deal.

[6]A Task Oriented Domain is concave if for all finite sets of tasks $X \subseteq Y, Z \subseteq T, c(Y \bigcup Z) - c(Y) \leq c(X \bigcup Z) - c(X)$. c is the cost function.

We are concerned with the design of mechanisms for communication in cooperative decentralized systems. Intuitively, agents exchange information to synchronize their knowledge and obtain full observability of the global state. Since communication has a cost associated with it, agents could only be synchronized from time to time. In between these periods agents work in a local manner on problems set by a mechanism such that eventually the agents approximate the actual global objective. Each local solution is computed optimally, and the policy of communication is an approximate or an optimal solution given a mechanism. In other words, the mechanism for communication is a means to interpret messages received and translate them into near-term problems that can be optimally solved locally. Notice that the communication in our case is not in the form of KQML commands where the reaction to a message received is the clear and expected response action to the performative command.

In a figurative manner, we can see why these approaches are different (see Figure 2):

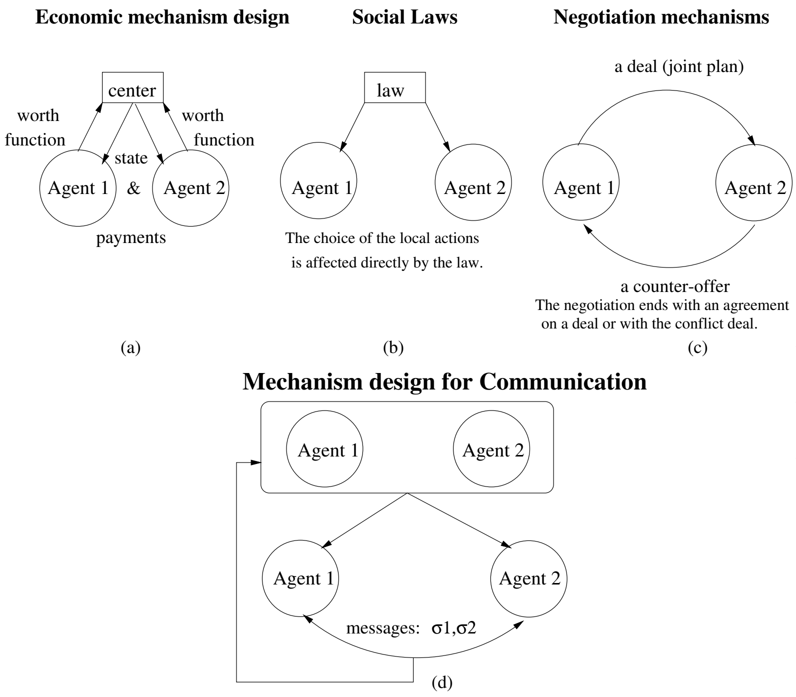

Figure 2: A comparison between mechanism design approaches.

Cases (a) and (d) — Economic agents are self-interested. The agents send their state-worth functions to the center who outputs a state and a payment structure. The agents' input to the center states how much they are willing to pay for a certain output-state (i.e., how desirable that state is with regard to the goal the agent intends to achieve). A mechanism sets the rules for a game; in our approach, a mechanism decomposes a global problem into two temporary local problems which need to be optimized to solve the original global problem. The agents may communicate based on the policy of communication found given the mechanism, in which case, the mechanism will be applied again. The game-theoretic approach sets the game to be played by the agents; in our case, the mechanism only determines temporary near-term problems which all together will compose the solution to the long-term problem. In the economic view, the mechanism itself is the computation of the solution. In the communication case, the agents need to find the optimal solution to each problem they are faced with.

Cases (b)(Goldman and Rosenschein approach) and (d) — Although the convention induces the agents to change a global state, by adopting a temporary local subgoal, the parameters of the law are based on the features of the agent's own observations, and not as a result of another agent sending its observations.

Cases (b)(Shoham and Tennenholtz approach) and (d) — This approach imposes a social law on the multi-agent system and then a transformed system is obtained whose set of actions is restricted to only those that are allowed by the law. From then on, the agents plan as usual. In our case the convention is actively applied by the agents while they compute their optimal policies of action and communication. The convention is based on the messages sent by the agents along the process.

Cases (c) and (d) — The mechanism is applied once and it provides the agents with the solution of a joint plan. Agents are self-interested and they negotiate assuming that both agents have fully-observable information (although it may not be reliable).

5. CONCLUSIONS

This paper has introduced the notion of mechanism design for communication in collaborative multi-agent systems. No algorithm is known today that can optimally solve the decentralized control problem with communication. Mechanism design is a means that enables the designer of a decentralized system to decompose a complex multi-agent problem into temporary, simpler, local problems. In addition, the agents compute a policy of communication that will allow them to synchronize their local information from time to time, based on the trade-off between the cost of this communication and the value of the information acquired. This paper presents the first steps that complements our research on decentralized control with communication, by formally presenting a framework where different mechanisms can be designed and further analyzed.

We have compared the mechanism design approach for communication with mechanism design studied in economy and distributed artificial intelligence (social laws and negotiation mechanisms). We have compared the approaches and have also proposed a set of criteria to evaluate the mechanisms for communication. The complexity analysis of solving the Dec-POMDP-Com with and without a mechanism for communication remains an open research question. Analyzing the competitive ratio of the algorithm that solves the decentralized control problem with mechanisms for communication is open as well (i.e., what the mechanisms are that will always attain a joint utility that is at most distant by a constant c from the maximal value of the optimal joint policy). We are also interested in the learning process where agents dynamically adapt the conventions for communication.

Acknowledgments

This work was partially supported by AFOSR Grant No. F49620-03-1-0090. Any opinions finding and conclusions or recommendations expressed in this publication are those of

the author(s) and do not necessarily reflect the views of the AFOSR.

6. REFERENCES

[1] Raphen Becker, Shlomo Zilberstein, Victor Lesser, and Claudia V. Goldman. Transition-independent decentralized Markov decision processes. In *Proceedings of the Second International Joint Conference on Autonomous Agents and Multi-Agent Systems*, Melbourne, Australia, 2003.

[2] D.S. Bernstein, R. Givan, N. Immerman, and S. Zilberstein. The complexity of decentralized control of Markov decision processes. *Mathematics of Operations Research*, 27(4):819—840, 2002.

[3] Craig Boutilier. Sequential optimality and coordination in multiagent systems. In *Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence*, pages 478–485, Stockholm, Sweden, 1999.

[4] Will Briggs and Diane Cook. Flexible social laws. In *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence*, Montreal, Quebec, 1995.

[5] Edward H. Clarke. Multipart pricing of public goods. *Public Choice*, 11:17—34, 1971.

[6] Edmund H. Durfee. *Coordination of Distributed Problem Solvers*. Kluwer Academic Publishers, Boston, 1988.

[7] Tim Finin, Yannis Labrou, and James Mayfield. KQML as an agent communication language. In Jeff Bradshaw, editor, *Software Agents*. MIT Press, 1997.

[8] David Fitoussi and Moshe Tennenholtz. Choosing social laws for multi-agent systems: Minimality and simplicity. *Artificial Intelligence*, 119, 2000.

[9] Piotr J. Gmytrasiewicz, Matthew Summers, and Dhruva Gopal. Toward automated evolution of agent communication languages. In *Proceedings of the 35th Hawaii International Conference on System Sciences (HICSS-35)*, 2002.

[10] Dani Goldberg and Maja J. Mataric. Coordinating mobile robot group behavior using a model of interaction dynamics. In *Proceedings of the Third International Conference on Autonomous Agents (Agents '99)*, pages 100–107, Seattle, Washington, 1999.

[11] Claudia V. Goldman and Jeffrey S. Rosenschein. Emergent coordination through the use of cooperative state-changing rules. In *Proceedings of the Twelfth National Conference on Artificial Intelligence*, pages 408–413, Seattle, Washington, 1994.

[12] Claudia V. Goldman and Shlomo Zilberstein. Optimizing information exchange in cooperative multi-agent systems. In *Proceedings of the Second International Joint Conference on Autonomous Agents and Multi-Agent Systems*, Melbourne, Australia, 2003.

[13] Barbara J. Grosz and Sarit Kraus. Collaborative plans for complex group action. *Artificial Intelligence*, 86(2):269—357, 1996.

[14] Junling Hu and Michael P. Wellman. Multiagent reinforcement learning: Theoretical framework and an algorithm. In *Proceedings of the the Fifteenth International Conference on Machine Learning (ICML98)*, pages 242–250, Madison, WI, 1998.

[15] Noa Kfir-Dahav, Dov Monderer, and Moshe Tenenholtz. Mechanism design for resource-bounded agents. In *Proceedings of the Fourth International Conference on Multi-Agent Systems*, 2000.

[16] Michael L. Littman. Markov games as a framework for multi-agent reinforcement learning. In *Machine Learning: Proceedings of the Eleventh International Conference*, pages 157—163, 1994.

[17] R. Duncan Luce and Howard Raiffa. *Games and Decisions*. John Wiley and Sons, Inc., 1957.

[18] Maja J. Mataric. Learning social behaviors. *Robotics and Autonomous Systems*, 20:191–204, 1997.

[19] John Moore. Implementation, contracts, and renegotiation in environments with complete information. In Jean-Jacques Laffont, editor, *Advances in economic theory Sixth World Congress Volume 1*, pages 182–282. Cambridge University Press, 1992.

[20] Noam Nisan and Amir Ronen. Algorithmic mechanism design. In *Proceedings of the Thirty First Annual ACM Symposium in Theory of Computing (STOC)*, 1999.

[21] Martin J. Osborne and Ariel Rubinstein. *A Course in Game Theory*. MIT Press, 1994.

[22] Leonid Peshkin, Kee-Eung Kim, Nicolas Meuleau, and Leslie Pack Kaelbling. Learning to cooperate via policy search. In *Proceedings of the the Sixteenth Conference on Uncertainty in Artificial Intelligence (UAI00)*, pages 489–496, Stanford, CA, 2000.

[23] David V. Pynadath and Milind Tambe. The communicative multiagent team decision problem: Analyzing teamwork theories and models. *Journal of Artificial Intelligence Research*, 16:389–423, 2002.

[24] Jeffrey S. Rosenschein and Gilad Zlotkin. *Rules of Encounter*. MIT Press, Cambridge, Massachusetts, 1994.

[25] Yoav Shoham and Moshe Tennenholtz. On social laws for artificial agent societies: Off-line design. *Artificial Intelligence*, 73:231–252, 1995.

[26] Yoav Shoham and Moshe Tennenholtz. On the emergence of social conventions: modeling, analysis and simulations. *Artificial Intelligence*, 94, 1997.

[27] Adam Walker and Michael Wooldridge. Understanding the emergence of conventions in multiagent systems. In *Proceedings of the First International Conference on Multi-Agent Systems, ICMAS'95*, 1995.

[28] Jun Wang and Les Gasser. Mutual online concept learning for multiple agents. In *Proceedings of the First International Joint Conference on Autonomous Agents and Multi-Agent Systems*, pages 362–369, Bologna, Italy, 2002.

[29] Michael Wellman. Multiagent systems. In R. Wilson and F. Kiel, editors, *MIT Encyclopedia of Cognitive Sciences*, pages 573—574. MIT Press, 1999.