Artificial Social Systems*

Yoram Moses

Department of Applied Math and CS
The Weizmann Institute of Science
Rehovot, 76100 Israel

Moshe Tennenholtz

Faculty of Industrial Engineering and Management
Technion – Israel Institute of Technology
Haifa 32000, Israel

Abstract

An *artificial social system* is a set of restrictions on agents' behaviors in a multi-agent environment. Its role is to allow agents to coexist in a shared environment and pursue their respective goals in the presence of other agents. This paper argues that artificial social systems exist in practically every multi-agent system, and play a major role in the performance and effectiveness of the agents. We propose artificial social systems as an explicit and formal object of study, and investigate several basic issues that arise in their design.

Keywords: Social Laws, Multi-Agent Systems, Off-Line Design

*This work was supported in part by the US-Israel Binational Foundation. The work of the first author was supported by an Alon Fellowship, and by a Helen and Marcus Kimmelman Career Development Chair. The second author was supported in part by an Eshkol Fellowship of the Israeli Ministry of Science and Technology, and later by the Air Force Office of Scientific Research. Part of the research was carried out while the second author was in the department of Applied Mathematics and CS in the Weizmann Institute, and part while he was in the CS department at Stanford University.

1 Introduction

Consider the following examples of environments involving many agents:

- A large automated warehouse uses fifty robots to fetch and store products, and serves tens of customers at a time.
- A truck company with a high volume of activity and many branches uses a large number of drivers and trucks to move goods across the country in an efficient way. The scheduling of drivers and trucks at the different branches is done locally, but affects the company's ability to respond to demands in the future.
- A major software project employs a large number of programmers. Different programmers write different parts of the program code, but the final behavior of each programmer's segment can affect and is affected by the software written by the others.

Each of the above contexts involves multiple agents whose actions are interdependent: What one does may affect the success of the other's actions. In addition, in each of the examples the purpose of distinct agents' actions at any given time may be motivated by different sources: Different robots in the warehouse may be serving different clients, two branches of the truck company may be involved in hauling goods for different customers, and different programmers may be handling different aspects of the software project. Obviously, in all of these cases the agents face a coordination problem. Indeed, similar coordination problems arise in most any system involving many agents that operate in a shared environment, in which the actions of one agent can affect the success of another agent's activities.

One extreme solution to such coordination problems is to provide centralized control of all of the relevant activities. For example, we could imagine having a central entity that determines the actions of all of the robots in the warehouse and ensures that no conflict or accident occurs. Such centralized solutions, however, come with a cost. As the number of agents and tasks involved grows, the costs in communication, synchronization, and processing grow dramatically. Moreover, the system depends crucially on such things as the reliability of the central element, of the communication lines, etc. In large and complex systems, these costs can become prohibitive. At the other extreme is the purely decentralized approach, in which the agents attempt to act in a completely independent manner. As a result, conflicts may arise. The goal then would be to work on methods of resolving conflicts once they arise. Clearly, however, if the cost of a conflict is dear, or if conflict resolution is difficult, completely independent behavior becomes unreasonable.

In this work, we suggest an intermediate approach to the design of multi-agent systems: Agents should be designed to act individually, but their actions should be restricted so that their behavior is mutually compatible. Such restrictions on agents' behaviors we call an *artificial social system*. An artificial social system is a mechanism for coordination that

reduces the need for centralized control. Moreover, by following a well-designed social system, agents can avoid many potential conflicts to begin with. Thus, artificial social systems bridge the gap between the completely centralized and purely decentralized approaches to coordination. Indeed, they allow for a wide spectrum of intermediate solutions.

Most actual systems involving many agents acting in a shared environment can be thought of as employing some type of social system. In human societies, for example, the social system consists of the legal system, together with various conventions regarding how people behave. Societies of animals, too, have conventions of behavior that constitute their social system [40]. We can also view conventions and restrictions employed in artificial multi-agent systems as constituting a social system. Our thesis, however, is that artificial social systems should be treated *explicitly* as a major component of the design of multi-agent systems. Our purpose in this work is to initiate the study of artificial social systems as an explicit and formal paradigm for design.

This paper presents the notion of *artificial social systems*. It is based on the original manuscripts that presented this notion [24, 25, 38]. We describe how artificial social systems suggest an approach to the design of multi-agent systems. Tradeoffs involved in the design of multi-agent systems that this approach uncovers are presented, and a methodology for design based on this approach is offered. Various issues of concern to the distributed/decentralized AI communities (DAI/DzAI) [3, 7, 8] are shown to fit naturally into the artificial social systems framework. Finally, we present semantics and a formal logical syntax in which reasoning about such systems can be carried out. This work has been followed by a number of papers that use and extend the notion of artificial social systems (e.g., [34, 33, 39]).

This paper is organized as follows. In the next section we introduce the idea of artificial social systems in the framework of a simple but widely applicable model. In Section 3 we discuss a number of essential aspects in the design of a social system. In particular, we discuss the *golden mean problem in artificial societies*, which we consider to be the central problem in the design of artificial social systems. In Section 4 we discuss the general semantics of artificial social system, and logical reasoning about such systems. Section 5 provides some final remarks, and discusses related work.

2 Social Automata

In this section we consider a simple framework in which we will demonstrate the idea of artificial social systems. While the idea can be used in more complex settings, the framework presented and discussed in this section already embodies many of the relevant issues.

Generally speaking, a multi-agent system consists of several agents. We assume that at any given point, each such agent is in one of several states. The agent's state represents the current situation of the agent, from the agent's point of view. In each of its states, an

agent can perform several actions. The actions an agent performs at a given point may affect how the state of this agent and the states of other agents will change. We now define an automata-based model of multi-agent activity in which these aspects come to play. This model will be used to study issues related to artificial social systems.

A system of *dependent automata* (DA system) consists of two (or more) agents, each of which may be in one of a finite number of different local states. We denote the set of local states of an agent i by L_i. The list $\langle s_1, \ldots, s_n \rangle$ of states of the different agents is called the system's *configuration*. At every step, each agent performs an action. The set of possible actions an agent i can perform is in general a function of the agent's local state. Thus, for every state $s \in L_i$ there is a set $A_i(s)$ of actions that i can perform when in local state s.

Let us call the list $\langle \mathtt{a}_1, \ldots, \mathtt{a}_n \rangle$ of actions the different agents perform at a given point their *joint action* there. An agent's next state is a function of the system's current configuration and the joint action performed by the agents. At any given point, a goal for an agent is identified with one of its states. We assume each agent has a set of potential goals it might like to reach. Each agent starts in a state taken from a set of potential initial states. We assume that an agent can perform computations to plan how to attain its goal, and to determine what actions to take at any given point. In such a model, the success of one agent's actions may depend in a crucial way on the other agents' actions.

Formally, a *plan* for agent i in a DA system is a function $p(s)$ that associates with every state s of agent i a particular action $\mathtt{a} \in A_i(s)$. A plan is said to *guarantee* the attainment of a particular goal starting from a particular initial state in a given DA system if by following this plan the agent will attain the goal, regardless of what the other agents do and what the initial states of the other agents are.

Clearly, computing such plans can be rather complex. Moreover, the resulting plan might not be very useful. A plan that needs to be able to respond to any possible behavior by the other agents may be very inefficient[1] in the number of steps it takes. In some cases, such a plan may even fail to exist!

A DA system is said to be *social* if, for every initial state s^i and goal state s^g, it is computationally feasible for an agent to devise, on-line, an efficient plan that guarantees to attain the goal s^g state when starting in the initial state s^i. DA systems in which the sets $A_i(s)$ represent the set of actions that agent i is physically capable of performing at state s will often fail to be social. For example, in a situation where an agent waiting at an intersection may at any point in time choose to move into the intersection, no other agent can have a plan that guarantees it will safely cross this intersection. We shall modify a DA system by what we call a *social law*. Formally, a social law Σ for a given DA system S consists of functions $\langle A'_1, A'_2, \ldots, A'_n \rangle$, satisfying $A'_i(s) \subseteq A_i(s)$ for every agent i and state $s \in L_i$. Intuitively, a social law will restrict the set of actions an agent is "allowed" to perform at any given state. Given a DA system S and a social law Σ for S, if we replace

[1] Unless stated otherwise, we will assume that a problem is feasible, efficient, or tractable, if there exists a polynomial algorithm for solving it. Other assumptions can be treated similarly.

the functions A_i of S by the restricted functions A'_i, we obtain a new DA system. We denote this new system by S^Σ. Intuitively, in S^Σ the agents can behave only in a manner compatible with the social law Σ.

From the point of view of artificial social systems, a number of computational questions are natural at this stage. These computational problems relate to finding a set of restrictions (called the social law) on the actions performed by different agents at different states of the original DA system. The restrictions will usually be determined off-line before the initiation of activity and will induce a system where agents are able to (efficiently) achieve their goals during the on-line activity. For example, given a DA system S we may be interested in restricting the agents' actions by a social law Σ to yield a system S^Σ so that either:

(a) in S^Σ every agent has, for each of its goals s^g, a plan guaranteed to achieve s^g;

(b) condition (a) holds, and the plans prescribed by (a) are efficient;

(c) condition (a) holds, and the problem of computing plans in S^Σ is tractable; or

(d) both (b) and (c) hold. (In this case we consider the system S^Σ to be *social*.)

Various assumptions about the structure of the DA system, for example regarding the number of local states agents have, or the number of actions an agent can perform in a state, may affect the above-mentioned computational problems. Analogues of problems (a)–(d) above will apply to more complex types of systems as well. We now turn to study a particular problem in the context of DA systems.

The following theorem illustrates the kind of computational results which we can obtain regarding the artificial social systems approach in the dependent automata setting. We first state the theorem and then discuss its interpretation. Formally, we will define the problem of designing a social system with respect to a DA system with n agents and an assignment of goals to each agent as follows. We are given a DA system $S = (L_1, \ldots, L_n, C_0, \mathsf{A}, A_1, \ldots, A_n, \tau)$, where the L_i's are sets of local states of agent i, $C_0 \subseteq \times L_i$ is a set of initial configurations, A is a set of actions and $A_i : L_i \rightarrow 2^{\mathsf{A}}$ ascribes a set of possible actions for each agent in every local state. Finally, τ is a transition function mapping configurations and joint actions into configurations. We define the *size* of such a system to be $|\mathsf{A}| + \max_i |L_i|$.[2] The goals are given by sets $G_i \subseteq L_i$, for $i = 1, \ldots, n$. A local state s of agent i is called an *initial state* if it appears in one of the configurations in C_0. We will be interested in a social law Σ, such that in S^Σ, given any agent i and any initial state $s_0 \in L_i$ and goal $s_i^g \in G_i$, there exists a plan $p^g : L_i \rightarrow A$ that is guaranteed to reach s_i^g starting from s_0.

We can show:

[2] Our results hold for other natural definitions of the size of the system as well. For example, they hold is we replace the term $\max_i |L_i|$ by $\Sigma_i(|L_i|)$.

Theorem 2.1: *Let $n \geq 2$ be a constant. Given a DA system S with n agents, the problem of finding a social law Σ, such that in S^{Σ} each agent can devise plans for reaching each goal state from each initial state, if such a law Σ exists, is NP-complete.*

Proving that a problem is NP-complete is usually taken as evidence that the problem is hard to solve. No efficient algorithms are known for solving NP-complete problems, and it is conjectured that none exist. In our case, however, the NP-completeness can be interpreted in a positive manner as well. Indeed, the fact that the problem is in NP shows that the verification of the design process can be done efficiently. Roughly speaking, the process of designing a social law in the setting of Theorem 2.1 corresponds to guessing a social law and associated plans, all of which can be encoded in polynomial space and can be verified in polynomial time. Since this design process will usually be done before the initiation of activity for a particular system and can be supported by automatic verification, we get that a trial and error procedure often becomes feasible in the design stage. As we will discuss later, the designer's ability to attempt to solve NP-hard problems off-line in the design stage is in general greater than the agents' ability to tackle such problems when they encounter a conflict on-line in the course of their activity.

The above discussion introduces a basic setting where artificial social systems can be discussed, and a (fairly positive) basic theorem regarding it. However, the main objective of the setting of DA systems in this paper is to illustrate the artificial social system approach; it is by no means the most general model in which the related ideas can be discussed and studied. One extension of this model is concerned with the case where the plans the agents execute are not restricted to be functions of their local state, but rather can depend on the full history of the agent's previous states and actions. In this extended setting, a social law is taken to be a restriction on the plans an agent might devise. We will refer to dependent automata setting with the extended notion of a plan as the *extended dependent automata setting*. If the (extended type of) social law enables to efficiently construct efficient (extended) plans for achieving each agent's goals from each of its initial states, then we will say the induced extended dependent automata setting is social. In the following, we are interested only in efficient plans, where the number of actions that might be executed in the course of following a given plan is polynomially bounded. We are able to show that an analogue of the above theorem (as well as its positive interpretations) holds for this extended setting as well.

Theorem 2.2: *Let $n \geq 2$ be a constant. Given an extended dependent automata setting with n agents, the problem of finding an extended social law that induces a social extended dependent automata setting, if one exists, is NP-complete.*

3 Designing Social Laws

In the previous section we studied social laws in the context of dependent automata. The same ideas apply in a much broader set of contexts. In general, we will have some model

of a multi-agent activity, and a social law will restrict the behavior of the agents in this model. Specifically, we can identify a plan for an agent with an individual *strategy* in the game-theoretic sense (or a *protocol* in the language of distributed systems). Intuitively, the social law will determine which strategies are "legal" and which are not. Nevertheless, most of the relevant issues remain the same as with dependent automata. For example, as we saw in Section 2, a social law can (i) enable an agent to design a plan that guarantees to attain a particular goal for which no plan exists without the social law; (ii) allow shorter and more efficient plans for certain goals; and (iii) simplify the domain in which plan design is performed, thereby simplifying the computational problem involved in designing a plan to reach a given goal.

Notice that in controlling the actions, or strategies, available to an agent, the social law plays a dual role: Roughly speaking, by reducing the set of strategies available to a given agent, the social system may limit the number of goals the agent is able to attain. By restricting the behaviors of the *other* agents, however, the social system may make it possible for the agent to attain more goals, and in some cases these goals will be attainable using simpler and more efficient plans than in the absence of the social system. An overly liberal social system will allow each agent to carry out a large number of strategies. The agent may therefore hope to be able to attain many useful goals. Other agents, however, are also able to carry out a very large number of strategies, and strategies of different agents are likely to be incompatible. As a result, instead of being able to devise plans to attain many goals, an agent may end up being able to attain only a small number of goals, and even they might be attainable only at a high cost to the agent. If, on the other hand, the social law is overly restrictive, then the number of legal strategies available to an agent is very small, and the likelihood of collision between different agents' strategies can be greatly reduced. In this case, however, the agents might be unable to attain their goals due to a lack of options in choosing what actions to perform.

A related issue is the fact that a social system will, in many cases, determine (possibly implicitly) what goals a given agent is able to attain, and what goals will be unattainable to the agent. Intuitively, there may often be cases in which allowing an agent to attain a certain goal may cause unreasonable damage to other agents. We can think of such goals as "anti social". The goal of hurting another agent is a blatant example of this. Less blatant examples may be getting on a bus without waiting in line. By forcing the agents to stand in line before getting on a bus, the social law may prevent some agents from attaining this goal, while making it possible for other, perhaps weaker or more polite, agents to receive fair service. We can associate with a given social system a set of *socially acceptable goals* for each agent. These are the goals that the social system allows the agent to attain.

Thus, in designing a social system, the designer needs to find a good compromise between competing objectives. From a given agent's point of view, an ideal social system would allow the agent to be able to attain as many goals as possible, and to do so as efficiently as possible. But what is ideal for one agent may be undesirable for another, and vice versa. As a result, the social system should strike the right balance: It should restrict the allowable behaviors of the various agents enough to serve the different agents

in a good manner. (What *good* here means will depend on the application.) We refer to the problem of finding such a compromise as the *golden mean problem* in artificial social systems. The golden mean problem as described here applies directly to the context of dependent automata discussed in Section 2. We remark that in a given scenario, in which we are given a model of multi-agent activity (e.g., a particular dependent automata) and a notion of what a good balance between the needs of different agents is, there need not be a unique social system that is good. Many acceptable social systems will usually exist for the given scenario.

In summary, practically any effective social system must come to grips with the golden mean problem in one way or another. In fact, we can view the design of a social system as consisting of searching for a reasonable solution to the golden mean problem. In particular, the computational problems we defined and investigated in the previous section are a collection of golden mean problems in the context of the dependent automata model. In these computational problems we assumed that all the goals are considered social, but other assumptions can be treated similarly. The difference between the golden mean problems we investigated in items (a)–(d) in Section 2 is in the definition of what we consider to be a good solution. Our results in Section 2 can therefore be interpreted as a study of the golden mean problem in the framework of a basic model for multi-agent activity.

We now consider a variant of the golden mean problem in another basic model, which we call a *one-shot social game*. The model we present is somewhat simplified, for purposes of exposition. Our aim is to see how the golden-mean problem is captured in a general game-theoretic (i.e. strategic) setting. We start with a set $\mathcal{S}$ of possible physical strategies, identical for all agents. We also assume a set G_{soc} of socially acceptable goals. With each goal $g \in G_{soc}$ and agent i we associate a payoff function $u_g(i)$ that assigns to each joint strategy in $\mathcal{S}^n$ a value between 0 and 1. In the formulation of the golden mean problem below we assume that the social restrictions on the strategies are similar for all agents. In addition, we assume that the value of a payoff function for an agent depends only on its current goal and the strategies executed. Hence, we can refer w.l.o.g. only to the payoff functions of the first agent, and drop the agent's number from the notation of a payoff function. Given an "efficiency parameter" $0 \leq \epsilon \leq 1$ we can now formulate the following problem:

Definition 3.1: [Basic Golden Mean] Let $n \geq 2$ be a constant. Given a set of n agents, a set $\mathcal{S}$ of possible physical strategies, a set G_{soc} of socially acceptable goals, and for each $g \in G_{soc}$ a payoff function $u_g : \mathcal{S}^n \longrightarrow [0,1]$, find a set $\bar{\mathcal{S}} \subseteq \mathcal{S}$ of "socially acceptable" strategies such that for all $g \in G_{soc}$ there exists $s \in \bar{\mathcal{S}}$ such that $u_g(s, \sigma) \geq \epsilon$ for all $\sigma \in \bar{\mathcal{S}}^{n-1}$.

This definition formalizes a particular variant of the golden mean problem in general game-theoretic terms.[3] Nevertheless, this definition captures the main issue involved: In

[3]Other variants would be formalized in a similar fashion. The case where we have a one-shot game of

solving a golden mean problem, the designer needs to disallow some of the possible strategies in order to ensure efficient achievement of certain goals, while on the other hand it is necessary to maintain enough strategies in order that agents can attain their goals in a reasonably efficient manner. Corresponding to the definition of the basic golden mean problem is a natural computational decision problem: Given a set $\mathcal{S}$ of possible physical strategies, the agents' payoff functions and a parameter ϵ, determine whether there exists a set $\bar{\mathcal{S}}$ that will satisfy the basic golden mean problem. In making this precise, one has to make certain assumptions on the number of strategies, the size of G_{soc}, how the strategies are presented to us, and how the utilities are computed. Under what are essentially very weak assumptions,[4] we can prove the following Theorem.

Theorem 3.2: *The decision problem corresponding to a basic golden mean problem is NP-complete (in the number of strategies and goals). If the number of goals is bounded by a constant, then the problem is polynomial.*

As in the case of Theorem 2.1, we can view the NP-completeness result here as a positive indication. The fact that the problem is in NP suggests that an off-line trial and error procedure for determining the social restrictions may be feasible.

Notice that we have been discussing the golden mean problem mainly in the context of the (off-line) design stage of an artificial social system. However, in a precise sense, instances of this problem need to be solved repeatedly to resolve conflicts between different agents' intended actions. In fact, in sufficiently rich contexts it seems crucial for agents to occasionally attempt to resolve such conflicts, thereby essentially solving a local golden mean problem. One thing that Theorem 3.2 implies is that if agents can reach arbitrary states of conflict, then there will be cases in which it will be computationally intractable for them to negotiate a compromise. This can be taken as further evidence for the importance of the design stage, or what we have been calling artificial social systems. One of the roles of the design would be to simplify the world so that, to the extent possible, agents do not reach unresolvable conflicts during the course of their activities. We further discuss this in Section 3.1. The second part of Theorem 3.2, referring to the case where the number of goals is bounded by a constant, is somewhat less important for the issues discussed in this paper. However, it motivated a general heuristic for design that we discuss in Section 3.2.

3.1 Off-line vs. on-line

We think of the design of a social law as a chiefly *off-line* activity; it is performed by the system designers before the agents start their actual activity. The agents' actions and plans are ultimately performed *on-line* in the multi-agent system. Namely, an agent

only two agents with only one goal, where the agents may have different local states, coincide with the computational problem discussed in [26].

[4] Details can be found in the Appendix.

must ultimately plan a course of action and carry it out with rather stringent constraints on the time, resources, and information available. Indeed, a sometimes crucial aspect of on-line activity is that the resources the agent is able to apply in deciding on a course of action may be extremely limited. The system designers, on the other hand, will often have access to considerably greater resources for the purpose of designing the system. The advantages of investing in the off-line design stage are threefold. First, since the designer's off-line resources are usually greater than the agents' on-line resources, some problems may be better solved by the designer than they would be solved on-line by the agents. For example, solving, or finding an approximate solution, to an NP-hard problem may be feasible in an off-line setting, while it may often be hopeless to handle on-line. The second, and perhaps more important, advantage of investing in the design stage is that an off-line design of a good social law will keep the agents from arriving at many of the conflicts to begin with. This can result in far more effective and efficient on-line activity by the agents. Finally, in many cases the design of a social system may be performed together with the design of other aspects of the multi-agent environment. In cases in which an effective social system is hard to come by, the environment may be modified in a manner that will simplify the problem of devising the social law. Examples are adding traffic lights, or changing the road system in various ways. Naturally, such operations are much harder to implement on-line than they are at the off-line design stage.

We have been focusing on how the design of an effective social system can allow agents to act individually in pursuit of their goals, thereby reducing, and in some fortunate cases perhaps even eliminating completely, the need for agents to communicate and explicitly coordinate their actions. In sufficiently complex situations, however, some communication and explicit coordination between the agents is inevitable, and in others it may be desirable. Indeed, a central concern in distributed and decentralized AI is the design of communication and interaction protocols [2, 14]. Our framework applies equally well to such situations. For agents in a multi-agent setting to communicate, they need to have a common language, specific protocols for interaction, and conventions for when and how these are used. A good social system will choose these so that the communication is efficient and effective. Again, by making the right choices in the course of the off-line design of a good social system, we may be able to improve the on-line behavior of the agents and increase their benefits.

3.2 Social routines: A heuristic for design

Imagine that there is a fixed number of basic tasks that an agent needs to be able to perform successfully in order to attain its goals. These may, for example, be the basic operators used in the agent's planning program (e.g., `go_from`(A)`_to`(B) where A and B are neighboring locations, or `put_down`(T), where the agent is currently holding T). This is the type of context in which high-level planning is normally studied [1].

In many cases, the set of basic tasks is rather small, while the class of behaviors that the agents can generate using them is rich and complex. We call the implementations of

these tasks *primitive routines*. We say that a set of such routines is *social* if an agent following a primitive routine in the set is guaranteed to successfully performed its desired task, so long as the other agents behave only according to the routines in this set. Given a set of basic tasks, an implementation of them by a social set of primitive routines can provide the agents with a simple and effective social system. For an example of a social primitive routine, consider the task of filling a glass with water. Normally, this may be implemented by first going to the sink, and then filling the glass. A social implementation, however, could consist of first entering the queue of agents waiting for the sink, and using the sink when the agent's turn comes. In this case, the social implementation guarantees that anyone who wants to use the sink will eventually be able to do so, while in its absence, weak or slow agents might be unable to fill their glasses on a bad day.

Given a small number of basic tasks, a social law can thus be reduced to requiring the agents to use only routines from a prescribed set of carefully planned primitive social routines. The task of verifying that a set of primitive routines is social is likely to be manageable. For example, one can associate the fact that we have a small number of basic tasks (for which we have to find corresponding social primitive routines) with the restriction of the number of primitive goals in the golden-mean problem to be bounded by a small constant. As we demonstrated, this case is computationally tractable. Moreover, an agent provided with such routines is spared the cost of verifying that a plan the agent devises is social. As long as the agent uses only social primitive routines, the overall plan is guaranteed to be social. An obvious setting that can be viewed as using social routines for effective overall behavior is driving: Traffic laws concentrate on drivers' behaviors in a small set of specific types of situations, having to do with intersections, passing, and similar issues. Similarly, in the three examples from the introduction, the use of social primitive routines, at least in a large part of the activity, can greatly simplify the task of coordinating compatible behavior. In particular, in the case of programmers working on a software project, the fact that the verification problem is reduced considerably once social primitive routines are used is of paramount importance. In the typical case, it is extremely difficult to verify that a program consisting of many processes working concurrently actually does what it is supposed to do.

3.3 Social systems and conflict resolution

Today more than ever before people face the problem of designing artificial environments. As illustrated in the three example contexts presented in the introduction, the agents operating in these environments may be robots, they may be people, and they may even be a heterogeneous collection of people, robots, and computers. The design of such environments is generally a very difficult problem.

We can consider the fundamental problem of the field of *distributed/decentralized artificial intelligence* (DAI/DzAI) [3, 7, 5] to be how to design artificial agents and environments for them to operate in. For such a design to be good, it should allow the agents to fruitfully

coexist and effectively function to obtain particular goals of interest. A substantial amount of explicit and formal work in distributed/decentralized artificial intelligence has gone into questions such as:

- How should artificial agents (robots or computer programs) negotiate?
- How should they strike deals with each other?
- What are good schemes for resolving conflicts among artificial agents?
- How do answers to such questions affect the structure of the agents?

We think of an artificial social system as a set of conventions and rules restricting the behavior of the agents. A major purpose of these conventions is, of course, to keep the agents from reaching conflicts to begin with, wherever possible. But avoiding conflicts by using an appropriate social system is not always attainable, since it is not always possible to consider all possible scenarios in advance. Moreover, in some cases the cost of avoiding conflicts of a particular type may be higher than the cost of resolving them once they occur. A comprehensive social system must therefore also contain a component that describes how conflicts are to be handled once they occur. The work on negotiations [18], deals [30], consensus [27], interaction protocols (e.g., [2]), as well as many other forms of conflict resolution can be viewed as handling this very delicate and complex aspect of the design of a social system.

4 Logical Reasoning about Social Systems

The previous sections introduced and investigated basic issues in artificial social systems and their design. In particular, we used an automata-theoretic and game-theoretic models in order to introduce and investigate computational aspects of artificial social systems. In this section we add another tool for reasoning about social systems. Namely, we present a general logical framework for reasoning about social systems. This will enable to supply a general semantics for artificial social systems, and will enable formal logical reasoning about the elements of social systems.

4.1 The semantics of Artificial Social Systems

As argued in [24], providing a clear semantics for artificial social systems is a necessary step in their design. In particular, it will enable formal logical reasoning about social systems. In this section we gradually construct a model for an artificial social system. We begin by defining a general multi-agent system:

Definition 4.1: A *multi-agent system* is a tuple $S = \langle N, W, \mathcal{K}_1, \ldots, \mathcal{K}_n, \mathcal{A}, \text{Able}_1, \ldots, \text{Able}_n, \mathcal{I}, T \rangle$, where:

- $N = \{1, \ldots, n\}$ is a set of *agents*;
- W is a set of *possible worlds*;
- $\mathcal{K}_i \subseteq W \times W$ are *accessibility relations* (we assume that $\mathcal{K}_i$ is an equivalence relation for all $i \in N$);
- $\mathcal{A}$ is a set of *primitive individual actions*;
- $\text{Able}_i : W \longrightarrow 2^{\mathcal{A}}$ is a function that determines the *possible physical actions* for agent i (in any given world).
- $\mathcal{I}$ is a set of possible *external inputs* for the agents.
- $T : W \times (A \times \mathcal{I})^n \longrightarrow W \cup \{-\}$ is a *state transition function.* This function determines what the next state of the world is going to be, as a function of the actions that each agent performs, and the input each agent receives in the current world. $T(w, \overline{(a, I)}) = -$ iff there exists an action a_i in $\overline{(a, I)}$ such that $a_i \notin \text{Able}_i(w)$.

The structure of the set W of possible worlds, and of the possible worlds themselves will be vitally important in any implementation of a multi-agent system. Roughly speaking, we are thinking of these along the lines of the situated automata literature [31], and the related work on knowledge in distributed systems [16]. The $\mathcal{K}_i$ relations are intended to capture the agents' knowledge. The possible external inputs $\mathcal{I}$ are intended to capture messages an agent may receive from outside the system. In particular, we can model dynamic receipt of goals by an agent, by having the agent receive external inputs sent by its master specifying new goals to pursue. Notice that the transition from one world to the next, specified by the function T, depends on the "joint action" consisting of the actions performed by all the agents. Thus, the action an agent performs will usually not uniquely determine the change the world will undergo. There will be many worlds that may potentially result from a given agent performing a specific action.

Within the context of such a multi-agent system, we define a *strategy* or *plan* for agent i to be a function $Ch_i : W \longrightarrow A$ that satisfies:

1. If $(w, w') \in \mathcal{K}_i$, then $Ch_i(w) = Ch_i(w')$
2. $Ch_i(w) \in \text{Able}_i(w)$ for all $w \in W$.

$Ch_i(w)$ is intended to represent the action that agent i *chooses* to perform (according to the plan) when in w. The first condition here requires that this action depend only on i's knowledge; i should choose the same action in worlds it can't distinguish from one another.

The second condition captures the idea that the action chosen must be physically possible for the agent to perform.

We identify an agent's *goal* g within a multi-agent system with a set $W_g \subseteq W$ of worlds. Intuitively, these are the worlds in which the goal has been achieved. Given our formal model, we say that *in world* w, i *has a plan for attaining* g if i has a plan that, starting in world w is guaranteed to yield a world in W_g.

Intuitively, we can think of the Able_i functions as corresponding to a "physical law", since they specify what actions the agents are physically capable of performing. A first step in extending our model to incorporate a social law is to define a normative system, which further restricts agents' actions:

Definition 4.2:

A *normative system* extending a multi-agent system S is defined to be the pair $\mathcal{N} = \langle S, \{\text{Legal}_i\}_{i \leq n} \rangle$, where $\text{Legal}_i : W \longrightarrow 2^{\mathcal{A}}$. Moreover, the functions Legal_i are required to satisfy the following three conditions:

1. (epistemological adequacy): $\text{Legal}_i(w) = \text{Legal}_i(w')$ for all $(w, w') \in \mathcal{K}_i$;
2. (physical adequacy): $\text{Legal}_i(w) \subseteq \text{Able}_i(w)$ for all i and w;
3. (non-triviality): $\text{Legal}_i(w) \neq \emptyset$ for all i and w; .

Intuitively, $\text{Legal}_i(w)$ specifies what actions agent i is *allowed* to perform in w, according to the underlying normative code. Under this interpretation, the epistemological adequacy requirement states that each agent will always know what actions it is allowed to perform. The physical adequacy requirement says that the actions the agent is allowed (or required) to perform are physically enabled. Finally, the nontriviality condition requires that an agent should always have some action it is allowed to perform. Implicitly, we are assuming that a null action, corresponding to "doing nothing", should be an explicit action. The reason for this is that there may be cases in which a normative system requires the agent to perform some active action (e.g., put out a fire). This is achieved by having $\text{Legal}_i(w)$ not contain a null action.

Given a normative system $\mathcal{N} = \langle S, \{\text{Legal}_i\}_{i \leq n} \rangle$, we say that a strategy, or plan, is *legal* with respect to $\mathcal{N}$, if in addition to being a valid strategy in S, all chosen actions are always legal actions. In other words, Condition 2 from the definition of a strategy is strengthened to: 2. $Ch_i(w) \in \text{Legal}_i(w)$ for all $w \in W$.

Our intention is to define a social system. Clearly, a social system is a particular type of a normative system. However, a social system has some additional structure. There is nothing inherent in the structure of a normative system that will guarantee that nothing bad ever happens. We will, in fact, ask even more than this from a social system. We will also require that agents should always be able to attain any "reasonable" or "socially

acceptable" goal. Thus, while the social system might disallow an agent to eat all of the Birthday cake leaving nothing for the others, it will make it possible for any agent that wants to eat a piece of the cake to do so. We capture these ideas as follows. In coming to design a social system, we start out with a set W_{soc} of "socially acceptable" worlds. Intuitively, the social system will be required to guarantee that the state of the world will never exit this set, so long as the agents obey the rules of the social system. In addition, we have a set G_{soc} of "reasonable", or "socially acceptable" goals, which an agent should always be able to attain. Given a multi agent system S, let us denote by W_0 the set of states that the world may be in "initially". We will assume for simplicity that $W_0 \subseteq W_{soc}$. Given a normative system extending S, we say that a world is *legally reachable* if it is reachable from a world in W_0 by a sequence of steps in each of which all of the agents act according to the rules of the normative system.

Formally, a social system for S consistent with W_{soc} and G_{soc} will be a normative system extending S that satisfies:

1. A world $w \in W$ is legally reachable only if $w \in W_{soc}$.
2. For every legally reachable world w, if the goal of agent i in w is $g \in G_{soc}$, then there is a legal plan for i that, starting in w, will attain g so long as all other agents act according to the normative (social) system.

Notice that W_{soc} and G_{soc} are, as stated, not necessary orthogonal to each other. In fact, given the level of abstraction at which we treat our states of the world (we haven't made any restrictions on what they can model), one could technically do away with either W_{soc} or G_{soc}, and make do solely with the other. However, we introduced both because we view each of them as serving to specify distinct aspects of the system. W_{soc} is intended to capture more global aspects of the behavior, perhaps more of the so-called "safety" and "fairness" aspects of the system. The purpose of G_{soc} is to guarantee that agents be able to act in a somewhat useful way; this would very roughly correspond to "liveness" in formal specification of systems.

The sets W_{soc} and G_{soc} are used by the designer in the process of designing the social system. Based on these sets, she is able to incrementally construct the rules of the system, and check them for suitability. In practice, we expect that in some cases the design stage will include some updating of the W_{soc} and G_{soc} sets, as experience is gained and the designer becomes better acquainted with the environment the agents are to operate in and what is reasonable to expect there. Once the design stage is over, we are given an appropriate social system. As remarked above, this is a particular instance of a normative system. This system will be used in two later stages of the process: By the *manufacturer* of agents, that are to act according to this social system, and by the agents themselves once they are in operation. The manufacturer will need to reason about whether his product will act in accordance to the rules of the system, while the agent will need to reason about the world, its own actions, and the actions of others, in the course of planning and acting

in the actual environment. In both cases, the set W_{soc} will no longer play a central role, and the reasoning will be performed with respect of the set of legally reachable worlds. (In fact, after the design stage, we can redefine W_{soc} to be the set of legally reachable worlds.)

The above definition of a social system captures the basic insight behind the methodology of artificial social systems. The definition can be extended in various ways. One extension consists of explicitly modeling the utilities that agents obtain from attaining goals via different routes. For example, an agent may want to get to the airport, and would prefer to be driven there over taking the bus. Thus, the same goal (getting to the airport) can be attained via routes yielding different utilities. Formally, utilities are added to a social system $\langle S, \{\text{Legal}_i\}_{i \leq n}, G_{soc}, W_{soc} \rangle$ by adding a function $u_{a,g} : W_g \rightarrow [0,1]$ for every agent a and goal $g \in G_{soc}$. Intuitively, the value of $u_{a,g}(w)$ represents how pleased a is about the state of attainment of g in the world w.

4.2 Reasoning about social systems

One of the main benefits of having a semantic definition of artificial social systems is the ability to reason about such systems. This reasoning can be performed by the designer, when evaluating the impact of adding or deleting various laws from the system. The manufacturer of agents (e.g., robots) that are to function in the social system will need to reason about whether its creation will indeed be equipped with the hardware and software necessary to follow the rules. Finally, agents within the system can reason about the state of the world and about what they and other agents need to do, based on their observations of the environment and on the social system.

In order to be able to reason formally, we need to decide on a language. For simplicity of exposition, we will choose a propositional language. The basic formulas will consist of a set Φ of primitive propositions, including distinguished atoms *social* and *legal*, corresponding to the statements that the world is social and that the world is legally reachable, respectively. In addition, we have the following facts dealing with the ability of agents to perform actions in a given world: For every agent $i \in N$ and action $a \in A$, $Pos_p(i,a)$, $Nec_p(i,a)$, $Pos_s(i,a)$, and $Nec_s(i,a)$ are formulas (read respectively as: a is *physically possible* for agent i, a is *physically necessary* for agent i, a is *socially possible* for agent i, and a is *socially necessary* for agent i). Later on we will also add basic formulas that deal with agents' goals and their attainability. We close the basic formulas under negation and conjunction, as well as under knowledge operators K_i for $i = 1, \ldots n$. The knowledge operators K_i will capture knowledge with respect to the physical multi-agent system. Much of the agents' planning and reasoning, however, will be based on the assumption that all agents are acting socially, or according to the rules. For this we will add an operator B_i^s for every agent i, intended to capture his beliefs generated by the assumption that the world is legally reachable.

A model for this language will be a pair $\mathcal{M} = (S, \pi)$, where S is a social system, and $\pi : \Phi \longrightarrow 2^W$ is a function associating with every primitive proposition the set of worlds

in which it holds. We now define when a formula φ is satisfied in a world w of $\mathcal{M}$, which we denote by $\langle \mathcal{M}, w \rangle \models \varphi$. The definition is by induction on the structure of φ.

(a) $\langle \mathcal{M}, w \rangle \models \varphi$ (for $\varphi \in \Phi$) if $w \in \pi(\varphi)$.

(b) $\langle \mathcal{M}, w \rangle \models \mathit{social}$ if $w \in W_{soc}$.

(c) $\langle \mathcal{M}, w \rangle \models \mathit{legal}$ if w is legally reachable.

(d) $\langle \mathcal{M}, w \rangle \models Pos_p(i, a)$ if $a \in Able_i(w)$.

(e) $\langle \mathcal{M}, w \rangle \models Nec_p(i, a)$ if $Able_i(w) = \{a\}$.

(f) $\langle \mathcal{M}, w \rangle \models Pos_s(i, a)$ if $a \in \mathrm{Legal}_i(w)$.

(g) $\langle \mathcal{M}, w \rangle \models Nec_s(i, a)$ if $\mathrm{Legal}_i(w) = \{a\}$.

(h) $\langle \mathcal{M}, w \rangle \models \neg\varphi$ if $\langle \mathcal{M}, w \rangle \not\models \varphi$.

(i) $\langle \mathcal{M}, w \rangle \models \varphi \wedge \psi$ if $\langle \mathcal{M}, w \rangle \models \varphi$ and $\langle \mathcal{M}, w \rangle \models \psi$.

(j) $\langle \mathcal{M}, w \rangle \models K_i\varphi$ if $\langle \mathcal{M}, w' \rangle \models \varphi$ for every w' satisfying $(w, w') \in \mathcal{K}_i$.

(k) $\langle \mathcal{M}, w \rangle \models B_i^s\varphi$ if $\langle \mathcal{M}, w \rangle \models K_i(\mathit{legal} \Rightarrow \varphi) \wedge \neg K_i \neg \mathit{legal}$.

The definition of the social belief operator B_i^s in clause (k) is an instance of "belief as defeasible knowledge", as defined in [23]. While B_i^s is a notion of belief, in that $B_i^s\varphi$ may hold when φ does not, the definition of B_i^s in clause (k) gives us a rigorous semantic handle on when facts are believed, and when they are not.

We remark that our choice of having only single actions as the subject of our formulas is not sufficiently general to express all of the facts about possibility and necessity that are encoded in the Able_i and the Legal_i functions of $\mathcal{M}$. We made this choice because our discussion will only involve statements of the simpler type. Extending the language to talk about sets of actions can be done in a straightforward way.

We say that a formula φ is *valid in* $\mathcal{M}$, denoted by $\mathcal{M} \models \varphi$, if $\langle \mathcal{M}, w \rangle \models \varphi$ for all worlds $w \in W$. The formula φ is *valid*, denoted $\models \varphi$, if it is valid in $\mathcal{M}$ for all models $\mathcal{M}$. Satisfiability is defined based on validity in the standard fashion. Given our choice of syntax and its semantics, we can now study what the valid formulas are. Clearly, there are some obvious validities, such as the axioms of propositional logic and the modal system S5 for the knowledge operators. In addition, the fact that $\mathrm{Able}_i(w) \supseteq \mathrm{Legal}_i(w) \neq \emptyset$ induces a particular relationship between the various necessity and possibility formulas. Another property of our formalism is that we assume that an agent performs a single action in every world. As a result, if an action a is socially necessary at a given point, then every other action b is not socially possible there. More instructive is the relationship between knowledge and social actions in our models. The key facts are that $\models Nec_s(i, a) \Rightarrow K_i Nec_s(i, a)$ and $\models Pos_s(i, a) \Rightarrow K_i Pos_s(i, a)$. Let us now consider a number of valid formulas that illustrate the power of our framework.

Proposition 4.3:

The following are valid formulas in our language:

1. $\models B_i^s(\varphi \vee Nec_s(i,a)) \Rightarrow (B_i^s\varphi \vee B_i^s Nec_s(i,a))$
2. $\models \neg B_i^s \neg Nec_s(i,a) \Rightarrow B_i^s Nec_s(i,a) \vee K_i(\neg legal)$
3. $\models B_i^s[(\varphi \Rightarrow Nec_s(i,a)) \wedge (\neg\varphi \Rightarrow \neg Pos_s(i,a))] \Rightarrow [B_i^s\varphi \vee B_i^s\neg\varphi]$

This proposition illustrates the relationship between social necessity and social belief. The first clause says that if an agent believes that either φ holds or it must perform the action a, then the agent must explicitly believe one of these facts: It either believes φ or believes that it must perform the action a. The second clause is self explanatory. The third clause says that if a fact φ determines whether or not the agent is allowed to perform the action a in the current world, then the agent must either explicitly believe φ or it must explicitly believe its negation. Properties such as those presented in this proposition tell us something about the structure of the Legal_i functions. These properties will guide the designer in constructing these functions (restricting the agents' actions). In addition, these properties can be used by the agents in reasoning about other agents' knowledge and in learning about the world by observing other agents' actions.

Goals and action in a social system

A main purpose of an artificial social system is to provide a framework in which the agents will be able to plan, act, and thereby manage to satisfy their goals. The reasoning presented above did not include issues related to agents' goals, such as satisfaction of goals, etc. We now extend the language to allow such reasoning. For simplicity, we will assume that in any given world an agent may have at most one distinguished *current goal*. The identity of this goal may change dynamically over time as a result of the agent receiving input from an external source, or as a result of the current goal being satisfied, or perhaps even by the agent interacting with other agents. In any case, the basic idea is that an agent actively pursues its current goal at any given point in time. To reason about such goals, we add the formulas of the form $current\_goal(i,g)$ to the language, for every agent i and goal g. Of course, $current\_goal(i,g)$ will hold whenever g is agent i's current goal.

Recall that we have associated with every goal g a set W_g of the worlds at which g is satisfied. In this sense, a goal can be thought of as a proposition. Satisfying a goal then coincides with satisfying the corresponding proposition. We will therefore treat goals as a special case of propositions and formulas. In order to reason about satisfaction of goals, we want to be able to talk about when a set T of agents can cause a fact φ to be satisfied. Here we are mainly interested in *social reachability*, by which we mean that the agents in T have a (joint) plan consisting solely of socially acceptable actions that is guaranteed to attain φ, so long as all other agents follow the rules of the social system. We denote this by

s-reachable(T, φ). In analogy to social reachability we also define *physical reachability* (in this case we consider all physically possible actions and plans). The corresponding notation in this case will be *p-reachable*(T, φ). We would also like to be able to reason about what will happen if a certain action will *actually* be executed. In order to do so we add appropriate parameters to the reachability operators. We will write *p-reachable*$(T, \varphi, do_i(a))$ if *p-reachable*(T, φ) holds in cases where agent i executes action a in the current world. We can similarly define the parameter to be any element in the closure of the $do_i(a)$'s under conjunction and negation. Similar parameters can be used in the *s-reachable* operator.

We can now formulate the two conditions in the definition of a social system in terms of *s-reachable*:

1. $\models$ *legal* $\Rightarrow \neg$*s-reachable*$(T, \neg social)$ for all sets $T \subseteq \{1, \ldots, n\}$ of agents.
2. $\models$ *legal* $\wedge$ *current_goal*$(i, g) \Rightarrow$ *s-reachable*(i, g) for every agent i and goal $g \in G_{soc}$.

Given the above formalism, the designer of the system and its users can reason about actions, goals and their achievement. For example, imagine that in a certain socially acceptable situation Alice needs to move to the other side of a door in order to reach a certain socially acceptable goal g, but she can do so only if Bob will first open the door. In this case, our designer will deduce that the social system must order Bob to open the door. Less straightforward is the following type of reasoning: Assume that David asks Bob whether Bob believes that Alice's goal is to achieve g. Then if Bob believes that Alice cannot attain g unless he opens the door, and he is not forced to open it, then Bob can deduce that g is not Alice's current goal. The following proposition demonstrates that such types of reasoning are supported by our formalism.

Proposition 4.4:

The following are valid formulas in our language:

$\models B_i^s[\neg$*p-reachable*$(N, g, \neg do_i(a))] \Rightarrow [B_i^s(\neg$*current_goal*$(j, g)) \vee B_i^s(Nec_s(i, a))]$

$\models [\text{legal} \wedge \neg$*p-reachable*$(N, \text{social}, do_i(a))] \Rightarrow \neg Pos_s(i, a)$

Notice that the first part of the proposition can be considered as a formalization of Bob's reasoning in the previous example. The second part captures potential reasoning of the designer when disallowing some of the actions by the social law: if in a social situation agent i has an action that will necessary lead to an asocial situation, then the designer has to conclude that this action must be socially impossible.

High level social laws

In our semantic model we have been treating the social law as a restriction on the Able$_i$ functions, or on the actions that the agents can perform in a given world. However, in

general we expect the social law to be stated in terms of some high-level formal language. We now show how it is possible to bridge the gap between the two, by giving examples of definitions of high-level rules in terms of the language defined above. For example, imagine that we want to have a rule that states that whenever the circumstances satisfy a fact φ (say, i's house is on fire), then all members of set A must help i to attain ψ (say, put out the fire). We say that $\mathcal{M}$ *enforces the rule* should_help$_{A,i}(\varphi, \psi)$ iff

$$\mathcal{M} \models \varphi \wedge \textit{s-reachable}_{A \cup \{i\}} \psi \Rightarrow \textit{s-reachable}_i \psi.$$

Another typical rule may be that whenever the question arises, i must prefer attaining ψ over attaining θ. As before, we now say that $\mathcal{M}$ *enforces* should_prefer$_i(\psi, \theta)$ iff
$\mathcal{M} \models [\textit{s-reachable}_i(\psi) \wedge \neg \textit{s-reachable}_i(\theta \wedge \textit{s-reachable}_i(\psi)) \wedge$
$\neg \textit{s-reachable}_i(\psi \wedge \textit{s-reachable}_i \theta)] \Rightarrow \quad \neg \textit{s-reachable}_i \theta.$

We may, for example, wish to have i notify j whenever i believes that φ holds. This can be formalized by: The system $\mathcal{M}$ is said to enforce should_notify$_i, j(\varphi)$ iff

$$\text{should\_help}_{i,j}(B_i^s \varphi, B_j^s B_i^s \varphi).$$

Of course, we could go on with a long list of rules now. Moreover, some of our definitions may be modified slightly to capture distinct senses of these and other terms. We hope that the reader is convinced by now that our semantic definitions and basic propositional language provide us with means to express high-level rules in a rigorous and concise fashion.

5 Conclusions and related work

There is a significant body of literature concerned with issues related to topics discussed in this paper. This includes work in the areas of organization theory (see [20],[12],[20]), team theory ([21]), and DAI (see for example [11]). The related work in these areas of research is especially concerned with the design of agents' roles and communication structures that enable cooperative achievement of a common goal. Our work on artificial social systems concentrates on a somewhat complementary issue: the off-line design and computation of laws that will enable each agent to work individually and successfully towards its own goals during the on-line activity, provided that the other agents obey these laws. Additional related work includes the synthesis of multi-agent programs ([28]) and work on cooperative discrete event systems (DES) ([29]). The artificial social systems approach to design talks about two essential stages in the design process. First, general rules governing the behavior of agents are given (this is the "social law"), and then the behavior of each agent will be determined, either by the designer or by the agent, in a fashion consistent with these rules. This two-stage process can be used as a methodology for the design of discrete event systems as well. For further discussion of this connection, see [35].

Society metaphors have been proposed before in the AI literature, albeit in somewhat different contexts. Minsky uses a society metaphor in his work on the society of mind [22].

The notion of *social choice* is an important element in e.g. the work of Jon Doyle [10]. Finally, social metaphors appear also in the works of Fox, Kornfeld and Hewitt, Malone, and Simon ([12], [17], [20], [36]) concerning organization theory. We treat the notion of an artificial social system in a relatively narrow sense, and with a particular point of view in mind. We wish to develop a theory to support the design of multi-agent environments, and to assist the reasoning necessary in creating or modifying agents to comply with a given social system. Our treatment does not attempt to subsume any of the other uses of society metaphors in AI, sociology, or ethology. We find the use of the term *social system* appropriate for our purposes because of the analogy to social order in natural (human and animal) populations. We are specifically interested in the context of loosely-coupled agents following uncorrelated and dynamically changing goals. Our thesis is that a society metaphor has an important role to play in this context, so long as sufficient care is taken in defining, studying, and applying social systems to multi-agent activity.

Much work has been devoted to an explicit and formal study of the centralized approach and of the on-line resolution of conflicts (e.g., [19], [15], [37], [30],[18]). Our work is the first to discuss explicitly and formally the computational mechanism that applies for non-centralized intermediate solutions.

In spite of the generality of our work, we wish to emphasize that our work in no way diminishs the crucial importance of mechanisms for interaction and communication among agents, nor the significance of the study of effective representations for agents. Some of the work developed in LIFIA [2, 9] provides considerable progress in these complementary directions. The discussion of various ways of modeling agents is of significant importance to the coordination between agents. Further study of artificial social systems may need to take various representation levels into account while addressing the construction of useful social laws. Indeed, our discussion on high-level social laws is in the spirit of the discussion on bridging the gap between intentional and reactive agents in [9]. Moreover, as we explained in Section 3.3, elaborated communication and negotiation mechanisms (as discussed in [2]) may serve as essential components of a social law.

Our work does not take into account the incentives agents have for cooperation. Some of the work in CNR [6, 4] has been concerned with issues such as defining goal adoption as a basic form of cooperation, and with the effect of social power on cooperation among agents. Although some of our technical machinery may enable to express concepts such as goal adoption, our emphasis is on the design of artificial systems where agents are assumed to conform to every law prescribed by the system's designer. Our work bridges the gap between centralized and decentralized approaches to coordination where agents are law-abiding agents. Extending work on Artificial Social Systems to include a treatment of cooperation incentives is one of the most challenging directions for future work.

A particular case study of the design of a social law is presented in [34]. There, the authors investigate traffic laws for mobile robots that operate on an n by n grid. They present nontrivial laws that allow the robots to carry out respective tasks without collision at a rate that is within a constant of the rate the tasks would take each of them if it had

the whole space to itself. This is an example of how appropriate off-line design of social laws guarantees very effective on-line behavior. In [33] the authors present a novel model that defines multi-agent systems while referring explicitly to the notion of social law. In the framework of this model they investigate the automatic synthesis of useful social laws and give precise conditions under which the problem becomes tractable. The treatment presented in that work, while having additional features of its own, can be considered an extension of the work on social automata presented in Section 2. In addition, that work discusses the conditions under which the problem of algorithmically synthesizing a social law becomes tractable. This work has been extended to non-homogeneous dynamic social structures in [39]. In a complementary work (see [32]) Shoham and Tennenholtz considered the interesting case of conventions and laws that are not determined off-line before the initiation of activity, but rather emerge during the on-line activity of the system. Their research concentrates on understanding how different agent behaviors and system characteristics affect the efficiency of convention evolution. Their work can also be viewed as a (nontrivial) extension of this one: Standards of behavior that are found to be efficient can be used as social laws that will lead to the successful emergence of specific useful conventions.

In most any environment involving many agents, the actions and behaviors of an agent affect and are affected by the actions of others, at least to a certain degree. In such settings, an agent's behavior is invariably somewhat different than what it would be had there been no other agents to consider. As a result, practically every multi-agent environment has a social system of some sort. This system may have been designed *a priori* in a careful fashion, or it may have evolved in various ways. It may be stable or it may change dynamically. In any case, however, we claim that it is there. Moreover, we argue that artificial social systems play a major role in the overall performance of agents in multi-agent systems. As a result, we claim that they deserve to be studied explicitly and formally, and that their role should be considered in the design and implementation of multi-agent systems. The study of artificial social systems we presented suggests a new perspective on multi-agent activity, one that gives rise to new and interesting problems. The last couple of years have seen a considerable amount of research initiated around the idea of artificial social systems.

Appendix: Proof of Theorems

Proof of Theorem 2.1: In order to show that the problem in NP we first observe that any social law can be encoded in polynomial space. Notice that the size of a plan for agent i is bounded by $|L_i| \cdot |A|$, and therefore is polynomial. Given this fact and since there are only polynomially many pairs of initial states and goal states, we get that the desired set of plans (which guarantee reaching from the initial states to the goal states in the restricted system), if one exists, can also be encoded in polynomial space. It remains to show that a verification that the plans indeed guarantee the achievement of the goals from the initial states, given the social law, can be done in polynomial time. This will be

done by a backward chaining procedure in the restricted system (i.e., in the system S^{Σ} where the actions of the agents are restricted by the law Σ). Let p be a plan for agent i that should guarantee reaching from initial state s to a goal state s^g. We consider the configuration space of the (restricted) dependent automata, and initially mark as "good" only the configurations where agent i's state is s^g. The procedure continues in iterations where in each iteration additional configurations are marked as "good", if the action that p selects in them leads to a configuration already marked as "good". The process stops when there are no more configurations that can be marked good. The plan p guarantees reaching from state s to state s^g if and only if all the initial configurations of the system where the state of agent i is s are marked "good" at the end of the above process.

We now prove that the problem is NP-hard by a reduction from 3-SAT [13]. Assume we are given an instance σ of 3-SAT, and let k be the number of clauses in σ. We assume that the dependent automata has two identical agents. For each of them, there is a single initial state s_0, a single failure state *bad*, and k goal states $s_1^g, \ldots, s_k^g$, each associated with a single clause of σ. We define the set A of possible actions to contain an action for pair (c, l), where c is a clause in σ, and l is a literal (or the negation of a literal) such that l is one of the disjuncts in c. Thus, A consists of at most $3k$ actions. We now describe the transition function τ. When the first agent performs the action (c, l) and the second agent performs (c', l') when they are both in the initial state, the following happens. If the actions *conflict*, by which we mean that l is the complement of l' (either $l = \neg l'$ or $l' = \neg l$), then both agents move to their respective *bad* states. Similarly, if $c = c'$ but $l \neq l'$, then again they both move to *bad*. If, however, l and l' are not complements, and if $c = c'$ then $l = l'$, then the first agent moves to the state corresponding to clause c, and the second to the state corresponding to clause c'. All states other than the initial states are sinks; once there, an agent will never move to another state.

We now claim that σ has a satisfying assignment if and only if the DA just constructed has a social law as desired. Assume that π is a satisfying assignment for σ. We define the social law Σ to allow only actions (c, l) where l is true under π. It is obvious that with this law, no agent will ever reach the *bad* state, since the social law guarantees that the agents will never generate conflicting actions. Since π is a satisfying assignment, it follows that for each clause c in σ there is at least one literal $l_c \in c$ that is true under π. It follows that an agent can reach the goal state corresponding to c by performing (c, l_c). It follows that each agent has a plan to reach each of its goal states, and we are done with the only if direction. It remains to show that if a social law of this type exists, then σ is satisfiable. Let Σ be such a social law. Clearly, it does not allow an agent to perform an action (c, l) in the initial state, while the other one performs a conflicting action (c', l'). In addition, for each clause $c \in \sigma$, only one action (c, l) is allowed. It follows that for every literal l such that some action (c, l) is allowed by Σ, no action (c', l') where l' is the complement of l is allowed by Σ. As a result, Σ defines a partial assignment to the literals of σ. Since Σ enables each agent to obtain any goal state from the initial state, it follows that every assignment π that is consistent with the partial assignment induced by Σ is guaranteed to satisfy σ. We conclude that the existence of a social law Σ of the desired type implies the

satisfiability of the 3-SAT instance σ. ∎

Proof of Theorem 2.2: The fact that the problem in NP-hard is proved as in Theorem 2.1. We now prove that the problem is in NP.

We will take a social law to be a set of plans for each agent, where a plan is associated with any pair of initial and goal states of each agent. We will show that we can guess and encode such social laws in polynomial space and verify that they are satisfactory in polynomial time. This will show that the problem is in NP. Notice that we require each plan to succeed no matter what the initial states and goals (and hence plans) of the other agents are. The fact that the plans constitute a social law, makes them common knowledge (although an agent will not know which plans are actually executed by the other agents), which is crucial for the proof of this theorem.

We will prove that the problem is in NP, by showing that if an appropriate plan exists, then it can be replaced by a plan that can be encoded and verified efficiently. Consider a plan p for agent i for reaching from one of its initial states to one of its goal states. The number of actions and observations (i.e. states visited) that might be made along any execution of p is polynomial (by our requirement.) Now, any assignment of initial states and goals for the other agents will correspond to one sequence of polynomial length of observations and actions induced by p. Since there are no more than polynomially many such assignments (of initial and goal states) there are no more than polynomially many such sequences. Combining the above, each plan p corresponds to polynomially many sequences, each of which is of polynomial length. These sequences are equivalent to the plan p and can be encoded in polynomial space. The verification that p achieves the goal is done by simulating the behavior of p (as described by the above concise representation) for any possible initial configuration and any possible behavior of the other agents (there are polynomially many such behaviors and they are again encoded concisely.) Combining the above, we get that the problem is in NP. ∎

Proof of Theorem 3.2: For ease of presentation, we will use the following assumptions:

1. There are only two agents. We refer to the agents as I and II. The case of any other constant number of agents is treated similarly.

2. We assume that both agents have the same set of possible physical strategies. Moreover, there is a given enumeration $\mathcal{S} = s_1, \ldots, s_m$ of the strategies.

3. There are k goals: $g_1, \ldots, g_k$. For each goal g_l there is a corresponding payoff function. Such a payoff function associates with each element $(s_i, s_j) \in \mathcal{S} \times \mathcal{S}$ a number between 0 and 1. This number stands for the payoff of (w.l.o.g) agent I when it has the goal g_l and executes the strategy s_i while agent II executes s_j.

With each goal g_l we associate an $m \times m$ matrix M_l. The value of the (i,j)'th term in M_l will be the payoff for agent I when its goal is g_l and it plays strategy s_i while agent

II plays s_j. Given a subset s of the numbers between 1 and n, we define M_l^s to be the sub-matrix of M_l generated by deleting each row and column whose number appears in s. The golden mean problem turns into: Find s s.t. each matrix M_l^s will satisfy that the maxmin on the its rows is greater or equal to ϵ.

We can now prove theorem 3.2:

The problem is in NP: We guess the strategies that are to be deleted, and then we check that for every goal there is a row in the appropriate sub-matrix that remains after the deletion and contains only 1's. The case where the number of goals is bounded by a constant will turn out to be polynomial, since we have to choose only a constant number of strategies (one for each goal) from the set of strategies. Therefore, there are only polynomially many such selections, each of which can be checked as mentioned above (in polynomial time).

We prove NP-hardness by reduction from SAT. We associate with each clause in the SAT formula a matrix. The i'th row and i'th column of this matrix correspond to the variable x_i if $1 \leq i \leq n$, and to the literal $\neg x_{i-n}$ if $n+1 \leq i \leq 2n$, where $x_1, \ldots, x_n$ are the variables in the appropriate formula. Each entry in the matrix of the form $(i, i+n)$, or $(i+n, i)$ will contain the value 0, and other entries in the matrix will contain a 1 in a certain row if the literal associated with its number appears in the appropriate clause and 0 otherwise. A similar thing is done for columns. Now we take $\epsilon = 1$, and we have a reduction to the golden mean problem. If a golden mean exists, then we can find a satisfying assignment by substituting 1 for any literal that corresponds to a row that is not deleted (if the row that corresponds to a literal and the row that corresponds to its negation are both not deleted then we can w.l.o.g assign 1 to one of them and 0 to the other). Notice that if a golden mean exists then in each sub-matrix created by the appropriate deletion there is a row with all 1's. Thus, we get that the literals corresponding to these rows will satisfy the appropriate clauses. On the other hand, if there is a satisfying assignment, then we will throw the rows and columns that correspond to literals that get the value 0, and will keep the others. It can be easily verified that the above gives us the desired result. ∎

Proof of Proposition 4.3:

1. Assume that $\langle \mathcal{M}, w \rangle \models B_i^s(\varphi \vee Nec_s(i,a))$.

   This implies that $\langle \mathcal{M}, w \rangle \models K_i(\neg legal \vee \varphi \vee Nec_s(i,a)) \wedge \neg K_i \neg legal$.

   We have to show that $\langle \mathcal{M}, w \rangle \models (K_i(\neg legal \vee \varphi) \vee K_i(\neg legal \vee Nec_s(i,a))) \wedge \neg K_i \neg legal$.

   It suffices to show that $\langle \mathcal{M}, w \rangle \models (K_i(\neg legal \vee \varphi) \vee K_i(\neg legal \vee Nec_s(i,a)))$.

   If the above does not hold, then there exist w_1, w_2, which are both indistinguishable from w, such that $\langle \mathcal{M}, w_1 \rangle \models legal \wedge \neg\varphi$, and $\langle \mathcal{M}, w_2 \rangle \models legal \wedge \neg Nec_s(i,a)$.

   However, if $\langle \mathcal{M}, w_2 \rangle \models \neg Nec_s(i,a)$, then $\langle \mathcal{M}, w_1 \rangle \models \neg Nec_s(i,a)$ as well. Therefore, $\langle \mathcal{M}, w_1 \rangle \models legal \wedge \neg\varphi \wedge \neg Nec_s(i,a)$, which contradicts our assumption (about $K_i(\neg legal \vee \varphi \vee Nec_s(i,a))$).

2. Assume that $\langle \mathcal{M}, w \rangle \models \neg B_i^s(\neg Nec_s(i,a))$.

   This implies that $\langle \mathcal{M}, w \rangle \models \neg K_i(\neg legal \vee \neg Nec_s(i,a)) \vee K_i \neg legal$.

   We need to show that $\langle \mathcal{M}, w \rangle \models (K_i(\neg legal \vee Nec_s(i,a)) \wedge \neg K_i \neg legal) \vee K_i \neg legal$.

   It is suffices to show that if we assume that $K_i(\neg legal)$ does not hold in w, and we assume that $\langle \mathcal{M}, w \rangle \models \neg K_i(\neg legal \vee \neg Nec_s(i,a))$, then $\langle \mathcal{M}, w \rangle \models K_i(\neg legal \vee Nec_s(i,a)) \wedge \neg K_i \neg legal$.

   However, if the latter does not kold, then there exists w', which is indistinguishable from w, where $legal \wedge \neg Nec_s(i,a)$ holds. This implies that $K_i(\neg Nec_s(i,a))$ holds, which contradicts our assumption.

3. Assume that $\langle \mathcal{M}, w \rangle \models B_i^s[(\varphi \Rightarrow Nec_s(i,a)) \wedge (\neg\varphi \Rightarrow \neg Pos_s(i,a))]$.

   From the above assumption we get that

   $\langle \mathcal{M}, w \rangle \models B_i^s[(\neg\varphi \wedge \neg Pos_s(i,a)) \vee (\varphi \wedge Nec_s(i,a))]$.

   This yields:

   $\langle \mathcal{M}, w \rangle \models \neg K_i(\neg legal) \wedge K_i[\neg legal \vee (\neg\varphi \wedge \neg Nec_s(i,a)) \vee (\varphi \wedge Nec_s(i,a))]$.

   We have to show: $\langle \mathcal{M}, w \rangle \models [B_i^s\varphi \vee B_i^s\neg\varphi]$.

   This implies that we have to show: $\langle \mathcal{M}, w \rangle \models \neg K_i(\neg legal) \wedge (K_i(legal \Rightarrow \varphi) \vee K_i(legal \Rightarrow \neg\varphi))$.

   It is clear that, given our assumption, $\langle \mathcal{M}, w \rangle \models \neg K_i(\neg legal)$ holds.

   Therefore, in order that the desired result will be obtained we have to show that the following pair of statements contradict our assumption: $\langle \mathcal{M}, w \rangle \not\models \neg K_i(legal \Rightarrow \varphi)$; $\langle \mathcal{M}, w \rangle \not\models \neg K_i(legal \Rightarrow \neg\varphi)$. If the above pair of statements hold, then there exists w', w'' such that $\langle \mathcal{M}, w' \rangle \models legal \wedge \neg\varphi$ and $\langle \mathcal{M}, w'' \rangle \models legal \wedge \varphi$.

   However, if $Nec_s(i,a)$ holds in w, then we get that $legal \wedge \neg\varphi \wedge Nec_s(i,a)$ holds in w'. This contradicts our assumption. If $\neg Nec_s(i,a)$ holds in w, then we get that $legal \wedge \varphi \wedge \neg Nec_s(i,a)$ holds in w'', which contradicts our assumption as well.

   Combining the above gives us the desired result.

∎

Proof of Proposition 4.4:

1. Assume that $\langle \mathcal{M}, w \rangle \models B_i^s[\neg p\text{-}reachable(N, g, \neg do_i(a))]$.

   This implies that $\langle \mathcal{M}, w \rangle \models \neg K_i(\neg legal) \wedge K_i(legal \Rightarrow \neg p\text{-}reachable(N, g, \neg do_i(a))$.

   We have to show that: $\langle \mathcal{M}, w \rangle \models \neg K_i(\neg legal) \wedge K_i(legal \Rightarrow (\neg current\_goal(j, g)) \vee (\neg K_i(\neg legal) \wedge Nec_s(i,a)))$

   Given our assumption, in order that the above will not hold, there should exist w_1, indistinguishable from w, where $legal \wedge current\_goal(j, g) \wedge \neg Nec_s(i,a)$ holds. On

the other hand, since we talk about a social system, we have to require $(\mathit{legal} \wedge \mathit{current\_goal}(j,g)) \Rightarrow \mathit{s\text{-}reachable}(j,g)$. However, our assumption tells us that $\neg \mathit{p\text{-}reachable}(j,g)$ unless i does a, which gives a contradiction, and yields the desired result.

2. Assume that $\langle \mathcal{M}, w \rangle \models [\mathit{legal} \wedge \neg \mathit{p\text{-}reachable}(N, \mathit{social}, \mathit{do}_i(a))]$.

   If $\langle \mathcal{M}, w \rangle \models \mathit{Pos}_s(i,a)$, then the relationships between social reachability and physical reachability, augmented with our assumption, imply that $\langle \mathcal{M}, w \rangle \models \mathit{legal} \wedge \mathit{s\text{-}reachable}(N, \neg \mathit{social})$. This contradicts the assumption that the system is social, and yields the desired result.

∎

Acknowledgements

The second author would like to thank Yoav Shoham for the collaboration and joint work on various aspects of artificial social systems.

References

[1] James Allen, James Hendler, and Austin Tate, editors. *Readings in Planning.* Morgan Kaufmann Publishers, 1990.

[2] S. Berthet, Y. Demazeau, and O. Boissier. Knowling Each Other Better. In *Decentralized AI 2*, pages 23–42, 1992.

[3] A. H. Bond and L. Gasser. *Readings in Distributed Artificial Intelligence.* Ablex Publishing Corporation, 1988.

[4] C. Castelfranchi. Social Power: A Point Missed in Multi-Agent, DAI and HCI. In Y. Demazeau and J.P. Muller, editors, *Decentralized AI*, pages 49–62. North-Holland/Elsevier, 1990.

[5] C. Castelfranchi and E. Werner. *Artificial Social Systems.* From Reactive to Intentional Agents, 1992.

[6] R. Conte, M. Miceli, and C. Castelfranchi. Limites and Levels of Cooperation: Disentangling Various Types of Prosocial Interaction. In Y. Demazeau and J.P. Muller, editors, *Decentralized AI 2*, pages 147–157. North-Holland/Elsevier, 1991.

[7] Y. Demazeau and J.P. Muller. *Decentralized AI.* North Holland/Elsevier, 1990.

[8] Y. Demazeau and J.P. Muller. *Decentralized AI 2.* North Holland/Elsevier, 1991.

[9] Y. Demazeau and J.P. Muller. From Reactive to Intentional Agents. In Y. Demazeau and J.P. Muller, editors, *Decentralized AI 2*, pages 3–10. North-Holland/Elsevier, 1991.

[10] J. Doyle and M.P. Wellman. Impediments to Universal Preference-Based Default Theories. In *Proceedings of the 1st conference on principles of knowledge representation and reasoning*, 1989.

[11] Edmund H. Durfee, Victor R. Lesser, and Daniel D. Corkill. Coherent Cooperation Among Communicating Problem Solvers. *IEEE Transactions on Computers*, 36:1275–1291, 1987.

[12] M. S. Fox. An organizational view of distributed systems. *IEEE Trans. Sys., Man., Cyber.*, 11:70–80, 1981.

[13] M. Garey and D. Johnson. *Computers and Intractability - A Guide to the Theory of NP-completeness.* W.H. Freeman and Company, 1979.

[14] G. Gaspar. Communication and Belief Changes in a Society of Agents: Towards a Formal Model of an Automanous Agent. In Y. Demazeau and J.P. Muller, editors, *Decentralized AI 2*, pages 245–255. North-Holland/Elsevier, 1991.

[15] M. P. Georgeff. Communication and Interaction in Multi-Agent Planning. In *Proc. of AAAI-83*, pages 125–129, 1983.

[16] J. Halpern and Y. Moses. Knowledge and common knowledge in a distributed environment. Technical Report RJ 4421, IBM, 1984.

[17] W. A. Kornfeld and C. E. Hewitt. The scientific community metaphor. *IEEE Trans. Sys., Man., Cyber.*, 11:24–33, 1981.

[18] S. Kraus and J. Wilkenfeld. The Function of Time in Cooperative Negotiations. In *Proc. of AAAI-91*, pages 179–184, 1991.

[19] Amy. L. Lansky. Localized Event-Based Reasoning for Multiagent Domains. Technical Report 423, SRI International, 1988.

[20] T. W. Malone. Modeling Coordination in Organizations and Markets. *Management Science*, 33(10):1317–1332, 1987.

[21] Jacob Marschak and Roy Radner. *Economic Theory of Teams.* Yale University Press, 1972.

[22] M. Minsky. *The Society of Mind.* Simon and Schuster, 1986.

[23] Y. Moses and Y. Shoham. Belief as Defeasible Knowledge. In *Proc. 11th International Joint Conference on Artificial Intelligence*, 1989.

[24] Y. Moses and M. Tennenholtz. Artificial Social Systems Part I: Basic Principles. Technical Report CS90-12, Weizmann Institute, 1990.

[25] Y. Moses and M. Tennenholtz. On Computational Aspects of Artificial Social Systems. In *the Proceedings of the Eleventh Workshop on Distributed Artificial Intelligence*, pages 267–283, 1992.

[26] C.H. Papadimitriou and J. Tsitsiklis. On the Complexity of Designing Distributed Protocols. *Information and Control*, 53(3):211–218, 1982.

[27] M. Pease, R. Shostak, and L. Lamport. Reaching agreement in the presence of faults. *Journal of the ACM*, 27(2):228–234, 1980.

[28] A. Pnueli and R. Rosner. Distributed Reactive Systems are Hard to Synthesize. In *Proc. 31th IEEE Symp. on Foundations of Computer Science*, 1990.

[29] P.G. Ramadge and W.M. Wonham. The Control of Discrete Event Systems. *Proceedings of the IEEE*, 77(1):81–98, January 1989.

[30] J. S. Rosenschein and M. R. Genesereth. Deals Among Rational Agents. In *Proc. 9th International Joint Conference on Artificial Intelligence*, pages 91–99, 1985.

[31] S. J. Rosenschein. Formal Theories of Knowledge in AI and Robotics. *New Generation Computing*, 3(3):345–357, 1985.

[32] Y. Shoham and M. Tennenholtz. Emergent Conventions in Multi-Agent Systems: initial experimental results and observations. In *Proc. of the 3rd International Conference on Principles of Knowledge Representation and Reasoning*, pages 225–231, 1992.

[33] Y. Shoham and M. Tennenholtz. On the Synthesis of Useful Social Laws for Artificial Agent Societies. In *Proc. of AAAI-92*, pages 276–281, 1992.

[34] Y. Shoham and M. Tennenholtz. On Traffic Laws for Mobile Robots. Proc. of the 1st Conference on AI planning systems (AIPS-92), 1992.

[35] Y. Shoham and M. Tennenholtz. Social Laws for Artificial Agent Societies: Off-line Design. *Artificial Inteligence*, 73, 1995.

[36] Herbert. A. Simon. *The Sciences of the Artificial.* The MIT Press, 1981.

[37] C.J. Stuart. An Implementation of a Multi-Agent Plan Synchronizer. In *Proc. 9th International Joint Conference on Artificial Intelligence*, pages 1031–1033, 1985.

[38] M. Tennenholtz. *Efficient Representation and Reasoning in Multi-Agent Systems.* PhD thesis, Weizmann Institute, Israel, 1991.

[39] M. Tennenholtz. On Computational Social Laws for Dynamic Non-Homogeneous Social Structurs. To appear in JETAI, 1994.

[40] E. O. Wilson. *The insect societies*. Harvard University Press, 1971.