A Survey of Multi-Agent Organizational Paradigms *

Bryan Horling and Victor Lesser
Multi-Agent Systems Lab
Department of Computer Science
University of Massachusetts, Amherst, MA
{bhorling,lesser}@cs.umass.edu

Abstract

Many researchers have demonstrated that the organizational design employed by a system can have a significant, quantitative effect on its performance characteristics. A range of organizational strategies have emerged from this line of research, each with different strengths and weaknesses. In this article we present a survey of the major organizational paradigms used in multi-agent systems. These include hierarchies, holarchies, coalitions, teams, congregations, societies, federations, and matrix organizations. We will provide a description of each, discuss their costs and benefits, and provide examples of how they may be instantiated and maintained.

1 Introduction

The organization of a multi-agent system is the collection of roles, relationships, and authority structures which govern its large-scale behavior. All multi-agent systems possess some or all of these characteristics and therefore all have some form organization, although it may be implicit and informal. Just as with human organizations, such agent organizations provide a description of how the members of the population interact with one another, not necessarily on a moment-by-moment basis, but over the potentially long-term course of a particular goal or set of goals. These structures might influence authority relationships, data flow, resource allocation, coordination patterns or any number of other system characteristics [36, 10]. They can help groups of simple agents exhibit complex behaviors, and help sophisticated agents reduce the complexity of their reasoning. Implicit in this concept is the assumption that the organization serves some purpose – that the shape, size and characteristics of the organizational structure can effect the behavior of the system [29]. It has been repeatedly shown that the organization of a system can have significant impact on its short and long-term performance [11, 76, 39, 64, 2, 70, 8], dependent on the characteristics of the agent population, scenario goals and surrounding environment. Because of this, the study of organizational characteristics, generally known as computational organization theory, has received much attention by multi-agent researchers.

It is generally agreed that there is no single type of organization that is suitable for all situations [43, 14, 54, 11]. In some cases, no single organizational style is appropriate for a particular situation, and a number of different organizational structures operating concurrently are needed [30, 40]. Some researchers go so far as to say no perfect organization exists for any situation, due the inevitable trade-offs that must be made and the uncertainty, lack of global coherence and dynamism present in any realistic population [75]. What is clear is that all approaches have different characteristics which may be more suitable for some problems and less suitable for others. Organizations can be used to limit the scope of interactions, provide strength in numbers, reduce or explicitly increase redundancy or formalize high-level goals which no single agent may be aware of. At the same time, organizations can also adversely affect computational or communication overhead, reduce overall flexibility or reactivity, and add an additional layer of complexity to the system [39]. By discovering and evaluating these characteristics, and then encoding them using an explicit representation [26], one can facilitate the process of organizational-self design [13] whereby a system automates the process of selecting and adapting an appropriate

*Effort sponsored in part by the Defense Advanced Research Projects Agency (DARPA) and Air Force Research Laboratory Air Force Materiel Command, USAF, under agreements number F30602-99-2-0525, F30602-03-C-0010 and DOD DABT63-99-1-0004. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright annotation thereon. This material is also based upon work supported by the National Science Foundation under Grant No. IIS-9812755. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the Defense Advanced Research Projects Agency (DARPA), Air Force Research Laboratory or the U.S. Government.

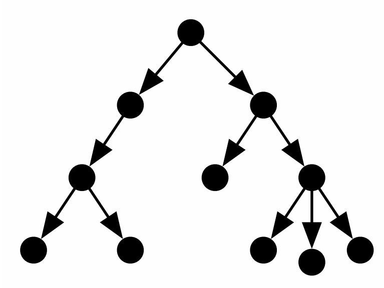

Figure 1. A hierarchical organization.

organization dynamically [54, 78]. This approach will ultimately enable suitably equipped agent populations to organize themselves, eliminating at least some of the need to exhaustively determine all possible runtime conditions a priori. Before this can occur, the space of organizational options must be mapped, and their relative benefits and costs understood.

These benefits and costs, and the potential advantages that could be provided by technologies able to make use of such knowledge, motivate the need to determine the characteristics of organizations and under what circumstances they are appropriate. While no two organizational instances are likely to be identical, there are identifiable classes of organizations which share common characteristics [75]. Several organizational paradigms suitable for multi-agent systems have emerged from this line of research [28]. These cover particularly common, useful or interesting structures that can be described in some general form. In this paper we will describe several of these paradigms, give some insight into how they can be used and generated, and compare their strengths and weaknesses. The vast amount of research which has been done in this field precludes a complete survey of any one technique; we hope to provide the reader with a concise description and a sample of the interesting work that has been done in each area.

In the following sections, we will describe the origin, form, function and characteristics of a typical structure for each organizational paradigm. Examples applications will be presented, along with a discussion of techniques that have been employed to create the structures. By separating these concepts, we will distinguish between the characteristics of the organization generation process and those of the organizational structure itself, independently of how it was generated.

2 Hierarchies

The *hierarchy* or *hierarchical organization* is perhaps the earliest example of structured, organizational design applied to multi-agent system and earlier distributed artificial intelligence architectures [27, 56, 28, 4, 66]. Agents are conceptually arranged in a tree-like structure, as seen in Figure 1, where agents higher in the tree have a more global view than those below them. In its strictest interpretation, interactions do not take place across the tree, but only between connected entities. More recent work [63] has explored starting with a strict hierarchy and augmenting it with cross links to allow more direct communication.

The data produced by lower-level agents in a hierarchy typically travels upwards to provide a broader view, while control flows downward as the higher level agents provide direction to those below [4]. The simplest instance of this structure consists of a two-level hierarchy, where the lower level agents' actions are completely specified by the upper, which produces a global view from the resulting information [12]. More complex instances have multiple levels, while data flow, authority relations or other organizationally-dictated characteristics may not be absolute.

Fox [27] describes several different types of organizational hierarchies. The *simple* hierarchy endows a single apex member with the decision making authority in the system. *Uniform* hierarchies distribute this authority in different areas of the system to achieve efficiency gains through locality. Decisions are made by the agents which have both the information needed to reason about the decision, and the organizational authority to do make the decision. Each level acts as a filter, explicitly transferring information and implicitly transferring decisions up the hierarchy only when necessary. *Multi-divisional* hierarchies further exploit localization by dividing the organization along "product" lines, where products might represent different physical artifacts, services, or high-level goals. Each division has complete control over their product, which facilitates the decision making and resource allocation process by limiting outside influences. The divisions themselves may still be organized under a higher-level entity which evaluates their performance and offers guidance, but is strictly separated from the divisional decision process. These more sophisticated hierarchies look very much like like holarchical organizations, which are discussed in Section 3.

2.1 Characteristics

The applicability of hierarchical structuring comes from the natural decomposition possible in many different task environments. Indeed, task decomposition trees are a popular way of modeling individual agent plan recipes [16]; a hierarchical organization can be thought of as an assignment of roles and interconnections inspired by the global goal tree. The hierarchy's efficiency is also derived from this notion of decomposition, because the divide-and-conquer approach it engenders allows the system to use larger groups of agents more efficiently and address larger scale problems

[101]. This type of organization can constrain agents to a number of interactions that is small relative to the total population size. This allows control actions and behavior decisions become more tractable, increased parallelism can be exploited, and because there is less potentially distracting data they can obtain a more cohesive view of the information pertinent to those decisions [66].

It is not sufficient to simply aggregate increasing amounts of information to obtain higher utility or better performance. This information must be matched with sufficient computational power and analysis techniques to make effective use of the information [53]. Without this, the effort to transfer the data may be wasted and the excess information distract the agent from more important tasks. Alternatively, the information can be summarized, approximated or otherwise processed on its way up the tree to reduce the information load. However, in doing so, a new dimension of uncertainty is introduced because of the potential for necessary details to be lost. In this situation, the decision making authority should be correctly placed within the structure to maximize the tractable amount of useful information that is available that retains an acceptable level of uncertainty or imprecision [27].

Using a hierarchy can also lead to an overly rigid or fragile organization, prone to single-point failures with potentially global consequences [65]. For example, if the apex agent were to fail the entire structure's cohesion could be compromised. Of course this agent could be replaced, but it may then prove costly to restore the concentrated information possessed by its predecessor. It is similarly susceptible to bottleneck effects if the scope of control decisions or data receipt is not contained - consider what would happen if that apex agent received all the raw data produced by a large group of agents below it.

2.2 Formation

Although the algorithm itself does not enforce a strict hierarchy such as the one described earlier, Smith's contract net protocol [88] provides a straightforward mechanism to construct a series of connections with most of the same characteristics. This structure is also implicitly based on the way the high-level goal is able to be decomposed. Upon receipt of a new task, an agent first chooses to perform the task itself, or search for agents willing to help complete the task. As part of this search process, the agent may decompose the task into subtasks or *contracts*. The agent, acting as a contractor, announces these contracts along with a bid specification to a subset of its peers who then decide if they wish to submit a bid. The bids which return to the contractor contain relevant information about the potential contractee which allows it to discriminate among competing offers. A contractee is selected and notified. Upon receipt of the new task, the contractee now faces the same question - should it perform the task itself or contract it out? Repeated invocations of this process produce a hierarchy of contractors and contractees. Because agents individually choose which contracts to bid on, and contractors choose which bids to accept, this strategy can effectively assign tasks among a population of agents without the need for a global view.

As with most organizational structures, the shape of the hierarchy can affect the characteristics of both global and local behaviors. A very flat hierarchy where agents have a high degree of connectivity can lead to overloading if agent resources are both limited and consumed as a result of these connections. Conversely, a very tall structure may slow the system's performance because of the delays incurred by passing information across multiple levels. One approach to making this tradeoff is the use of agent *cloning* [43, 20, 65]. An agent in such a system may opt to create a copy or clone of itself, possessing the same capabilities as the original, in response to overloaded conditions. If additional resources are available for this clone to use, this process allows the agent to dynamically create an assistant that can relieve excess burden from the original, reducing load-related errors or inefficiencies in the process. If the new agent is subordinate to the original, then a hierarchical organization has been formed in the process. In [81] Shehory discusses using cloning when other task-reallocation strategies are not viable. In this work, an agent's overall load is a function of its local processing, free memory and communication. It uses a dynamic programming technique to compute an optimal time to clone, after which an appropriately idle computational node is sought out to house the new agent. The new agent receives a subset of the original task(s). The clones themselves require resources, and the results they produce may require an additional hop to get to their ultimate destination, so they may also be merged or destroyed when these costs outweigh their benefits.

3 Holarchies

The term *holon* was first coined by Arthur Koestler in his book *The Ghost In The Machine* [50]. In this work, Koestler attempts to present a unified, descriptive theory of physical systems based on the nested, self-similar organization that many such systems possess. For example, biological, astrological and social systems are all comprised of multi-leveled, grouped hierarchies. A universe is comprised of a number of galaxies, which are comprised of a number of solar systems, and so on, all the way down to subatomic particles. Each grouping in these systems has a character derived but distinct from the entities that are members of the group. At the same time, this same group contributes to the properties of one or more groups above it. The structure of each of these groupings is a basic unit of organization

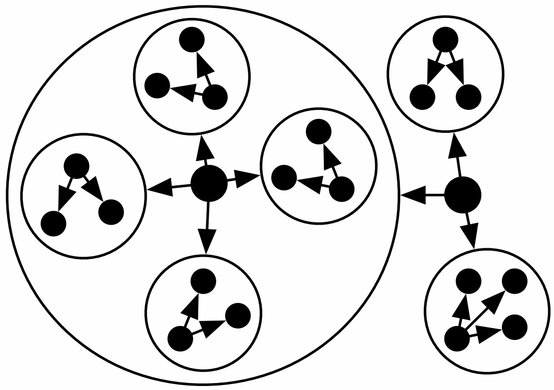

Figure 2. A holarchical organization.

that can be seen throughout the system as a whole. Koestler called such units holons, from the Greek word *holos*, meaning "whole", and *on*, meaning "part". Each holon exists simultaneously as both a distinct entity built from a collection of subordinates and as part of a larger entity.

True to Koestler's intent, this notion of a hierarchical, nested structure does accurately describe the organization of many systems. This concept has been exploited, primarily in business and manufacturing domains [102, 25, 98], to define and build structures called *holarchies* or *holonic organizations* which have this dual-nature characteristic. A sample such organization is shown in Figure 2. Enterprises, companies, divisions, working groups and individuals can each be viewed as a holons taking part in a larger holarchy.

The defining characteristic of a holarchy is the partially-autonomous holon. Each holon is composed of one or more subordinate entities, and can be a member of one or more superordinate holons. Holons frequently have both a software and physical hardware component [102, 97], although this does not preclude their usage in purely computational domains. The degree of autonomy associated with an individual holon is undefined, and could differ between levels or even between similar holons at the same level. There is the presumption, however, that the level of autonomy is neither complete nor completely absent, as these extremes would lead to either a strict hierarchy or an unorganized grouping, respectively. Within the hierarchy, the chain of command generally goes up - that is, subordinate holons relinquish some of their autonomy to the superordinate groupings they belong to. However, there is also the more heterarchical notion that individual holons determine how to accomplish the tasks they are given, since they are likely the locus of relevant expertise. Many holonic structures also support connections between holons across the organization, which can result in more amorphic, web-like organizational structures that can change shape over time [25, 102]. These richer models then begin to resemble and take on the characteristics of nearly-decomposable hierarchies [84], where lateral interactions are weak but still relevant. Very flat holarchies can also begin to resemble federations, which will be discussed in Section 8.

Holarchies have been used in agent-based systems primarily because of their ability to closely model existing business, manufacturing and enterprise systems. Fischer [25], Zhang [102], and Ulieru [98] have each organized agent systems by modeling explicit or implied divisions of labor in real-world systems as holons. In doing so, they create abstractions of these divisions, imparting capabilities to individual holons instead of individual agents. This layer of abstraction allows other entities in the system to make more effective use of these capabilities, by removing the need to for those entities to make the fine-grained decisions and interactions that take place within the holon.

3.1 Characteristics

As with normal hierarchies, holarchies are more easily applied to domains where goals can be recursively decomposed into subtasks that can be assigned to individual holons (although this is not essential). Given such a decomposition, or a capability map of the population, the benefits the holonic organizations provide are derived primarily from the partially autonomous and encapsulated nature of holons. Holons are usually endowed with sufficient autonomy to determine how best to satisfy the requests it receives. Because the requester need not know exactly how the request will be completed, the holon potentially has a great deal of flexibility in its choice of behaviors, which can enable it to closely coordinate potentially complimentary or conflicting tasks. This characteristic reduces the knowledge burden placed on the requester and allows the holon's behavior to adapt dynamically to new conditions without further coordination, so long as the original commitment's requirements are met. A drawback to this approach is that, lacking such knowledge, it is difficult to make predictions about the system's overall performance [5].

3.2 Formation

The challenge in creating a holonic organization revolves around selecting the appropriate agents to reside in the individual holons. The purpose of the holon must be useful within the broader context of the organization's high-level goals, and the holon's members must be effective at satisfying that purpose. Zhang [102] uses a model of static holons along with so-called mediator holons to create and adapt the organization. The static groups consist of product, product model and resource holons, each of which corresponds to a group of physical or information objects in the environment (e.g. manufacturing device, design plans, conveyors, etc.).

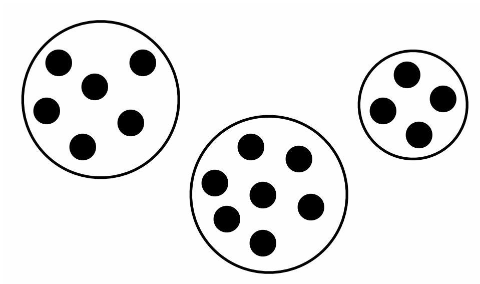

Figure 3. A coalition-based organization.

The mediator holon ties these together, by managing orders, finding product data and coordinating resources in a manner very similar to a federation, which will be discussed in Section 8. Each new task is represented by a dynamic mediator holon (DMH), which is created by the mediator holon. The DMH is destroyed when the task is completed.

Another approach to holarchy construction uses *fuzzy entropy* minimization to guide the formation of individual holonic clusters [90, 97]. In this work, the collection of holons is assumed to be initially described with a set of source-plans, each of which describes a potential assignment of holons to clusters, along with a set of probabilities that describe the degree of occurrence of those clusters. From this initial uncertain information, one can derive the preferences which agents have to work with one another, and then choose the source plan which has the minimal entropy with respect to those preferences. The goal of this technique is to ensure that each holon has the necessary knowledge and expertise needed to perform its task. The preference that one agent has for another represents this knowledge or expertise requirement, so the minimally fuzzy set will satisfy this goal by clustering agents which have common preferences. In [97], Ulieru adds a genetic algorithm approach to this scheme to help explore the space of possible clustering assignments.

4 Coalitions

The notion of a *coalition* of individuals has been studied by the game theory community for decades, and has proved to be a useful strategy in both real-world economic scenarios and multi-agent systems. If we view the population of agents A as set, then each subset of A is a potential *coalition*. Coalitions in general are goal-directed and short-lived; they are formed with a purpose in mind and dissolve when that need no longer exists, the coalition ceases to suit its designed purpose, or critical mass is lost as agents depart. Related research has extended this to longer-term agreements based on trust [6]. They may form in populations of both cooperative and self-interested agents.

A population of agents organized into coalitions is shown in Figure 3. Within a coalition, the organizational structure is typically flat, although there may be a distinguished "leading agent" which acts as a representative and intermediary for the group as a whole [49]. Once formed, coalitions may be treated as a single, atomic entity. Therefore, although coalitions have no explicit hierarchical characteristic, it is possible to form such an organization by nesting one group inside another. Overlapping coalitions are also possible [80]. The agents in this group are expected to coordinate their activities in a manner appropriate to the coalition's purpose. Coordination does not take place among agents in separate coalitions, except to the degree that their individual goals interact. For example, if one coalition's goal depends on the results of another, these two groups might need to agree upon a deadline by which those results are produced. In this case, it would be the leading or representative agents forming the commitment, not arbitrary members of the coalition.

In addition to the problem of generating coalition structures, one must also determine how to solve the goal presented to the coalition. If the population is self-interested, a division of value to be apportioned to participants once that goal has been satisfied must also be generated and agreed upon [77].

4.1 Characteristics

The motivation behind the coalition formation is the notion that the value of at least some of the participants may be superadditive along some dimension. Analogously, participants' costs may be subadditive. This implies that utility can be gained by working in groups – this is the same rationale behind buying clubs, co-ops, unions, public protests and the "safety in numbers" principle. For instance, in an economic domain, a larger group of agents might have increased bargaining strength or other monetary reward [96]. In computational domains we might expect more efficient task allocation, or the ability to solve goals with requirements greater than any single agent can offer [80]. In physically-limited systems, coalitions have been used to trade off the scope of agent interactions with the effectiveness of the system as a whole [86]. This last application directly affects the coordination costs incurred by the system; we will see that this capability and purpose are shared by congregations in Section 6.

One could argue that all agents in the environment should always join to form the all-inclusive *grand coalition*. Indeed, under certain circumstances this is appropriate, since the structure would have the resources of all available agents at its disposal, which theoretically would provide the maximum value. There are costs associated with forming and maintaining such a structure however, and in real world scenarios this can be both an impractical and unnecessarily coarse solution [77]. Therefore, the problem of coalition formation becomes one of selecting the appropriate set(s) $S \subseteq A$ which maximizes the utility (value minus costs) that coalition v_S can achieve in the environment. The value and cost of the coalition are generic terms, which may in fact be functions of other domain-dependent and independent characteristics of the structure.

4.2 Formation

The complexity of the coalition formation task depends on the conditions under which the coalitions will exist, and the types of coalitions which are permitted. As with all organizations, operating in dynamic environments will be harder to maintain than those in static ones. Additional complexity is also incurred if the partitioning of agents is not disjoint; that is, agents can have concurrent membership in more than coalition. Uncertain rewards, self-interested agents and a potential lack of trust while coordinating add further obstacles to the process.

Sandholm [76] analyzes the worst case performance of forming exhaustive, disjoint coalitions over a static agent population from a centralized perspective. They show that, by searching only the two lowest levels of a complete coalition structure graph, an a-approximate value solution can be found to the partitioning problem, where $a = |A|$. Although the search of 2^{a-1} possible allocations still grows exponentially with a, the fraction of coalition structure needing to be searched approaches zero. They also present an anytime algorithm which can meet tighter bounds given additional time. Later work empirically evaluates the average-case performance of three anytime search techniques [51]. In these tests the algorithms' performance varied by domain characteristics, and no single technique was best in all conditions.

Shehory [80] has studied how coalitions may be used to enable task achievement by a group of agents. In their scenario, a set of interdependent (precedence) tasks must be accomplished, some of which require multiple agents to perform. The agents are cooperative and potentially heterogeneous in their capabilities. The strategy they employ draws on techniques used by Chvatal's set covering algorithm. The initial values of all possible size-bounded coalitions are first computed and then iteratively refined in a distributed manner by the agents, taking into account task ordering and capability requirements. Once computed, the highest valued coalitions, either disjoint or overlapping depending on the selection algorithm, are instantiated. This algorithm was also augmented to support dynamically arriving tasks. A drawback to this addition is that, in the worst case, the organization process needs to be redone for each task, incurring a significant communication cost. Also limiting the potential scalability of this approach is the need for each agent to have full knowledge of the available agents and tasks.

Lerman [52] presents a scalable strategy where coalitions are formed between self-interested agents based only on local decision making. In this work agents operate in an electronic marketplace consisting of a number of extant purchase orders. Agents form coalitions by adding a purchase request to an order, and can leave that coalition by removing their order. Agents in the system can move at will between purchase orders, searching for the one which offers the best value (lowest cost). An analysis based on differential equations shows that this strategy reaches equilibrium. It also has very low communication and computational requirements. However, it does not provide guarantees on the achievable value or convergence rate, which would be affected by scale, and does not have a notion of deadlines on the purchase orders.

Soh [89] presents a technique where coalitions are dynamically created in response to the recognition of tracking tasks in a distributed sensor network. In this work, agents are assumed to have incomplete, uncertain knowledge and must respond to events in real time for goal achievement to be possible. As such, coalitions are formed in a saticificing, rather than optimal manner. An agent initiates coalition formation by first using local knowledge to select a subset of candidate partners that it believes will satisfy its requirements, both in terms of capabilities and willingness to cooperate. Next, it sequentially engages these candidates, in utility-ranked order, in argumentative negotiation, where offers and counteroffers are exchanged. This proceeds until satisfactory membership is decided, or the candidate list is exhausted. Agents are cooperative, so during this negotiation process agents explicitly decide what coalition(s) they are willing to join based on perceived gains in utility. This approach does not make any guarantees about coalition value, or even that a satisfactory coalition will be found, but given the relatively short time in which an allocation must be made it would seem to be a reasonable strategy. In addition, reinforcement learning is used over the course of events to estimate candidate utility more accurately and select the most beneficial negotiation strategy, which should improve coalition value in the long run for reasonably stable environments. By storing preferences over multiple episodes, this learning also implicitly adds longevity to coalitions, giving organizational structures produced by this technique an interesting mix of dynamic and long-term characteristics.

5 Teams

An agent *team* consists of a number of cooperative agents which have agreed to work together towards a common goal [28, 92, 3]. In comparison to coalitions, teams at-

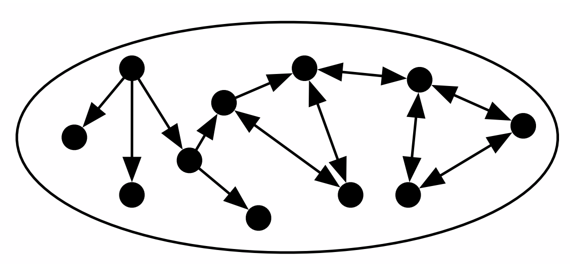

Figure 4. A team-based organization.

tempt to maximize the utility of the team (goal) itself, rather than that of the individual members. Agents are expected to coordinate in some fashion such that their individual actions are consistent with and supportive of the team's goal. Within a team, the type of interactions can be quite arbitrary, as seen in Figure 4, but in general each agent will take on one or more roles needed to address the subtasks required by the team goal. Those roles may change over time in response to planned or unplanned events, while the high-level goal itself usually remains relatively consistent (although exception handling may promote the execution of previously dormant subtasks).

This description of agent teams is quite general, and nearly any cooperative agent system has characteristics that are similar to these, if only implicitly. However, systems which maintain an explicit representation of their teamwork or joint mental state are differentiated in their ability to reason more precisely about the consequences of their teamwork decisions [44, 35, 92]. For example, they will typically have representations of shared goals, mutual beliefs and team-level plans. This type of representation provides flexibility and robustness by allowing the agents to explicitly reason about team-level behaviors, where a less explicit system may rely on a set of assumptions that ultimately make the system brittle in the face of unexpected situations.

5.1 Characteristics

The primary benefit of teamwork is that by acting in concert, the group of agents can address larger problems than any individual is capable of. Other potential benefits, such as redundancy, the ability to meet global constraints, and economies of scale can also be realized [37]. However, it is the ability of the team (members) to reason explicitly about the ramifications of inter-agent interactions which gives the team the needed flexibility to work in uncertain environments under unforeseen conditions. The drawback to this tighter coupling is increased communication [71], so the team and joint goal representations, domain characteristics and task requirements are frequently used to determine what level of cooperation (and therefore communication) is needed [73].

Jennings [44] describes an electricity transportation management system which employs teamwork to organize the activities of diagnostic agents. Lacking such structure, the agents were prone to incoherent and wasteful activities, since they did not always share useful behavior information or propagate important environmental knowledge. By providing agents with an explicit representation of shared tasks and the means by which cooperation should progress, the agents were able to accurately reason about and resolve these interactions by employing team-level knowledge. Similarly, in [92], teamwork is used to provide the structure and coordination needed by agents to address interdependent goals in dynamic environments, such as tactical military exercises and competitive soccer games. These works demonstrate how pathological, but hard to predict failures can be addressed if the plans are backed up by a general model of teamwork.

5.2 Formation

The challenges associated with team formation involve three principle problems: determining how agents will be allocated to address the high-level problem, maintaining consistency among those agents during execution, and revising the team as the environment or agent population changes [44, 62, 94].

The selection and role-assignment of agents that will work on the high-level depends on the goal's requirements, the capabilities of the candidate agents, and the knowledge of the selecting process itself [95, 3]. Initially, the process or agent performing the team construction must be aware of the agents which could potentially form the team. In the case of a static, reasonably sized agent population this can be done off-line as part of the system design, or the members can be dynamically discovered and assessed. This latter technique can be accomplished using well-known discovery mechanisms such as the contract net protocol [88] or matchmaker intermediaries [91]. Once a suitable pool has been found, the capabilities and preexisting responsibility of those agents must be evaluated relative to the needs of the goal. Typically, agents are each denoted to have a set of capabilities, while the goal's subtask(s) are of a particular type. If an agent's capabilities include that subtask's type, it can perform the task [95, 24]. The discovery mechanisms may include an implicit ranking technique, such as the bidding process employed in contract net, which makes the selection process relatively straightforward. Tidhar [95] suggests a different technique where the agent characteristics at derived at compile time, either through designer input or automatic analysis of the agent's plan library. Candidate teams comprised of a subset of those agents may also be specified, which also are marked with their characteristics. At runtime, these characteristics are matched with the goal requirements as part of the team allocation search. By including these characteristic labels, the number of possible team combinations can be greatly reduced.

Tambe's STEAM (Shell for TEAMwork) [92] architecture provides a flexible method for representing and adapting team behaviors. It is based largely on the joint intentions framework [57] (also used by Jennings' GRATE* teamwork architecture [44]) which formally defines how agents should reason over joint commitments and shared goals. This ensures a consistency of belief, or a desire to enact such a belief, across all team members. The commitments formed through the joint intentions process provide the explicit structure needed to reason about and monitor performance on a team level. Team plans are represented using a hierarchical decomposition tree, with nodes representing tasks for both teams and individuals, with associated preconditions, application and termination rules. Agents may simultaneously take part in several different tasks, and corresponding roles. The team's cohesion is derived primarily from the joint intentions created as part of executing the team plans. Upon selecting a team task, agents first broadcasts this intention to affected agents, and waits until a commitment to that task has been established between all participants. The existence of this commitment directs agents to propagate changes whenever the task is perceived to be achieved, unachievable or irrelevant, before taking local action itself. This trades off the potential reaction speed of the team and the cost of communication with group conformity. A decision theoretic approach is used to guide communication acts, which explicitly trades off the costs of communication with those of inconsistent beliefs. Nair [68] has also explored the possibility of using simulated emotions to provide the motivation to enforce team-level behaviors.

In STEAM, monitoring and repair of the team is accomplished with the use of role constraints [92]. Team members are assigned a role, based on the particular task they are working on. These roles are further constrained such that some particular combination of them (e.g. and, or) are needed to accomplish the task. One can then monitor if a task is achievable by monitoring the health of the individual agents, and using that information to evaluate the satisfiability of the role constraints. Such monitoring can be performed through explicit queries, environmental observations or by eavesdropping on communication. Kaminka [46] has demonstrated that the latter technique can perform well when coupled with a plan-recognition algorithm. Failures can thus be detected, and potentially resolved through an appropriate role-substitution, or the task abandoned if no substitution is possible. Alternately, one could use a diagnosis system [44, 38] to more precisely identify the root cause of the failure. Interestingly, this repair operation can itself be cast as a team task, so mutual agreement that a repair is necessary must be achieved before potentially drastic measures are taken. Nair [69] shows how an MDP incorporating team and role-allocation knowledge can improve the system's response in cases of multiple role failure. In this case, a suitable locally optimal policy for the reallocation decision problem can be found by analyzing the team's plans, and then used to guide online responses to failures. This work showed that such policies can provide improved performance versus more heuristic and analytic techniques. A similar technique was also shown in that work to improve initial role allocation.

Tidhar [94] uses a similar hierarchical plan representation to represent teamwork in a tactical air mission scenario. Team membership and role assignment are performed by matching agent capabilities to one or more role's requirements. As in STEAM, teams can be broken down into subteams, and agents may use both implicit (observation) and explicit (messaging) forms of coordination.

The Generalized Partial Global Planning (GPGP) framework also employs techniques that allow agents to act using team semantics [17, 55]. Where a STEAM-driven system will typically organize in an explicit, controlled fashion in response to a perceived goal, a GPGP-team is created in a more dynamic, emergent fashion. GPGP agents are provided with a set of individual plans which model a range of alternative ways that goals may be achieved. The sub-goals modeled in these plans may affect or be affected by other agents in the environment, although this may not be initially recognized. By communicating with one another and exchanging plans and schedules, these *non-local interrelationships* between tasks may be recognized. For example, the results from one agent's activity may be a strict prerequisite for another agent's task. They may alternately be a facilitating, but not required input to a task. By recognizing these interrelationships, and sharing knowledge of what goals are being pursued, agents gradually build an internal model of how their actions may affect others. This knowledge is similar to that created by the more formal joint intentions of STEAM, and allows agents to influence local behavior and communicate results as if they were members of a common team.

6 Congregations

Similar to coalitions and teams, agent *congregations* are groups of individuals who have banded together into a typically flat organization in order to derive additional benefits. Unlike these other paradigms, congregations are assumed to be long-lived and are not formed with a single specific goal in mind. Instead, congregations are formed among agents with similar or complementary characteristics to facilitate the process of finding suitable collaborators, as modeled in Figure 5. Individual agents do not necessarily have a single or fixed goal, but do have a stable set

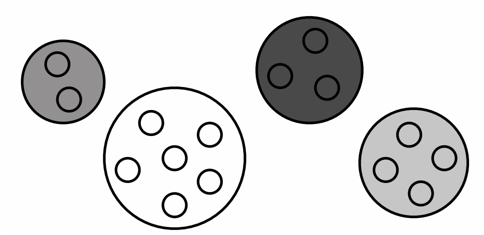

Figure 5. Congregations of agents.

of capabilities or requirements which motivate the need to congregate [9]. Analogous human structures include clubs, marketplaces, support groups, secretarial pools, academic departments and religious groups, from which the name is derived.

Congregating agents are expected to be individually rational, by maximizing their local long-term utility. Group or global rewards are not used in this formalism [9]. It is this desire to increase local utility which drives congregation selection, because it is the utility that can be provided by a congregation's (potential) members that determine how useful it is to the agent. Agents may come and go dynamically over the existence of the congregation, although clearly there must be a relatively stable number of participants for it to be useful. Agents must also take enough advantage of the congregation so that that the time and energy invested in forming and finding the group is outweighed by the benefits derived from it. Since congregations are formed in large part to reduce the complexity of search and limit interactions, communication does not occur between agents in different congregations, although the groups are not necessarily disjoint. The net result of the congregating can produce a higher average utility per cycle spent computing or communicating [7].

6.1 Characteristics

Although congregations can theoretically share many of the same benefits of coalitions, their function in current research has been to facilitate the discovery of agent partners by restricting the size of the population that must be searched. The downside to this strategy is that the limited set may be overly restrictive, and not contain the optimal agents one might select given infinite resources. So, in forming the congregation, one is trading off quality and flexibility for a reduction in time, complexity or cost. If an appropriate balance can be found, this will result in a net gain in utility.

This hypothesis is borne out in the experiments from an information economy domain presented in [7]. This work varied the number of congregations that agents were allowed to form. Since the population size was static, the average congregation size decreased as the number of congregations increased. The accumulated quality decreased proportionally because of less flexibility in agent interactions. However, these smaller congregations also incurred lower overhead, and thus had less cost. A median point was discovered in the space which produced maximum value.

6.2 Formation

Like coalition formation, congregation formation involves selecting or creating an appropriate group to join, and suffers from similar complexity problems as the agent population grows. Because congregations are more ideologically or capability driven, and there is usually no specific goal or task to unite them, one must first define how these groups may be differentiated. In [8] Brooks proposes using labels to address this problem. A label is a suitably descriptive tag assigned to each congregation which serves to both distinguish it from other groups and advertise the characteristics of its (desired) members. Assuming that agents have an ordered preference for such labels, the congregators' action is simply to move to the congregation for which it has the highest preference. The problem is then to create a number of logical points where agents may congregate and then decide upon the labels each congregation point will have; these labels help determine the makeup of the population which gathers there. Each agent was placed into one of several affinity groups, and a congregation is stable if and only if it contains only members of the same affinity group. Different numbers of labelers were then added which could attach labels to the congregation points. As with the congregators, the labelers were stable if and only if the congregation they provided the label to was homogeneous. The experimental and analytic results demonstrated that by increasing the number of labelers the system converged more quickly.

Brooks [7] presents a variation of this formation technique used in an information economy which also takes into account the costs associated with congregation size. In this scenario there are a set of buyers and sellers. Each buyer has an information preference, and each seller may choose what type of information to offer. The buyer's preference is soft – they have an optimal type, but are also willing to purchase related information, where similarity determines how much they are willing to pay. Instead of explicitly labeling congregation points, agents freely move through the system seeking groups that provide acceptable utility. The scenario is episodic, where during each episode agents elect to stay in place or randomly move to a new congregation. At the end of each episode an auction takes place from which buyers and sellers obtain their utility. The utility is based on the price of the goods bought and sold, combined with the costs incurred during the auction. This cost, divided uniformly among the congregation members, is proportional to

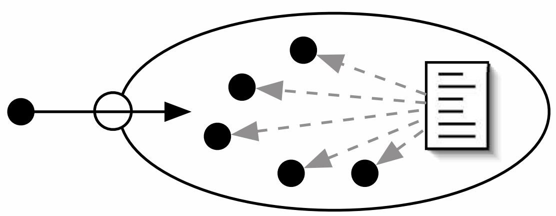

Figure 6. An agent society.

the complexity of the auction, which is itself determined by the number of participants. Satisfied agents remain, while those which do not obtain enough utility move. This process results in an emergent population of congregations that trades off utility for computation time.

Although it does not strictly deal with congregating agents, Sen's work on reciprocal behavior [79] has some of the same characteristics. In this system, agents become more inclined to cooperate or assist another agent when it has a favorable history with that other agent. Specifically, agents track if others have cooperated with it in the past, or if it has cooperated with them, along with the approximate costs of those experiences. If an agent has a favorable balance of cooperation, it will be more inclined to give or receive assistance. The cooperation decision process is stochastic, enabling reciprocal relationships to be created or promoted even when a strictly positive balance does not exist. Weak groups may form between agents using this strategy who have complementary capabilities, which is similar to the notion of congregations we have presented. Because agents will more likely communicate with those that will help it, interactions can become implicitly confined within the group. These groupings are not formalized or well-defined, however, and communication is not necessarily restricted by the approximate boundaries that form. Sen showed that, among a group of self-interested agents operating in a package delivery domain, a population containing reciprocal agents outperformed a selfish population.

7 Societies

Drawing from our own experiences with biological societies, a *society* of agents intuitively brings to mind a long-lived, social construct. Unlike some other organizational paradigms, agent societies are inherently open systems. Agents of different stripes may come and go at will while the society persists, acting as an environment through which the participants meet and interact. A canonical example of this paradigm is the electronic marketplace, consisting of buyers and sellers striving to maximize their individual utility [100, 21, 1]. Agents will have different goals, varied levels of rationality, and heterogeneous capabilities; the societal construct provides a common domain through which they can act and communicate. Societies are also more ephemeral constructs than others paradigms we have seen so far. They impose structure and order, but the specific arrangement of interactions can be quite flexible. Within the society, agents may be sub-organized into other organizations, or be completely unrelated.

A second distinguishing characteristic of societies is the set of constraints they impose on the behavior of the agents, commonly known as *social laws* or *norms*. This arrangement is shown abstractly in Figure 6. These are rules or guidelines by which agents must act, which provides a level of consistency of behavior and interface intended to facilitate coexistence. For example, it might constrain the type of protocol(s) agents can use to communicate, specify a currency by which they can transfer utility, or limit the behaviors the agent can exhibit in the environment. Penalties or sanctions may also exist to enforce these laws.

The set of laws embedded in a society must strike a balance among objectives. It must be sufficiently flexible that goals are achievable, but not so flexible that agents must become unreasonably complex to operate with such flexibility (recall that this is an open system, where agents are not all designed by one individual). It must also be fair, such that the goals of one class of individuals are not incorrectly valued higher than those of another. These issues arise naturally in any structured, multiple participant system, and [67] argues that most multi-agent systems have some form of social laws in place, if only implicitly.

7.1 Characteristics

In [83], Shoham presents a grid world where robots must move from one location to another in accordance with a set of dynamically arriving tasks. Conflicts can arise when two or more agents attempt to occupy the same location at the same time along their chosen paths. They argue that a centralized solution is untenable, because of the potentially large number of interactions that must be continuously reasoned over in the heterogeneous population. Neither is a fully decentralized solution appropriate, because of the number of negotiation events that would need to take place at each time step. This motivates the need for "traffic laws", a type of social law which does not eliminate such interactions, but should minimize the need for them. The traffic laws in this research are computed offline, and constrain the robots' movement patterns in such a way that collisions do not occur, and destinations are reachable within a bounded amount of time. Vehicular traffic laws serve the same purpose in human societies. When driving a car there is no central authority which determines when and where we should go, and neither is there a free-for-all on the roads where one must talk to every other driver before proceeding. The challenge then is to design a set of laws that minimizes conflicts and encourages efficient solutions.

Although social laws were used to provide efficiency benefits in the work above, the purpose of an agent society is not always as quantitatively-driven as other organizational constructs. Indeed, most of research on agent societies is more concerned with how the concepts they embody can be used to facilitate the construction of large-scale, open agent systems in general. For example, Moses [67] argues that social laws can provide a formal structure upon which more complex inter-agent behaviors can be built. By limiting and enforcing these restrictions, agents can make simplifying assumptions about the behavior of other agents, which can make interaction and coordination more tractable. Dellarocas [21] presents a complementary but more concrete view of agent societies, where some of the social laws are instantiated and enforced using social institutions provided in the environment. Agents are expected to formalize their interactions using contracts, which are independently verified by these institutions, thereby relocating some of the traditionally agent-centric complexity into a service available to the population as a whole. This reduces the burden placed on agent designers, and provides a mechanism where systemic (non-localized or long-term) failures may be detected more readily. This more rigorous enforcement of social laws also addresses the problem of unreliable, dishonest or malicious agents operating in the open environment.

Huhns [42] provides similar motivation for common communication languages, shared or interoperable ontologies and coordination and negotiation protocols, all of which may be specified as part of the society's structure. These beliefs can be supported by our own experiences in real life. It should be clear that complex human societies are founded upon the ability to interact with one another. Mutually understood and respected norms simplify many aspects of day-to-day existence. These principles can be used to the same effect in agent societies.

7.2 Formation

There are two aspects to the society formation problem. The first is to define the roles, protocols and social laws which form the foundation of the society. Given such a definition, the second problem is to implement the more literal formation of the society; that is, determining how agents may join and leave it.

If the society is to be an open and flexible system, its structure must be formally encoded so that potential members may analyze it and determine compatibility. This description can be as simple as a set of common interfaces that must be implemented, or a complex description of permissible roles, high-level objectives and social laws. Dignum [22] presents a three-part framework, consisting of organizational, social and interaction models. The organizational model defines the roles, norms, interactions and communication frameworks that are available in the environment. The social model, instantiated at run-time, defines which roles agents have taken on. The interaction model, also created at run-time, encodes the interactions between agents that have been agreed-upon, including the potential reward and penalties. The latter two models are supported by contracts between the relevant entities. This formalism is similar to [1], which provides additional details describing operators that can be used to encode social laws, roles and normative relations. Because the society is intended to be open, these structures do not involve the internal implementation of agents, but describe only the intended or expected externally observable characteristics of the participants and environment.

Dellarocas defines the act of an agent entering a society to be the *socialization* process [21]. In that work, they suggest this can be accomplished through an explicit negotiation process between the agent and a representative of the society, as shown in the left side of Figure 6. This exchange results in a *social contract*, or an explicit agreement made between the agent and the society indicating the conditions under which the agent may join that society. This allows the possibility of capable agents dynamically learning, and potentially negotiating over, the rules it must abide by in that society. A similar view is taken by Glaser in [33], with the additional stipulation that the joining agent must increase the utility of the society. This naturally extends to multi-society environments, where an agent's skills and goals define how good a fit it is with a particular society. Some of the challenges associated with operating in multi-society environments seem to be comparable, though larger in scale, to those encountered during coalition or congregation formation.

Because of their inherent flexibility, a great deal of additional complexity may be associated with social organizations. Sophisticated legal systems, communication bridges, ontologies, exception handling services, directories may all be part of the society model [21, 22, 48]. Some or all of these may be directly instantiated by trusted agents taking on so-called facilitation roles (differentiated from the operational roles taken on by worker agents).

8 Federations

Agent *federations*, or *federated systems*, come in many different varieties. All share the common characteristic of a group of agents which have ceded some amount of autonomy to a single delegate which represents the group [32]. This organizational style is modeled on the governmental system of the same name, where regional provinces retain some amount of local autonomy while operating under a single central government. The delegate is a distinguished

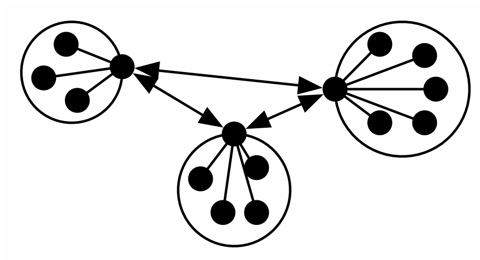

Figure 7. An agent federation.

agent member of the group, sometimes called a facilitator, mediator or broker [91, 36]. Group members interact only with this agent, which acts as an intermediary between the group and the outside world, as shown in Figure 7. Typically, the intermediate accepts skill and need descriptions from the agents, which it uses to match with requests from intermediaries representing other groups. In this way the group is provided with a single, consistent interface.

8.1 Characteristics

The capabilities provided by the intermediary are what differentiate a federation from other organizational types. The intermediary functions on one hand by receiving potentially undirected messages from its group members. These may include skill descriptions, task requirements, status information, application-level data and the like. These will typically be communicated using some general, declarative communication language which the facilitator understands [32]. Outside of the group, the intermediary sends and receives information with the intermediaries of other groups. This could include task requests, capability notifications and application-level data routed as part of a previously created commitment. Implicit in this arrangement is that, while the intermediary must be able to interact with both its local federation members and with other intermediaries, individual normal agents do not require a common language as they never directly interact. This makes this arrangement particularly useful for integrating legacy or an otherwise heterogeneous group of agents [32, 82].

The intermediary itself can function in many different capacities. It may act as a translator, perform task allocation, or monitor progress, among other things. An intermediary which accepts task requests and allocates that task among its members is known as a broker or a facilitator. As part of the allocation, the broker may decompose the problem into more manageable subtasks. This allows agents to take advantage of all the capabilities of the (potentially changing) federation, without requiring knowledge of which agents perform a task or how they go about doing it. This results in a savings in complexity and messaging for the client, but also has the potential of making the broker itself a bottleneck [36] (a possibility common to all intermediaries). An intermediary acting as go-between among agents is known variously as a translator, embassy or mediator depending on its specific characteristics. Embassy agents provide a layer of security for members of their federation, by having the ability to deny communication requests. Mediator agents store representations of all related parties, reducing their individual complexity by providing a layer of abstraction. This capacity can be further exploited to arbitrate conflicts [59]. Intermediaries which provide the ability to track the state of one or more of its participants are known as monitors. For example, result information can be automatically propagated to interested parties. Of course, one or more of these roles may be combined into a single intermediary which offers several types of services.

8.2 Formation

Singh and Genesereth in [87, 32] describe how a general federated system would work. All agents are expected to communicate using the Agent Communication Language (or ACL, a somewhat-generic term used by many researchers to describe their agents' communication protocol), which is a combination of the first-order predicate calculus KIF with the KQML agent messaging language. Knowledge and statements sent between agents are encoded as KIF statements, which are then wrapped in KQML to provide a standard mechanism for specifying the sender, receiver, intent, and so forth. This provides a common language and set of behavioral constraints which will allow the various agents to interact. Not all agents must implement the entire class of concepts in ACL, but the aspects they do use must be correct with respect to the ACL specification. In addition, although they speak the same language, not all agents must use the same vocabulary to describe a particular situation, although to interact there must be an intermediary capable of translating the vocabularies. The system is initialized with a set of intermediaries called facilitators, which serve many of the roles outlined above, notably brokering. Agents connecting to the system start by sending their capabilities to the local facilitator. Implicit in this communication is the notion that the agent is willing to use those capabilities in service of requests posed by the facilitator. Needs are similarly routed to the facilitator, which then attempts to find other facilitators which can service that need. Each facilitator provides a yellow pages function which supports this search. Khedro's Facilitators [47] and the jointly developed PACT project [15] have produced very similar systems that also use ACL and a community of intermediaries to produce a robust and dynamic task decomposition and allocation scheme among a group of heterogeneous participants.

The MetaMorph I [65] and II [82] architectures de-

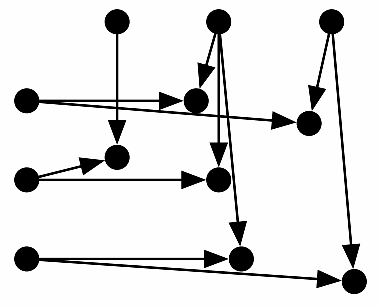

Figure 8. A matrix organization.

scribed by Maturana and Shen demonstrate a federated agent system for use in intelligent manufacturing. In this domain, agents are used to drive aspects of product design and manufacturing, contending with heterogeneous resources, dynamically changing conditions, and hard and soft constraints on behavior. MetaMorph's name is derived from the fact that the system can continuously change shape, adapting to new conditions as they are perceived. This is accomplished in part through the use of intermediaries called mediators, which are responsible for brokering, recruiting and conflict resolution services. The recruiting service is similar to brokering, but is differentiated by the fact that the intermediary can remove itself from the relationship once the partners have been discovered. This weaker form of federation provides efficiency gains at the cost of less flexibility, both due to the loss of the layer of abstraction that exists in the brokered approach. The federations themselves are dynamically created in response to new task arrivals or requests from other groups using a contract net [88] approach, or are statically created from agents in a common subsystem (e.g. tools, workers, etc.).

9 Matrix Organizations

We have seen that the strict hierarchical organization method is based on a tree-like structure of control. Agents or agent teams report to a single manager, which provides the agents with goals, direction and feedback. *Matrix organizations* relax the one-agent, one-manager restriction, by permitting many managers or peers to influence the activities of an agent. This is in some sense a closer approximation to how humans exist. An individual may receive guidance or pressures from their boss, co-workers, spouse, children, colleagues, etc. Even in a purely business setting one might have to report to an immediate supervisor, project managers, vendors, and peers at cooperating businesses. Interrelationships can come from many directions, each with its own objectives, relative importance and pertinent characteristics.

The term matrix organization comes from a grid based view of the participants. One can place managers around a group of "worker" agents, and use an arc to indicate authority, as in Figure 8. Alternately, agents are the rows and managers the columns (these sets may overlap), and a check is used to denote where an authority relationship exists. Like the hierarchy's tree, the matrix provides a graphical way to depict which managers can influence the activities of each agent.

9.1 Characteristics

Matrix organizations provide the ability to explicitly specify how the behaviors of an agent or agent group may be influenced by multiple lines of authority [18]. In this way, the agent's capabilities may be shared, and the agent's behaviors (hopefully) influenced so as to benefit all. This is particularly important if the agents themselves are viewed as a functional, limited resources. For example, if a particular skill is needed by two separate tasks, the agent can be used to address both, provided it has sufficient computational power. In the case where the agent has multiple ways of performing a task, it can also choose the method which best satisfies its peers.

This sharing come as a price, however, because the shared agent becomes a potential point of contention. If its managers disagree, the agent's actions may become dysfunctional as it is pulled in too many directions at once [78, 75]. To operate effectively, the agent must have a commitment ranking mechanism and sufficient autonomy to resolve local conflicts, or the ability to promote conflicts to a higher level where they may be resolved [60].

9.2 Formation

Decker [18] describes the MACRON organizational architecture, in which agents form a matrix organization. The domain for their system is cooperative information gathering, where multiple agents search for relevant in response to a user's query. Individual agents are separated into predefined functional groups, which contain agents that can access a particular type of information. These groups are under the control of a functional manager, who assigns agents to query tasks as they arrive. User query agents generate those query tasks, and therefore use the functional managers to dynamically select agents to satisfy their own goals. Individual gathering agents report to two agents: a static functional manager, and a query manager which changes depending on the user's actions. This has the effect of assigning the minimal needed set of agents to the query, increasing efficiency when compared to a system employing a set of static teams where particular team members might go unused, depending on the query characteristics. At the same time, this approach uses fewer resources than one lacking

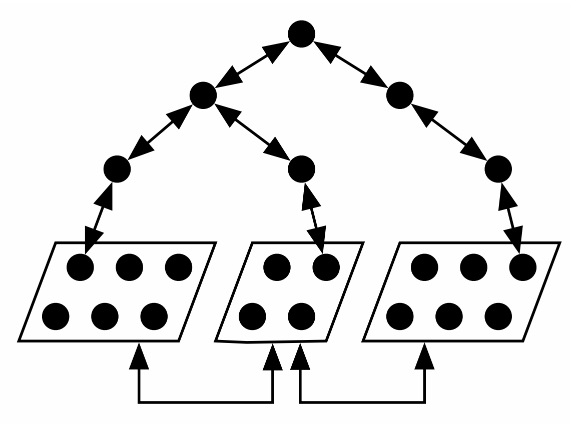

Figure 9. A compound organization.

functional groups, which would have to search through all available agents for each query.

In [40], Horling and Mailler describe a distributed sensor network application where a matrix organization is used to address a resource allocation problem. In this case, the sensors themselves were limited resources, since their heterogeneous locations and orientations made each one unique. The tracking process for each target was controlled by a different track manager, which was responsible for discovering and coordinating with the sensors needed to track its target. When multiple targets came in close proximity to the same sensor, a matrix organization is dynamically formed as the relevant managers interact with that sensor. At the same time, that sensor may have previously been given tasks by a regional manager responsible for detecting new targets. The result is an individual which may be under contention by three or more managers, and which must then decide how best to meet those demands. This was done using a combination of a predefined ranking scheme (tracking has higher priority than scanning for new targets), local autonomy (round robin scheduling) and conflict elevation (track managers negotiate directly once aware of the conflict).

10 Compound Organizations

Not all organizational structures fit neatly into a particular category, and some architectures may include characteristics of several different styles. A system may have one organization for control, another for data flow, a third for discovery, and so on. For example, Durfee's PGP [23] incorporates one organization for planning, and another separate structuring of the same agents to manage coordination problems. Compound organizations can be overlapped, operating as virtual peers at the same conceptual level, or be nested, so that some subset of agents in a group are organized in a potentially different way within the larger context. A sample such organization is shown in Figure 9. As with singular organizations, they may be created or adapted over time, or they may be instantiated as part of a transient form while a population shifts between organizational styles. Ideally, these compound architectures can use the most effective structure for the particular goal at hand, without limiting options that might be used elsewhere in the system. The tradeoff in this situation is usually one of complexity. Because an individual agent might take on different roles in response to different organizational demands, the agent itself must have sufficient sophistication to act efficiently and asynchronously in all those roles.

Some of the organizational paradigms which have been discussed so far are more amenable to coexistence than others. In much of the teamwork research, for example, a loose hierarchy of control was created among the agents after the team had formed [92, 95]. Hierarchical structures for interpreting and consolidating raw data are a popular mechanism for handling scale that can augment a preexisting or lower-level structure [101]. Societies also generally have an internal organizational structure within the larger context defined by the social laws and norms [21, 22]. They can also easily be viewed as a common "pool" of agents, from which other organizations can be constituted. In this type of compound organization, the society may exist in support of other, more dynamic structures which may be created address particular tasks. This begins to touch on the notion of organization longevity, which will be addressed in Section 12.

10.1 Characteristics

The positive and negative characteristics of a compound organization are derived primarily from its constituent parts. However, the interplay between organizations can lead to unexpected consequences. For example, if the distinguished intermediary in a federated system plays a key role in a separate overlay organization, it may be unable to fulfill both roles adequately. Similar to a matrix organization, agents may be faced with conditions where it is not clear which of two competing objectives it should satisfy [75]. Conversely, its knowledge of the requirements of both organizations may enable it to make more globally effective decisions. The possible interactions and formation strategies among arbitrary coexisting organizations are difficult to characterize in a general manner; instead we will proceed with a discussion of example systems employing this technique.

10.2 Example Compound Organizations

The distributed sensor network solution described by Horling and Mailler [40] uses several different overlapping organizational techniques. Agents are first partitioned into federations, called sectors, where membership is based on their geographic proximity. A distinguished member of each group is given the role of sector manager, who provides a form of recruiting service to other agents in the environment. This recruiting service supports the activities of track managers, who must discover and use the appropriate sensors as part of its tracking task. In forming the federations, the search time is reduced because only a subset of the population (the sector managers) needs to be interacted with, and communication requirements requirements are reduced because only the necessary subset of sensors will be returned. As discussed in Section 9, both the sector and track managers provide tasks to individual sensors, forming a matrix organization in the process. This arrangement facilitates resource sharing by allowing the sensors to guide their local activities based on the needs of potentially several interested parties, but can also lead to conflicts caused by over-demand. Because the sensor is a finite resource, a cloning technique like the one discussed in Section 2 cannot be used to address the conflict. Instead, a loose peer-to-peer relationship between track managers allows them to negotiate directly, alleviating the conflict through demand relaxation or by using alternate sensors. This resource allocation scheme employs a second, weaker form of federation through its use of mediators [59]. The conflicts, which may be potentially multi-linked and far-reaching, are partially centralized by a mediator agent which acts on the part of the relevant agents to find a suitable solution. In [39] the quantitative effects of these organizations are presented. Those experiments demonstrate how even a subset of the parameters which govern these structures can affect several different characteristics of the large-scale system.

Yadgar in [101] describes a different approach in a distributed sensor environment. Groups of geographically-related sensors are first formed into sampler groups, which are essentially federations with a single agent called the sampler group leader acting as the intermediary. These groups then form the lowest level of a data aggregation hierarchy that exists above them. This arrangement is similar to the example organization shown in Figure 9. The sampler group leader collects raw data from the members of its group, and passes the data to its parent agent in the hierarchy, known as a zone leader. It is this zone leader's responsibility to interpret the sensor data to the best of its ability, by building motion equations and combining data perceived to be from the same target. This more abstract view is then passed to the next level of the hierarchy, where the process repeats. This will eventually terminate at the apex agent which should be able to reconstruct a global view from the abstract pieces it receives. The hierarchy itself is strict, communication is only permitted between connected agents, which reduces the level of sophisticated needed by the agents. The experimental results showed that this solution could scale to thousands of sensors and targets. The tradeoff they discovered was that shorter hierarchies produced more accurate results, because the fragmentation of the area was minimized, which in turn reduced the number of fusion processes data must survive before it is incorporated. Conversely, taller hierarchies dramatically reduced the computational load placed on any one agent, because the area each agent was responsible for became relatively small. By weighing these characteristics against the domain requirements one can select and appropriate structure to use.

11 Other Organizational Topics

In this survey we have focused entirely on particular organizational paradigms. However, there are a number of other topics related to organizational design which we will not cover in detail, but are sufficiently important to warrant mention. These are outlined below:

1. *Global Organizational Representation* Implicit in the concept of an intentional organizational design is an explicit representation of its structure. This is of use to designers, as a means of specification and exploration, and to the agents themselves, as a template and diagnostic tool. A number of modeling techniques have been proposed, notably by Fox [26], Tambe [93], Hübner [41], Pattison [72] and Sims [85].

2. *Local Organizational Representation* The organization's global view is not always the most appropriate vehicle to guide agents' behaviors. It can be too coarse in granularity, too qualitative or simply too large to be of practical use. Agents require a well-defined, quantitative mechanism that can be used to select appropriate local actions while respecting global organizational specifications. This process was originally described as *local elaboration* by March and Simon [61], where the activities performed by an agent are first constrained by its position in the organization, and then selected using local information and capabilities. The social consciousness model suggested by Glass and Grosz [34], Decker's TÆMS language [19], Shoham's social laws [83], and Wagner's MQ framework [99] provide ways to accomplish this.

3. *Generative Paradigms* In each section, we have presented different ways in which organizations may be formed. We have not, however, presented a unified discussion of specific generative paradigms – a classification of the techniques that may be used to produce organizations. These may be broadly separated into at least three classes: scripted, controlled and emergent. The first includes organizations that are produced from

statically predefined instructions, possibly from an external third party or during start-up. The second includes those that are explicitly applied to a population by an individual or group of individuals in response to perceived conditions. The third captures techniques which have no central or global direction, but are instead grown organically through the individual actions of agents. In practice, it may be difficult to clearly classify particular techniques. For example, congregations emerge from individual agent decisions using the technique described by Brooks [8]. However, the fact that it uses heuristics intended to simulate a controlled decision, along with agents which provide labels to guide the formation, gives the appearance of a controlled process.

4. *Organizational Adaptation* Although we have briefly touched on adaptation previously, an organization's ability to adapt is a general concept that is critical in any dynamic environment. The organization must have the ability to detect and react to changes in a timely manner in realistic, open domains [2, 38]. Related to this is the notion of social pathologies, which occur when an organization adapts inappropriately [45]. Any organizational change which occurs at runtime will have associated costs. These costs may be observed in direct consumption of resources, such as bandwidth or processing power, or indirectly because of inefficiencies or opportunities missed while in an intermediate state. The ability to adapt an organization depends on first recognizing potential problems, evaluating the costs and benefits of candidate solutions, and then implementing the selected changes.

5. *Coordination* Many of the organizational styles that we have covered assume some that some sort of interaction or coordination will take place between agents. This is seen in the authority relationships of hierarchies, the joint intentions of teams, data routing protocols in federations, and negotiations of society members. The characteristics provided by these interactions are critical to the effective qualities of these paradigms. For example, aggregating nodes and managers in hierarchies and intermediaries in federations frequently take on responsibilities related to coordination, by assigning tasks or routing information in such a way that interrelationships among its subordinates can be avoided [29]. Such capabilities can heavily influence the interactions needed for such a group to operate, ultimately affecting the organizational structure as well.

6. *Human Organizational Analogues* For much of the time that multi-agent organizations have been researched, attempts have been made to draw upon the large body of work that has been done one human organizations. The fields of sociology, anthropology, biology, economics, business management and formal organization theory (among others) contain a wealth of analytic and case study information describing how human organizations are structured and perform [28, 31]. Although on the surface much of this work is intimately tied to the human experience, attempts to extract concepts and abstractions have met with some success.

7. *Diversity* Although role assignment clearly plays a critical role in an organizational specification, the notion of agent diversity is rarely treated as or reasoned about as a first-class characteristic. As with stock portfolios, animal populations and security techniques, diversity can play an important role in agent systems susceptible to failure. Enforcing agent diversity through heterogeneous roles, agent types or division of labor, can impart semantic and capability fault-tolerance on the system as a whole [74, 14, 58]. Diversity can be embedded in the organizational design to encourage such characteristics.

12 Discussion

In this article we have presented a number of methods by which a multi-agent system could be organized. A brief comparison of the potential benefits and drawbacks of each strategy is summarized in Figure 10. A more complete depiction of the range of relevant organizational characteristics in general has been compiled by Carley and Gasser [10]. This information begins to provide information needed to select an appropriate organization, given the conditions and requirements under which the system is to be run. It should also be clear from this discussion that no single approach is better than all others; the selection made by a designer will be dictated by the needs imposed by the system's goals and the environment in which the participants will exist. That said, if one looks at the depth of available research and how frequently their concepts have been applied, it is the case that hierarchical, team-centric, and coalition-based organizations have proved to be most popular among multi-agent researchers. These three paradigms seem to offer the most in terms of flexibility, ease of implementation and their innate ability to produce demonstrable, positive effects. Hierarchies are very effective at addressing issues of scale, particular if the domain can be easily decomposed along some dimension. Teamwork of some sort is critical when working on large-grained tasks which require the coordinated capabilities of more than one agent. Coalitions allow agents to take advantage of economies of scale, without necessarily ceding authority to other agents. We furthermore feel that if

Paradigm	Benefits	Drawbacks
Hierarchy	Facilitates decomposition; handles scale well	Potentially brittle; can lead to bottlenecks or delays
Holarchy	Exploit autonomy of functional units	Must organize holons; lack of predictable performance
Coalition	Exploit strength in numbers	Short term benefits may not outweigh organization construction costs
Team	Address larger grained problems; increased cohesion	Increased communication
Congregation	Facilitates agent discovery	Sets may be overly restrictive
Society	Open, long-lived system; well defined conventions	Potentially complex, agents may require additional society-related capabilities
Federation	Matchmaking, brokering, translation services; facilitates dynamic agent pool	Intermediaries become bottlenecks
Matrix	Resource sharing; multiply-influenced agents	Potential for conflicts; need for increased agent sophistication
Compound	Exploit benefits of several organizational styles	Increased sophistication; drawbacks of several organizational styles

Figure 10. Comparing the qualities of various organization paradigms.

the broad vision of an agent-connected or agent-facilitated world which many proponents of multi-agent technology describe is to be realized, many of the characteristics of the agent society paradigm must be applied [31].

Other conditions may in fact preclude the use of particular paradigms. For instance, it can be difficult to generate optimal coalition or congregation structures when there is either limited time or a large population. When individual agent resources are constrained, particular instances of organizations which suffer from bottleneck effects, such as hierarchies, federations and holarchies, can become inefficient. We have also previously noted how some types of structures, such as matrices, societies, and certain compound organizations, require a somewhat higher level of sophistication of the participating agents.

As research progresses in these areas, typically by adding features and relaxing assumptions, it can become difficult to precisely categorize a particular approach. For example, we noted how hierarchies and holarchies are closely related, as are coalitions and congregations. To a certain extent, we have focused on the extreme or most constrained examples of organizations in this paper to delineate discrete classes, and it is frequently the case that the "rules" of a particular paradigm as we have presented them have been broken by someone attempting to broaden its abilities or applicability. While this might frustrate one's attempt at categorization, we believe the convergent evolution of these strategies towards a common form lends additional credence to the applicability of that form.

A somewhat more elusive goal is to define what exactly constitutes an organization in general. At what level of abstraction in the system's design should the influence of the organization stop and more transient "operational" decisions become more important? Must a structure exist for some period of time or some number of iterations before it is considered an organization? We have looked at strategies that are generally short-lived, such as coalitions, while societies may outlive the lifetime of any of its participants. Teams may exist to satisfy only a single goal, while federations see a continuous stream of different tasks. In each of these cases, the pattern of interactions between the agents is a defining characteristic, influencing the behaviors and qualities exhibited by the system. If this same pattern exists in two different circumstances, is one an organization and the other not? To a certain extent, this is just a matter of semantics, and we could just as easily name it a "pattern of interactions" and leave it at that. However, by labeling them all organizations allows one to more easily recognize that commonalities may exist between superficially different circumstances that are derived specifically from these interactions. Thus, we propose that under all circumstances this pattern serves as organization. The fact that it may exist for a single moment or a single task certainly impacts its performance and construction, but we believe much of the underlying purpose and qualities of the structuring remain the same, and should be recognized as such.

A popular approach not mentioned above is the (sparsely) connected graph structure, where agents interact because of particular role-based requirements but no overarching design principle is explicitly applied. The connection pattern superficially resembles a team, but without a team's strong interaction semantics. Typically, some aspects of the structure will be statically defined, although they may also emerge dynamically. Other times, an *adhocracy* might be employed, where the set of agent interactions may change in response to every newly recognized goal.

These approaches can be very effective and cost-efficient in some domains, however, as the environment scales or the agent population becomes more dynamic, a more structured organization can provide additional framework to address the more demanding context. Corkill and Lander [14] enumerate several other factors which motivate the need for explicit organization, including scarce resources, the potential for collaboration and the amount of repetition of work. As multi-agent technology is used to address more complex, real-world problems, it seems inevitable that they will encounter these problems, making organizational paradigms such as those we have presented an important component of these systems.

13 Acknowledgments

The authors would like to thank Sherief Abdallah, Dan Corkill and Mark Sims for their helpful critiques and comments on this paper.

References

[1] A. Artikis and J. Pitt. A formal model of open agent societies. In *Proceedings of the fifth international conference on Autonomous agents*, pages 192–193. ACM Press, 2001.

[2] K. S. Barber and C. E. Martin. Dynamic adaptive autonomy in multiagent systems: Representation and justification. *International Journal of Pattern Recognition and Artificial Intelligence*, 15(3):405–433, 2001.

[3] G. Beavers and H. Hexmoor. Teams of agents. In *Proceedings of the IEEE Systems, Man, and Cybernetics Conference*, pages 574–582, 2001.

[4] A. H. Bond and L. Gasser. An analysis of problems and research in DAI. In A. H. Bond and L. Gasser, editors, *Readings in Distributed Artificial Intelligence*, pages 3–35. Morgan Kaufmann, 1988.

[5] L. Bongaerts. *Integration of Scheduling and Control in Holonic Manufacturing Systems*. PhD thesis, Katholieke Universiteit Leuven, Belgium, 1998.

[6] S. Breban and J. Vassileva. Long-term coalitions for the electronic marketplace. In *Proceedings of Canadian AI Workshop on Novel E-Commerce Applications of Agents*, pages 6–12, 2001.

[7] C. Brooks and E. Durfee. Congregating and market formation. In *Proceedings of the first international joint conference on Autonomous agents and multiagent systems*, pages 96–103. ACM Press, 2002.

[8] C. Brooks and E. Durfee. Congregation formation in multiagent systems. *Journal of Autonomous Agents and Multiagent Systems*, 7(1-2):145–170, 2003.

[9] C. Brooks, E. Durfee, and A. Armstrong. An introduction to congregating in multiagent systems. In *Proceedings of the Fourth International Conference on Multiagent Systems*, pages 79–86, 2000.

[10] K. M. Carley and L. Gasser. Computational organization theory. In *Multiagent systems: a modern approach to distributed artificial intelligence*, pages 299–330. MIT Press, 1999.

[11] K. M. Carley and L. Gasser. Computational organization theory. In G. Weiss, editor, *Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence*, pages 299–330. The MIT Press, 1999.

[12] B. Chandrasekaran. Natural and social system metaphors for distributed problem solving: Introduction to the issue. *IEEE Transactions on Systems, Man, and Cybernetics*, 11(1):1–5, Jan. 1981.

[13] D. Corkill and V. Lesser. The use of meta-level control for coordination in a distributed problem solving network. *Proceedings of the Eighth International Joint Conference on Artificial Intelligence*, pages 748–756, August 1983.

[14] D. D. Corkill and S. E. Lander. Diversity in Agent Organizations. *Object Magazine*, 8(4):41–47, May 1998.

[15] M. R. Cutkosky, R. S. Englemore, R. E. Fikes, M. R. Genesereth, T. R. Gruber, W. S. Mark, J. M. Tenenbaum, and J. C. Weber. PACT: An experiment in integrating concurrent engineering systems. In M. N. Huhns and M. P. Singh, editors, *Readings in Agents*, pages 46–55. Morgan Kaufmann, San Francisco, CA, USA, 1997.

[16] K. Decker. TAEMS: A Framework for Environment Centered Analysis & Design of Coordination Mechanisms. In *Foundations of Distributed Artificial Intelligence, Chapter 16*, pages 429–448. G. O'Hare and N. Jennings (eds.), Wiley Inter-Science, January 1996.

[17] K. Decker and V. Lesser. Generalizing the Partial Global Planning Algorithm. *International Journal on Intelligent Cooperative Information Systems*, 1(2):319–346, June 1992.

[18] K. Decker, V. Lesser, N. Prasad, and T. Wagner. MACRON: An Architecture for Multi-Agent Cooperative Information Gathering. *Proccedings of the CIKM Workshop on Intelligent Information Agents*, December 1995.

[19] K. Decker and V. R. Lesser. Quantitative Modeling of Complex Environments. *International Journal of Intelligent Systems in Accounting, Finance and Management. Special Issue on Mathematical and Computational Models and Characteristics of Agent Behaviour.*, 2:215–234, January 1993.

[20] K. Decker, K. Sycara, and M. Williamson. Intelligent adaptive information agents. In I. Imam, editor, *Working Notes of the AAAI-96 Workshop on Intelligent Adaptive Agents*, Portland, OR, 1996.

[21] C. Dellarocas and M. Klein. Civil agent societies: Tools for inventing open agent-mediated electronic marketplaces. In *Agent Mediated Electronic Commerce (IJCAI Workshop)*, pages 24–39, 1999.

[22] V. Dignum, J.-J. Meyer, H. Weigand, and F. Dignum. An organizational-oriented model for agent societies. *Proceedings of International Workshop on Regulated Agent-Based Social Systems: Theories and Applications (RASTA'02)*, July 2002.

[23] E. Durfee and V. Lesser. Partial Global Planning: A Coordination Framework for Distributed Hypothesis Formation. *IEEE Transactions on Systems, Man, and Cybernetics*, 21(5):1167–1183, September 1991.

[24] S. S. Fatima and M. Wooldridge. Adaptive task and resource allocation in multi-agent systems. In J. P. Müller, E. Andre, S. Sen, and C. Frasson, editors, *Proceedings of the Fifth International Conference on Autonomous Agents*, pages 537–544, Montreal, Canada, 2001. ACM Press.

[25] K. Fischer. Agent-based design of holonic manufacturing systems. *Journal of Robotics and Autonomous Systems*, 1999.

[26] M. Fox, M. Barbuceanu, M. Gruninger, and J. Lin. An Organizational Ontology for Enterprise Modeling. In M. J. Prietula, K. M. Carley, and L. Gasser, editors, *Simulating Organizations: Computational Models of Institutions and Groups*, pages 131–152. AAAI Press / MIT Press, 1998.

[27] M. S. Fox. Organization structuring: Designing large complex software. Computer Science Technical Report CMU-CS-79-155, Carnegie-Mellon University, December 1979.

[28] M. S. Fox. An organizational view of distributed systems. *IEEE Transactions on Systems, Man, and Cybernetics*, 11(1):70–80, Jan. 1981.

[29] J. Galbraith. *Organization Design*. Addison-Wesley, Reading, MA, 1977.

[30] L. Gasser. Social conceptions of knowledge and action: DAI foundations and open systems semantics. *Artificial Intelligence*, 47(1-3):107–138, 1991.

[31] L. Gasser. Perspectives on organizations in multi-agent systems. In *Mutli-agents systems and applications*, pages 1–16. Springer-Verlag New York, Inc., 2001.

[32] M. R. Genesereth. An agent-based framework for interoperability. In J. Bradshaw, editor, *Software agents*, pages 317–345. MIT Press, 1997.

[33] N. Glaser and P. Morignot. Societies of autonomous agents and their reorganisation. In Tschacher and Dauwalder, editors, *Nonlinear Systems Approaches to Cognitive Psychology and Cognitive Science — Dynamics, Synergetics, Autonomous Agents*. World Scientific, 1998.

[34] A. Glass and B. Grosz. Socially conscious decision-making. In C. Sierra, M. Gini, and J. S. Rosenschein, editors, *Proceedings of the Fourth International Conference on Autonomous Agents*, pages 217–224, Barcelona, Catalonia, Spain, 2000. ACM Press.

[35] B. J. Grosz and S. Kraus. Collaborative plans for complex group action. *Artificial Intelligence*, 86(2):269–357, 1996.

[36] S. Hayden, C. Carrick, and Q. Yang. Architectural design patterns for multi-agent coordination. In *Proceedings of the International Conference on Agent Systems '99*, May 1999.

[37] H. Hexmoor and G. Beavers. Towards teams of agents. In *Proceedings of the International Conference in Artificial Intelligence (IC-AI'2001)*. CSREA Press, 2001.

[38] B. Horling, B. Benyo, and V. Lesser. Using Self-Diagnosis to Adapt Organizational Structures. *Proceedings of the 5th International Conference on Autonomous Agents*, pages 529–536, June 2001.

[39] B. Horling, R. Mailler, and V. Lesser. A Case Study of Organizational Effects in a Distributed Sensor Network. In *Proceedings of the AAAI-04 Workshop on Agent Organizations: Theory and Practice*, San Jose, California, July 2004.

[40] B. Horling, R. Mailler, J. Shen, R. Vincent, and V. Lesser. Using Autonomy, Organizational Design and Negotiation in a Distributed Sensor Network. In V. Lesser, C. Ortiz, and M. Tambe, editors, *Distributed Sensor Networks: A multiagent perspective*, pages 139–183. Kluwer Academic Publishers, 2003.

[41] J. F. Hübner, J. S. Sichman, and O. Boissier. A model for the structural, functional, and deontic specification of organizations in multiagent systems. In *Proceedings of the Brazilian Symposium on Artificial Intelligence (SBIA'02)*, pages 118–128.

[42] M. N. Huhns and L. M. Stephens. Multiagent systems and societies of agents. In G. Weiss, editor, *Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence*, pages 79–120. The MIT Press, Cambridge, MA, USA, 1999.

[43] T. Ishida, L. Gasser, and M. Yokoo. Organization self-design of distributed production systems. *IEEE Transactions on Knowledge and Data Engineering*, 4(2):123–134, 1992.

[44] N. Jennings. Controlling cooperative problem solving in industrial multi-agent systems. *Artificial Intelligence*, 75(2):195–240, 1995.

[45] D. Jensen and V. Lesser. Social Pathologies of Adaptive Agents. *Safe Learning Agents: Papers from the 2002 AAAI Spring Symposium.*, TR SS-02-07, August 2002.

[46] G. A. Kaminka, D. V. Pynadath, and M. Tambe. Monitoring teams by overhearing: A multi-agent plan-recognition approach. *Journal of Artificial Intelligence Research*, 17:83–135, 2002.

[47] T. Khedro and M. Genesereth. Facilitators: A networked computing infrastructure for distributed software interoperation. In *Working Notes of the IJCAI-95 Workshop on Artificial Intelligence in Distributed Information Networks.*, 1995.

[48] M. Klein, J. Rodriguez-Aguilar, and C. Dellarocas. Using domain-independent exception handling services to enable robust open multi- agent systems: The case of agent death. *Journal of Autonomous Agents and MultiAgent Systems*, 7(1-2):179–189, 2003.

[49] M. Klusch and A. Gerber. Dynamic coalition formation among rational agents. *IEEE Intelligent Systems*, 17(3):42–47, 2002.

[50] A. Koestler. *The Ghost In The Machine*. Hutchinson Publishing Group, Arkana, London, 1967.

[51] K. Larson and T. Sandholm. Anytime coalition structure generation: An average case study. *Journal of Experimental and Theoretical AI*, 11:1–20, 2000.

[52] K. Lerman and O. Shehory. Coalition formation for large-scale electronic markets. In *Proceedings of the International Conference on Multi-Agent Systems (ICMAS'2000)*, 2000.

[53] V. Lesser. A Retrospective View of FA/C Distributed Problem Solving. *IEEE Transactions on Systems, Man, and Cybernetics*, 21(6):1347–1363, November 1991.

[54] V. Lesser. Reflections on the Nature of Multi-Agent Coordination and Its Implications for an Agent Architecture. *Autonomous Agents and Multi-Agent Systems*, 1:89–111, January 1998.

[55] V. Lesser, K. Decker, T. Wagner, N. Carver, A. Garvey, B. Horling, D. Neiman, R. Podorozhny, M. NagendraPrasad, A. Raja, R. Vincent, P. Xuan, and X. Zhang. Evolution of the GPGP/TAEMS Domain-Independent Coordination Framework. *Autonomous Agents and Multi-Agent Systems*, 9(1):87–143, July 2004.

[56] V. Lesser and L. Erman. Distributed Interpretation: A Model and an Experiment. *IEEE Transactions on Computers Special Issue on Distributed Processing*, C-29(12):1144–1163, December 1980.

[57] H. J. Levesque, P. R. Cohen, and J. H. T. Nunes. On acting together. In *Proceedings of the Eighth National Conference on Artificial Intelligence*, pages 94–99, July 1990.

[58] D. Lybäck. Transient diversity in multi-agent systems. Master's thesis, Department of Computer and Systems Sciences, Stockholm University and the Royal Institute of Technology, September 1999.

[59] R. Mailler and V. Lesser. Solving Distributed Constraint Optimization Problems Using Cooperative Mediation. *Proceedings of Third International Joint Conference on Autonomous Agents and MultiAgent Systems (AAMAS 2004)*, 2004.

[60] R. Mailler, V. Lesser, and B. Horling. Cooperative Negotiation for Soft Real-Time Distributed Resource Allocation. In *Proceedings of Second International Joint Conference on Autonomous Agents and MultiAgent Systems (AAMAS 2003)*, pages 576–583, Melbourne, July 2003. ACM Press.

[61] J. G. March and H. A. Simon. *Organizations*. Wiley, New York, 1958.

[62] S. Marsella, M. Tambe, J. Adibi, Y. Al-Onaizan, G. A. Kaminka, and I. Muslea. Experiences acquired in the design of robocup teams: A comparison of two fielded teams. *Autonomous Agents and Multi-Agent Systems*, 4(1/2):115–129, 2001.

[63] P. Mathieu, J. C. Routier, and Y. Secq. Dynamic organization of multi-agent systems. In *Proceedings of the first international joint conference on Autonomous Agents and Multi-Agent Systems*. ACM Press, 2002.

[64] E. Matson and S. DeLoach. Using dynamic capability evaluation to organize a team of cooperative, autonomous robots. In *Proceedings of The 2003 International Conference on Artificial Intelligence (IC-AI'03)*, 2003.

[65] F. Maturana, W. Shen, and D. Norrie. Metamorph: An adaptive agent-based architecture for intelligent manufacturing. *International Journal of Production Research*, 37(10):2159–2174, 1999.

[66] T. A. Montgomery and E. H. Durfee. Search reduction in hierarchical distributed problem solving. In *Proceedings of the Eleventh International Workshop on Distributed AI*, Glen Arbor, Michigan, Feb. 1992.

[67] Y. Moses and M. Tennenholtz. Artificial social systems. *Computers and AI*, 14(6):533–562, 1995.

[68] R. Nair, T. Ito, M. Tambe, and S. Marsella. The role of emotions in multiagent teamwork: A preliminary investigation. In *Who needs emotions: the brain meets the robot*. Oxford University Press, 2003.

[69] R. Nair, M. Tambe, and S. Marsella. Role allocation and reallocation in multiagent teams: Towards a practical analysis. In *Proceedings of Second International Joint Conference on Autonomous Agents and Multi-agent Systems (AAMAS-03)*, 2003.

[70] Y. pa So and E. H. Durfee. Designing organizations for computational agents. In *Simulating Organizations*, pages 47–64. AAAI Press/ MIT Press, 1998.

[71] L. E. Parker. Designing control laws for cooperative agent teams. In *IEEE International Conference on Robotics and Automation*, pages 582–587, 1993.

[72] H. E. Pattison, D. D. Corkill, and V. R. Lesser. Instantiating Descriptions of Organizational Structures. *Distributed Artificial Intelligence, Research Notes in Artificial Intelligence*, pages 59–96, 1987.

[73] D. V. Pynadath and M. Tambe. The communicative multiagent team decision problem: Analyzing teamwork theories and models. *Journal of AI research*, 16:389–423, 2002.

[74] S. Reed and V. Lesser. Division of Labor in Honey Bees and Distributed Focus of Attention. *University of Massachusetts/Amherst Computer and Information Science Department Technical Report 80-17*, November 1980.

[75] P. Romelaer. Organization: A Diagnosis Method. Technical Report 78, University Paris IX Dauphine, Crepa Laboratory, June 2002.

[76] T. Sandholm, K. Larson, M. Andersson, O. Shehory, and F. Tohme. Coalition structure generation with worst case guarantees. *Artificial Intelligence*, 111(1-2):209–238, 1999.

[77] T. Sandholm and V. Lesser. Coalitions Among Computationally Bounded Agents. *Artificial Intelligence, Special Issue on Economic Principles of Multi-Agent Systems*, 94(1):99–137, January 1997.

[78] M. Schwaninger. A theory for optimal organization. Technical Report 38, Institute of Management at the University of St. Gallen, Switzerland, 2000.

[79] S. Sen. Reciprocity: a foundational principle for promoting cooperative behavior among self-interested agents. In *Proc. of the Second International Conference on Multiagent Systems*, pages 322–329, 1996.

[80] O. Shehory and S. Kraus. Methods for task allocation via agent coalition formation. *Artificial Intelligence*, 101(1–2):165–200, 1998.

[81] O. Shehory, K. Sycara, P. Chalasani, and S. Jha. Agent cloning: an approach to agent mobility and resource allocation. *IEEE Communications Magazine*, 36(7):58–67, 1998.

[82] W. Shen and D. Norrie. A hybrid agent-oriented infrastructure for modeling manufacturing. In *Proceedings of the Knowledge Acquisition Workshop (KAW'98)*, 1998.

[83] Y. Shoham and M. Tennenholtz. On social laws for artificial agent societies: Off-line design. *Artificial Intelligence*, 73(1-2):231–252, 1995.

[84] H. A. Simon. *The Sciences of the Artificial*. MIT Press, Cambridge, MA, 1968.

[85] M. Sims, D. Corkill, and V. Lesser. Separating Domain and Coordination in Multi-Agent Organizational Design and Instantiation. In *Submitted to IAT 2004*, April 2004.

[86] M. Sims, C. Goldman, and V. Lesser. Self-Organization through Bottom-up Coalition Formation. In *Proceedings of Second International Joint Conference on Autonomous Agents and MultiAgent Systems (AAMAS 2003)*, pages 867–874, Melbourne, AUS, July 2003. ACM Press.

[87] N. Singh, M. R. Genesereth, and M. Syed. A distributed and anonymous knowledge sharing approach to software interoperation. *International Journal of Cooperative Information Systems*, 4(4):339–368, 1995.

[88] R. G. Smith. The contract net protocol: High-level communication and control in a distributed problem solver. *IEEE Transctions on Computers*, 29(12):1104–1113, 1980.

[89] L.-K. Soh, C. Tsatsoulis, and H. Sevay. A Satisfi cing, Negotiated, and Learning Coalition Formation Architecture. In V. Lesser, C. Ortiz, and M. Tambe, editors, *Distributed Sensor Networks: A multiagent perspective*, pages 109–138. Kluwer Academic Publishers, 2003.

[90] D. Stefanoiu, M. Ulieru, and D. Norrie. Fuzzy modeling of multi-agent systems behavior. vagueness minimization. In *Proceedings of World Multiconference on Systemics, Cybernetics and Informatics (SCI'2000)*, volume III, pages 118–123, July 2000.

[91] K. Sycara, K. Decker, and M. Williamson. Middle-agents for the internet. In *Proceedings of IJCAI-97*, January 1997.

[92] M. Tambe. Towards fexible teamwork. *Journal of Artifi cial Intelligence Research*, 7:83–124, 1997.

[93] M. Tambe, J. Adibi, Y. Alonaizon, A. Erdem, G. A. Kaminka, S. Marsella, and I. Muslea. Building agent teams using an explicit teamwork model and learning. *Artifi cial Intelligence*, 110(2):215–239, 1999.

[94] G. Tidhar, C. Heinze, and M. Selvestrel. Flying together: Modelling air mission teams. *Journal of Applied Intelligence*, 8:3, May 1998.

[95] G. Tidhar, A. Rao, and L. Sonenberg. Guided team selection. In *ICMAS96 proceedings*, pages 369–376, Japan, 1996.

[96] M. Tsvetovat and K. Sycara. Customer coalitions in the electronic marketplace. In C. Sierra, M. Gini, and J. S. Rosenschein, editors, *Proceedings of the Fourth International Conference on Autonomous Agents*, pages 263–264, Barcelona, Catalonia, Spain, 2000. ACM Press.

[97] M. Ulieru. Emergence of holonic enterprises from multi-agent systems: A fuzzy-evolutionary approach. In V. Loia, editor, *Soft Computing Agents: A New Perspective on Dynamic Information Systems*, pages 187–215. IOS Press, 2002.

[98] M. Ulieru, S. Walker, and R. Brennan. Holonic enterprise as a collaborative information ecosystem. In *In Proceedings of the Workshop on Holons: Autonomous and Cooperating Agents for Industry 2001*, 2001.

[99] T. Wagner and V. Lesser. Relating Quantifi ed Motivations for Organizationally Situated Agents. In *Intelligent Agents VI — Proceedings of the Sixth International Workshop on Agent Theories, Architectures, and Languages, Lecture Notes in Artifi cial Intelligence*, pages 334–348. N. R. Jennings and Y. Lesperance (eds.), Springer-Verlag, Berlin, April 1999.

[100] M. P. Wellman and P. R. Wurman. Market-aware agents for a multiagent world. *Robotics and Autonomous Systems*, 24:115–125, 1998.

[101] O. Yadgar, S. Kraus, and C. Ortiz. Scaling up distributed sensor networks: cooperative large-scale mobile-agent organizations. In V. Lesser, C. Ortiz, and M. Tambe, editors, *Distributed Sensor Networks: a multiagent perspective*, pages 185–218. Kluwer publisching, 2003.

[102] X. Zhang and D. Norrie. Holonic control at the production and controller levels. In *Proceedings of IMS 99*, pages 215–224, 1999.